Linear Programming Notes

Lecturer: David Williamson, Cornell ORIE
Scribe: Kevin Kircher, Cornell MAE

These notes summarize the central definitions and results of the theory of linear programming, as taught by David Williamson in ORIE 6300 at Cornell University in the fall of 2014. Proofs and discussion are mostly omitted. These notes also draw on *Convex Optimization* by Stephen Boyd and Lieven Vandenberghe, and on Stephen Boyd's notes on ellipsoid methods. Prof. Williamson's full lecture notes can be found here.

Contents

1 The linear programming problem

A linear program (LP) is an optimization problem with objective and constraint functions that are linear in the optimization variables. Formally, the general LP problem is

$$\begin{array}{ll} \text{maximize} & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & A\mathbf{x} \leq \mathbf{b} \end{array} \tag{1}$$

where $A \in \mathbf{R}^{m \times n}$ and $m \geq n$. The inequalities are interpreted component-wise: $A\mathbf{x} \leq \mathbf{b}$ means that $(A\mathbf{x})_i \leq b_i$ for all $i \in \{1, \ldots, m\}$. It's often useful to refer to the columns $\mathbf{a}_j$ or rows $\boldsymbol{\alpha}_i^T$ of A:

$$A = \begin{bmatrix} \mathbf{a}_1 & \ldots & \mathbf{a}_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{\alpha}_1^T \\ \vdots \\ \boldsymbol{\alpha}_m^T \end{bmatrix}.$$

We will occasionally abbreviate LP (1) as $\max\{\mathbf{c}^T\mathbf{x} \mid A\mathbf{x} \leq \mathbf{b}\}$.
Vocabulary:

- The **optimization variables** (or just **variables**) are $x_1, \ldots, x_n$.
- The **objective function** is $\mathbf{c}^T\mathbf{x}$.
- The **data** or **inputs** are A, $\mathbf{b}$, and $\mathbf{c}$.
- The **feasible region** is the set $\mathcal{F}(P) = \{\mathbf{x} \in \mathbf{R}^n \mid A\mathbf{x} \leq \mathbf{b}\}$. If $\mathcal{F}(P)$ is nonempty, then the LP is **feasible**.
- If $\mathbf{c}^T\mathbf{x}^* \leq \mathbf{c}^T\mathbf{x}$ for all $\mathbf{x} \in \mathcal{F}(P)$, then $\mathbf{x}^*$ is an **optimal solution** (or just **solution**) and $\mathbf{c}^T\mathbf{x}^*$ is the **optimal value** (or just **value**) of the LP.
- If there exist an $\mathbf{x} \in \mathcal{F}(P)$ and a direction $\mathbf{d} \in \mathbf{R}^n$ such that $\mathbf{c}^T\mathbf{d} > 0$ and $\mathbf{x}+\lambda\mathbf{d} \in \mathcal{F}(P)$ for all $\lambda \geq 0$, then LP (1) is **unbounded**.

The LP (1) is in **basic form**. The general **standard form** LP is

$$\begin{array}{ll} \text{minimize} & \tilde{\mathbf{c}}^T\tilde{\mathbf{x}} \\ \text{subject to} & \tilde{A}\tilde{\mathbf{x}} = \tilde{\mathbf{b}} \\ & \tilde{\mathbf{x}} \geq \mathbf{0} \end{array} \tag{2}$$

where $\tilde{A} \in \mathbf{R}^{\tilde{m} \times \tilde{n}}$, $\tilde{m} \leq \tilde{n}$, and $\operatorname{rank}(\tilde{A}) = \tilde{m}$. The standard form LP (2) can be written in basic form by identifying

$$\mathbf{x} = \tilde{\mathbf{x}}, \quad \mathbf{c} = -\tilde{\mathbf{c}}, \quad A = \begin{bmatrix} \tilde{A} \\ -\tilde{A} \\ -I \end{bmatrix} \in \mathbf{R}^{(2\tilde{m}+\tilde{n}) \times \tilde{n}}, \quad \mathbf{b} = \begin{bmatrix} \tilde{\mathbf{b}} \\ -\tilde{\mathbf{b}} \\ \mathbf{0} \end{bmatrix}$$

because $\operatorname{argmin} \tilde{\mathbf{c}}^T \tilde{\mathbf{x}} = \operatorname{argmax} -\tilde{\mathbf{c}}^T \tilde{\mathbf{x}}$; $\tilde{A}\tilde{\mathbf{x}} = \tilde{\mathbf{b}} \iff (\tilde{A}\tilde{\mathbf{x}} \leq \tilde{\mathbf{b}}$ and $-\tilde{A}\tilde{\mathbf{x}} \leq -\tilde{\mathbf{b}})$; and $\tilde{\mathbf{x}} \geq \mathbf{0} \iff -I\tilde{\mathbf{x}} \leq \mathbf{0}$. With these definitions, LPs (1) and (2) are equivalent: although the objective value of one is the negative of the other, they have the same feasible regions and optimal solutions.

Similarly, the basic form LP (1) can be written in standard form by identifying

$$\tilde{\mathbf{x}} = \begin{bmatrix} \mathbf{x}^+ \\ \mathbf{x}_- \\ \mathbf{s} \end{bmatrix}, \quad \tilde{\mathbf{c}} = \begin{bmatrix} -\mathbf{c} \\ \mathbf{c} \\ \mathbf{0} \end{bmatrix}, \quad \tilde{A} = \begin{bmatrix} A & -A & I \end{bmatrix} \in \mathbf{R}^{m \times (2n+m)}, \quad \tilde{\mathbf{b}} = \mathbf{b}$$

because $\mathbf{x} \in \mathbf{R}^n \iff (\mathbf{x} = \mathbf{x}^+ - \mathbf{x}_-$ and $\mathbf{x}^+, \mathbf{x}_- \geq \mathbf{0})$; $\operatorname{argmax} \mathbf{c}^T\mathbf{x} = \operatorname{argmin} -\mathbf{c}^T\mathbf{x}$; and $A\mathbf{x} \leq \mathbf{b} \iff (A\mathbf{x} + \mathbf{s} = \mathbf{b}$ and $\mathbf{s} \geq \mathbf{0})$. The variables $s_1, \ldots, s_m$, which are introduced to transform inequality constraints into equality constraints and nonnegativity constraints, are called **slack variables**.

2 Duailty

Much LP theory is built around the concept of duality: any **primal** LP in the optimization variables $\mathbf{x} \in \mathbf{R}^n$ has a **dual** LP in the dual variables $\mathbf{y} \in \mathbf{R}^m$.

Rules for taking a dual:

Primal	Dual
maximize $\mathbf{c}^T\mathbf{x}$	minimize $\mathbf{b}^T\mathbf{y}$
$A\mathbf{x} \le \mathbf{b}$	$\mathbf{y} \ge \mathbf{0}$
$\mathbf{x} \in \mathbf{R}^n$	$A^T\mathbf{y} = \mathbf{c}$
$A\mathbf{x} = \mathbf{b}$	$\mathbf{y} \in \mathbf{R}^m$
$\mathbf{x} \ge \mathbf{0}$	$A^T\mathbf{y} \ge \mathbf{c}$

So the dual of the basic form primal LP (1) is

$$\begin{array}{ll} \text{minimize} & \mathbf{b}^T\mathbf{y} \\ \text{subject to} & A^T\mathbf{y} = \mathbf{c} \\ & \mathbf{y} \ge \mathbf{0} \end{array} \tag{3}$$

and the dual of the standard form primal LP (2) is

$$\begin{array}{ll} \text{maximize} & \tilde{\mathbf{b}}^T\mathbf{y} \\ \text{subject to} & \tilde{A}^T\mathbf{y} \le \tilde{\mathbf{c}}. \end{array} \tag{4}$$

We denote the feasible region of the dual LP by $\mathcal{F}(D)$.

Theorem 1 *The dual of the dual is the primal.*

Theorem 2 (Weak duality) *The optimal value of the primal (1) is less than or equal to the optimal value of the dual (3).*

Proof:

$$\mathbf{c}^T\mathbf{x} = (A^T\mathbf{y})^T\mathbf{x} = \mathbf{y}^T(A\mathbf{x}) \le \mathbf{y}^T\mathbf{b}$$

where the last inequality follows from $A\mathbf{x} \le \mathbf{b}$ and $\mathbf{y} \ge \mathbf{0}$. □

It can also be shown that the optimal value of the primal (2) is greater than or equal to the optimal value of the dual (4).

3 Geometry

After solving a few low-dimensional LPs by hand, a pattern emerges: optima are 'corners' of the feasible region. This section formalizes that idea.

Definition 1 (Convex set) *A set $C \subseteq \mathbf{R}^n$ is convex if for all $\mathbf{x}, \mathbf{y} \in C$ and for all $\lambda \in [0,1]$, $\lambda\mathbf{x} + (1-\lambda)\mathbf{y} \in C$.*

Definition 2 (Polyhedron) *The set $P = \{\mathbf{x} \in \mathbf{R}^n \mid A\mathbf{x} \leq \mathbf{b}\}$ is a polyhedron.*

Polyhedra are convex.

Definition 3 (Vertex) *Let $C \subseteq \mathbf{R}^n$ be a convex set. A vector $\mathbf{x} \in C$ is a vertex of C if there exists a $\mathbf{c} \in \mathbf{R}^n$ such that for all $\mathbf{y} \in C$ with $\mathbf{y} \neq \mathbf{x}$, $\mathbf{c}^T\mathbf{x} < \mathbf{c}^T\mathbf{y}$.*

In other words, a vertex is an element of C that is the unique minimizer of $\mathbf{c}^T\mathbf{x}$ on C, for some cost vector $\mathbf{c}$.

Definition 4 (Extreme point) *Let C be a convex set. A vector $\mathbf{x} \in C$ is an extreme point of C if $\mathbf{x}$ cannot be written as $\lambda\mathbf{y} + (1-\lambda)\mathbf{z}$ for any $\mathbf{y}, \mathbf{z} \in C$, $\mathbf{y}, \mathbf{z} \neq \mathbf{x}$, $\lambda \in [0,1]$.*

In other words, an extreme point of C can't be written as a convex combination of any other points in C.

Definition 5 (Basic solution) *Let $P = \{\mathbf{x} \in \mathbf{R}^n \mid A\mathbf{x} \leq \mathbf{b}\}$, and let $A_=(\mathbf{x})$ be a matrix such that a row $\boldsymbol{\alpha}_i^T$ of A is a row of $A_=(\mathbf{x})$ if and only if $\boldsymbol{\alpha}_i^T\mathbf{x} = b_i$. A vector $\mathbf{x} \in \mathbf{R}^n$ is a basic solution of P if $\operatorname{rank}(A_=(\mathbf{x})) = n$.*

In other words, at least n linearly independent constraints (and possibly more linearly dependent ones) are binding at a basic solution.

There are at most $\binom{m}{n}$ basic solutions of P.

Definition 6 (Basic feasible solution) *A vector $\mathbf{x} \in \mathbf{R}^n$ is a basic feasible solution of P if $\mathbf{x} \in P$ and $\mathbf{x}$ is a basic solution of P.*

Theorem 3 (Characterization of vertices) *Let $P = \{\mathbf{x} \in \mathbf{R}^n \mid A\mathbf{x} \leq \mathbf{b}\}$ be nonempty. The following are equivalent:*

(1) $\mathbf{x}$ *is a vertex of P,*

(2) $\mathbf{x}$ *is an extreme point of P, and*

(3) $\mathbf{x}$ *is a basic feasible solution of P.*

Definition 7 (Convex combination) *A vector $\mathbf{v} \in \mathbf{R}^n$ is a convex combination of $\mathbf{v}_1, \ldots, \mathbf{v}_k$ if there exists a $\boldsymbol{\lambda} \in \mathbf{R}^k$ such that $\boldsymbol{\lambda} \geq \mathbf{0}$, $\mathbf{1}^T\boldsymbol{\lambda} = 1$, and $\mathbf{v} = \sum_{i=1}^k \lambda_i \mathbf{v}_i$.*

Definition 8 (Convex hull) *The set* $\mathrm{conv}(\mathbf{v}_1, \ldots, \mathbf{v}_k) \subseteq \mathbf{R}^n$*, called the convex hull of* $\mathbf{v}_1, \ldots, \mathbf{v}_k$*, is the set of all convex combinations of* $\mathbf{v}_1, \ldots, \mathbf{v}_k \in \mathbf{R}^n$*.*

Theorem 4 (Carathéodory) *If* $\mathbf{x} \in \mathrm{conv}(\mathbf{v}_1, \ldots, \mathbf{v}_k) \subseteq \mathbf{R}^n$*, then* $\mathbf{x}$ *can be written as a convex combination of* $n + 1$ *or fewer of the* $\mathbf{v}_i$*.*

Compare Carathéodory's theorem to the analogous result for linear combinations: if $\mathbf{x} \in \mathrm{span}(\mathbf{v}_1, \ldots, \mathbf{v}_k) \subseteq \mathbf{R}^n$, then $\mathbf{x}$ can be written as a linear combination of n or fewer of the $\mathbf{v}_i$.

Definition 9 (Polytope) *The convex hull of a finite number of vectors is a polytope.*

- Polytopes are convex.
- Any extreme point of the polytope $\mathrm{conv}(\mathbf{v}_1, \ldots, \mathbf{v}_k)$ is one of the $\mathbf{v}_i$.

Lemma 5 *Let* $Q = \mathrm{conv}(\mathbf{v}_1, \ldots, \mathbf{v}_k)$ *with* $\mathbf{v}_i \in \mathbf{R}^n$ *for all* i*. For any cost* $\mathbf{c} \in \mathbf{R}^n$*, at least one of the* $\mathbf{v}_i$ *satisfies* $\mathbf{c}^T\mathbf{v}_i \leq \mathbf{c}^T\mathbf{x}$ *for all* $\mathbf{x} \in Q$*.*

Definition 10 (Bounded polyhedron) *A polyhedron* P *is bounded if there exists an* $M > 0$ *such that for all* $\mathbf{x} \in P$*,* $\|\mathbf{x}\| \leq M$*.*

Theorem 6 (Representation of bounded polyhedra) *A bounded polyhedron is the convex hull of its vertices, and is therefore a polytope.*

The proof is by induction on the rank of $A_=(\mathbf{x})$ and hinges on the equivalence of vertices, extreme points and basic feasible solutions.

Note that the boundedness assumption is necessary (see Figure 1).

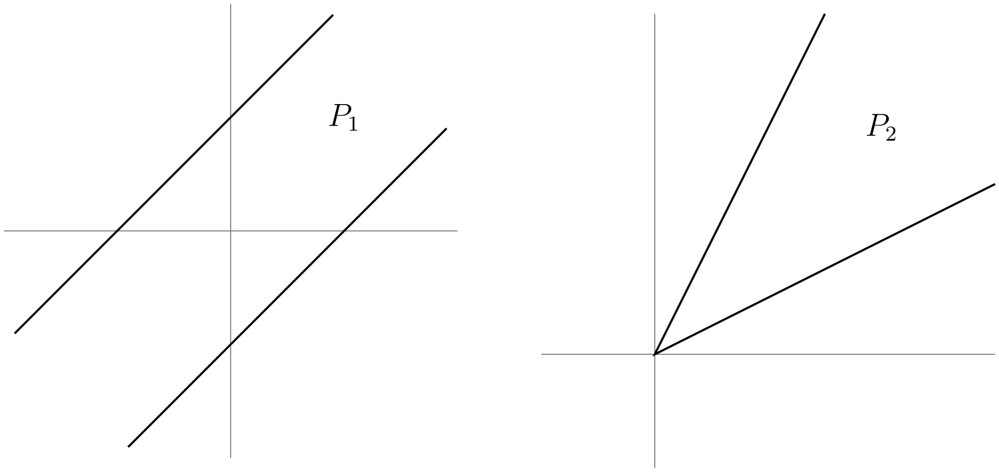

Figure 1: the unbounded polyhedra P_1 and P_2 are not polytopes.

Theorem 7 (Separating hyperplane) *Let $C \subseteq \mathbf{R}^n$ be closed, nonempty and convex, and let $\mathbf{y} \in \mathbf{R}^n$, $\mathbf{y} \notin C$. Then there exist an $\mathbf{a} \in \mathbf{R}^n$ and $b \in \mathbf{R}$ such that $\mathbf{a}^T\mathbf{y} > b$ and $\mathbf{a}^T\mathbf{x} < b$ for all $\mathbf{x} \in C$.*

The proof requires the Weierstrass theorem from real analysis (if $C \subseteq \mathbf{R}^n$ is closed, nonempty and bounded, then $f : C \to \mathbf{R}$ attains a maximum and minimum on C.)

Definition 11 (Polar) *The polar of a set $S \subseteq \mathbf{R}^n$ is $S^\circ = \big\{\mathbf{z} \in \mathbf{R}^n \mid \mathbf{z}^T\mathbf{x} \leq 1 \text{ for all } \mathbf{x} \in S\big\}$.*

Lemma 8 *If $C \subseteq \mathbf{R}^n$ is closed and convex and $\mathbf{0} \in C$, then $(C^\circ)^\circ = C$.*

The proof uses the separating hyperplane theorem.

Theorem 9 (Representation of polytopes) *A polytope is a bounded polyhedron.*

The proof is based on polars and Lemma 8.

4 Optimality conditions

Consider the primal-dual pair in standard form,

$$\begin{array}{llll} \text{minimize} & \mathbf{c}^T\mathbf{x} & \text{maximize} & \mathbf{b}^T\mathbf{y} \\ \text{subject to} & A\mathbf{x} = \mathbf{b} & \text{subject to} & A^T\mathbf{y} \le \mathbf{c} \\ & \mathbf{x} \ge \mathbf{0} & & \end{array} \qquad (5,\ 6)$$

where $A \in \mathbf{R}^{m\times n}$ and $m < n$. Let $\mathcal{F}(P) = \{\mathbf{x} \in \mathbf{R}^n \mid A\mathbf{x} \le \mathbf{b},\ \mathbf{x} \ge \mathbf{0}\}$, $\mathcal{F}(D) = \{\mathbf{y} \in \mathbf{R}^m \mid A^T\mathbf{y} \le \mathbf{c}\}$, and $A = \begin{bmatrix}\mathbf{a}_1 & \dots & \mathbf{a}_n\end{bmatrix}$.

How to tell whether an $\mathbf{x} \in \mathcal{F}(P)$ is optimal? This section gives three answers.

4.1 Answer 1: $\mathbf{c}^T\mathbf{x} = \mathbf{b}^T\mathbf{y}$ for a $\mathbf{y} \in \mathcal{F}(D)$

Theorem 10 (Farkas' lemma) *Let $A \in \mathbf{R}^{m\times n}$ and $\mathbf{b} \in \mathbf{R}^m$. Exactly one of the following holds:*

(1) *There exists an $\mathbf{x} \in \mathbf{R}^n$ such that $A\mathbf{x} = \mathbf{b}$ and $\mathbf{x} \ge \mathbf{0}$.*

(2) *There exists a $\mathbf{y} \in \mathbf{R}^m$ such that $A^T\mathbf{y} \ge \mathbf{0}$ and $\mathbf{b}^T\mathbf{y} < 0$.*

The proof of Farkas' lemma uses the separating hyperplane theorem.

Theorem 11 (Farkas' lemma′) *Let $A \in \mathbf{R}^{m\times n}$ and $\mathbf{b} \in \mathbf{R}^m$. Exactly one of the following holds:*

(1′) *There exists an $\mathbf{x} \in \mathbf{R}^n$ such that $A\mathbf{x} \le \mathbf{b}$.*

(2′) *There exists a $\mathbf{y} \in \mathbf{R}^m$ such that $A^T\mathbf{y} = \mathbf{0}$, $\mathbf{b}^T\mathbf{y} = -1$ and $\mathbf{y} \ge \mathbf{0}$.*

The following is equivalent to (2′):

(2″) *There exists a $\mathbf{y} \in \mathbf{R}^m$ such that $A^T\mathbf{y} = \mathbf{0}$, $\mathbf{b}^T\mathbf{y} < 0$ and $\mathbf{y} \ge \mathbf{0}$.*

Farkas' lemma′ can be derived from Farkas' lemma.

Theorem 12 (Strong duality) *There are four possibilities:*

(1) *Both the primal and the dual are infeasible.*

(2) *The primal is infeasible and the dual is unbounded.*

(3) *The primal is unbounded and the dual is infeasible.*

(4) *Both the primal and the dual are feasible and their optimal values are equal.*

The proof of strong duality uses both versions of Farkas' lemma.

4.2 Answer 2: complementary slackness holds for a $\mathbf{y} \in \mathcal{F}(D)$

Definition 12 (Complementary slackness) *The vectors $\mathbf{x} \in \mathcal{F}(P)$ and $\mathbf{y} \in \mathcal{F}(D)$ obey the complementary slackness conditions if for all $j \in \{1, \ldots, n\}$, either $x_j = 0$ or $\mathbf{a}_j^T \mathbf{y} = c_j$.*

Lemma 13 *The vectors $\mathbf{x} \in \mathcal{F}(P)$ and $\mathbf{y} \in \mathcal{F}(D)$ are optimal if and only if the complementary slackness conditions hold.*

Proof: Let $\mathbf{x} \in \mathcal{F}(P)$ and $\mathbf{y} \in \mathcal{F}(D)$. By strong duality, $\mathbf{x}$ and $\mathbf{y}$ are optimal if and only if $\mathbf{c}^T\mathbf{x} = \mathbf{b}^T\mathbf{y}$. But $\mathbf{b}^T\mathbf{y} = \mathbf{y}^T\mathbf{b} = \mathbf{y}^T(A\mathbf{x}) = (A^T\mathbf{y})^T\mathbf{x}$, so $\mathbf{x}$ and $\mathbf{y}$ are optimal if and only if

$$\begin{aligned} & \mathbf{c}^T\mathbf{x} = (A^T\mathbf{y})^T\mathbf{x} \\ \iff \quad & (\mathbf{c} - A^T\mathbf{y})^T\mathbf{x} = 0 \\ \iff \quad & \sum_{j=1}^{n} (c_j - \mathbf{a}_j^T\mathbf{y})x_j = 0. \end{aligned}$$

Since $\mathbf{x} \in \mathcal{F}(P)$ and $\mathbf{y} \in \mathcal{F}(D)$, $x_j \geq 0$ and $\mathbf{a}_j^T\mathbf{y} \leq c_j$ for all $j \in \{1, \ldots, n\}$, so each term is nonnegative. The sum is therefore zero if and only if each term is zero, *i.e.*, for all $j \in \{1, \ldots, n\}$, either $x_j = 0$ or $\mathbf{a}_j^T\mathbf{y} = c_j$. □

4.3 Answer 3: the verifying y is in $\mathcal{F}(D)$

Lemma 14 *A vector $\mathbf{x} \in \mathcal{F}(P)$ is a vertex of $\mathcal{F}(P)$ if and only if $\{\mathbf{a}_j \mid x_j > 0\}$ is linearly independent.*

The proof uses the fact that if an optimal solution exists, then an optimal vertex exists (Lemma 5). It also uses the equivalence of vertices and basic feasible solutions.

Definition 13 (Basis) *A set $B \subseteq \{1, \ldots, n\}$ is a basis if $|B| = m$ and $\{\mathbf{a}_j \mid j \in B\}$ is linearly independent.*

It's also useful to define the set $N = \{1, \ldots, n\} - B$.

Given a basis B, the objective and constraints can be decomposed into their basic and nonbasic components by permuting columns of A and elements of $\mathbf{x}, \mathbf{b}$, and $\mathbf{c}$ such that

$$\mathbf{c}^T\mathbf{x} = \begin{bmatrix} \mathbf{c}_B^T & \mathbf{c}_N^T \end{bmatrix} \begin{bmatrix} \mathbf{x}_B \\ \mathbf{x}_N \end{bmatrix} = \mathbf{c}_B^T\mathbf{x}_B + \mathbf{c}_N^T\mathbf{x}_N$$

$$A\mathbf{x} = \begin{bmatrix} A_B & A_N \end{bmatrix} \begin{bmatrix} \mathbf{x}_B \\ \mathbf{x}_N \end{bmatrix} = A_B\mathbf{x}_B + A_N\mathbf{x}_N = \mathbf{b}$$

where $\mathbf{a}_j$ is a column of $A_B \in \mathbf{R}^{m \times m}$ if and only if $j \in B$. Similarly, $\mathbf{a}_j$ is a column of $A_N \in \mathbf{R}^{m \times (n-m)}$ if and only if $j \in N$. By definition of basis, A_B^{-1} exists.

Lemma 15 *For any basis B, there is a unique corresponding basic solution to $A\mathbf{x} = \mathbf{b}$.*

The proof involves setting $\mathbf{x}_N = \mathbf{0}$ and $\mathbf{x}_B = A_B^{-1}\mathbf{b}$.

Note that the basic solution corresponding to B is not necessarily feasible.

Definition 14 (Degenerate) *A basic solution $\mathbf{x} \in \mathcal{F}(P)$ is degenerate if there exists a $j \in B$ such that $x_j = 0$.*

Lemma 16 *Let B be a basis with corresponding basic solution $\mathbf{x}$.*

(1) *If there exists a $\mathbf{y} \in \mathcal{F}(D)$ such that $A_B^T\mathbf{y} = \mathbf{c}_B$, then $\mathbf{x}$ is optimal.*

(2) *If $\mathbf{x}$ is nondegenerate and optimal, then there exists a $\mathbf{y} \in \mathcal{F}(D)$ such that $A_B^T\mathbf{y} = \mathbf{c}_B$.*

The proof uses complementary slackness.

Definition 15 (Verifying y) *For a basis B, $\mathbf{y} = A_B^{-T}\mathbf{c}_B$ is the verifying $\mathbf{y}$.*

It's called 'verifying' because if $\mathbf{y} = A_B^{-T}\mathbf{c}_B \in \mathcal{F}(D)$, then $\begin{bmatrix}\mathbf{x}_B \\ \mathbf{x}_N\end{bmatrix} = \begin{bmatrix}A_B^{-1}\mathbf{b} \\ \mathbf{0}\end{bmatrix}$ is optimal.

Note that $A_B^{-T} = (A_B^{-1})^T = (A_B^T)^{-1}$.

5 The simplex method

Consider again the standard form primal (5) and dual (6).

5.1 Reformulation

Definition 16 (Reduced cost) *For any* $\mathbf{y} \in \mathbf{R}^m$*, the reduced cost with respect to* $\mathbf{y}$ *is* $\bar{\mathbf{c}} = \mathbf{c} - A^T\mathbf{y}$.

- A vector $\mathbf{y} \in \mathbf{R}^m$ is dual feasible if and only if the reduced cost with respect to $\mathbf{y}$ is nonnegative:
$$\bar{\mathbf{c}} = \mathbf{c} - A^T\mathbf{y} \geq \mathbf{0} \quad \iff \quad A^T\mathbf{y} \leq \mathbf{c}.$$
- The basic component of the reduced cost with respect to the verifying $\mathbf{y}$ is zero:
$$\bar{\mathbf{c}} = \mathbf{c} - A^T\mathbf{y} = \mathbf{c} - A^T A_B^{-T}\mathbf{c}_B$$
$$\iff \quad \begin{bmatrix} \bar{\mathbf{c}}_B \\ \bar{\mathbf{c}}_N \end{bmatrix} = \begin{bmatrix} \mathbf{c}_B \\ \mathbf{c}_N \end{bmatrix} - \begin{bmatrix} A_B^T \\ A_N^T \end{bmatrix} A_B^{-T}\mathbf{c}_B = \begin{bmatrix} \mathbf{0} \\ \mathbf{c}_N - (A_B^{-1}A_N)^T\mathbf{c}_B \end{bmatrix}.$$

Lemma 17 *Let* $\bar{\mathbf{c}}$ *be the reduced cost for some* $\mathbf{y} \in \mathbf{R}^m$*. Then* $\mathbf{x}$ *is optimal for* $\min\{\bar{\mathbf{c}}^T\mathbf{x} \mid A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ *if and only if* $\mathbf{x}$ *is optimal for* $\min\{\mathbf{c}^T\mathbf{x} \mid A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$.

Proof: The LPs have the same feasible region, so consider their objectives:
$$\bar{\mathbf{c}}^T\mathbf{x} = (\mathbf{c} - A^T\mathbf{y})^T\mathbf{x} = \mathbf{c}^T\mathbf{x} - \mathbf{y}^T(A\mathbf{x}) = \mathbf{c}^T\mathbf{x} - \mathbf{y}^T\mathbf{b}.$$
Since $\mathbf{y}^T\mathbf{b}$ is constant for a given $\mathbf{y}$, the LPs have the same optimal solution. □

Lemma 17 allows reformulation of the primal as the equivalent LP
$$\begin{array}{ll} \text{minimize} & \bar{\mathbf{c}}_N^T\mathbf{x}_N \\ \text{subject to} & \bar{A}\mathbf{x}_N \leq \bar{\mathbf{b}} \\ & \mathbf{x}_N \geq \mathbf{0} \end{array} \tag{7}$$
where $\bar{A} = A_B^{-1}A_N$, $\bar{\mathbf{b}} = A_B^{-1}\mathbf{b}$, and $\bar{\mathbf{c}}_N = \mathbf{c}_N - \bar{A}^T\mathbf{c}_B$ is the nonbasic component of the reduced cost with respect to $\mathbf{y} = A_B^{-T}\mathbf{c}_B$. If $\mathbf{x}_N$ is optimal for (7), then $\mathbf{x} = \begin{bmatrix}\mathbf{x}_B^T & \mathbf{x}_N^T\end{bmatrix}^T$ and $\mathbf{y}$, where $\mathbf{x}_B = \bar{\mathbf{b}} - \bar{A}\mathbf{x}_N$, are optimal for (5) and (6).

5.2 Algorithm

Given[1] an initial basis B and corresponding basic feasible solution $\mathbf{x} = \begin{bmatrix} \mathbf{x}_B \\ \mathbf{x}_N \end{bmatrix} = \begin{bmatrix} A_B^{-1}\mathbf{b} \\ \mathbf{0} \end{bmatrix}$, the simplex method consists of four steps.

[1] More on producing an initial feasible solution and corresponding basis in §5.4.

1. **Check for optimality**

   Set $\mathbf{y} = A_B^{-T}\mathbf{c}_B$ and $\bar{\mathbf{c}} = \mathbf{c} - A^T\mathbf{y}$, so $\bar{\mathbf{c}}_B = \mathbf{0}$ and $\bar{\mathbf{c}}_N = \mathbf{c}_N - \bar{A}^T\mathbf{c}_B$. If $\bar{\mathbf{c}} \geq \mathbf{0}$, then $\mathbf{x}$ is optimal. Otherwise, there exists a $j \in N$ such that $\bar{c}_j < 0$. Pick[2] one such j.

2. **Check for unboundedness**

   Since $\bar{c}_j < 0$, increasing x_j from zero will improve the objective $\bar{\mathbf{c}}_N^T\mathbf{x}_N$. The upper bound on x_j is given by

$$\begin{aligned} & \bar{A}\mathbf{x}_N \leq \bar{\mathbf{b}} \\ \iff \quad & \begin{bmatrix} \bar{\mathbf{a}}_1 & \dots & \bar{\mathbf{a}}_{n-m} \end{bmatrix} \mathbf{x}_N \leq \bar{\mathbf{b}} \\ \iff \quad & x_j\bar{\mathbf{a}}_j \leq \bar{\mathbf{b}} \\ \iff \quad & x_j\bar{a}_{ij} \leq \bar{b}_i \quad \text{for all} \quad i \in B \end{aligned}$$

   where the third line follows from $x_i = 0$ for all $i \neq j$. Note that the row indices of $\bar{A} = A_B^{-1}A_N$ and $\bar{\mathbf{b}} = A_B^{-1}\mathbf{b}$ are the row indices of A_B^{-1}, which are the elements of B. If $\bar{\mathbf{a}}_j \leq \mathbf{0}$, then x_j can be increased arbitrarily while remaining feasible, *i.e.*, the primal is unbounded.

3. **Ratio test**

   If $\bar{\mathbf{a}}_j \nleq \mathbf{0}$, then there exists at least one $i \in B$ such that $\bar{a}_{ij} > 0$. The first such i for which the constraint $x_j\bar{a}_{ij} \leq \bar{b}_i$ becomes binding determines the upper bound on x_j. Set[3]

$$\varepsilon = \min_{i:\bar{a}_{ij}>0} \frac{\bar{b}_i}{\bar{a}_{ij}} \quad \text{and} \quad i^* = \operatorname*{argmin}_{i:\bar{a}_{ij}>0} \frac{\bar{b}_i}{\bar{a}_{ij}}.$$

4. **Pivot**

   Set $\hat{\mathbf{x}}_N = \mathbf{x}_N + \varepsilon\mathbf{e}_j = \varepsilon\mathbf{e}_j$ and $\hat{\mathbf{x}}_B = \bar{\mathbf{b}} - \bar{A}\hat{\mathbf{x}}_N = \mathbf{x}_B - \varepsilon\bar{\mathbf{a}}_j$ (so $x_{i^*} = 0$). Update the basis to $\hat{B} = B \cup \{j\} - \{i^*\}$ and $\hat{N} = N \cup \{i^*\} - \{j\}$.

[2] More on choosing a j in §5.6.

[3] Multiple i may attain the minimum. More on resolving this ambiguity in §5.6.

5.3 Convergence

Claim 18 (Nonincreasing objective) $\mathbf{c}^T\hat{\mathbf{x}} \leq \mathbf{c}^T\mathbf{x}$

Proof: By Lemma 17, it suffices to show that $\bar{\mathbf{c}}^T\hat{\mathbf{x}} \leq \bar{\mathbf{c}}^T\mathbf{x}$. Because $\bar{\mathbf{c}}_B = \mathbf{0}$ and $\hat{\mathbf{x}}_N = \varepsilon\mathbf{e}_j$,

$$\bar{\mathbf{c}}^T\hat{\mathbf{x}} = \bar{\mathbf{c}}_B^T\hat{\mathbf{x}}_B + \bar{\mathbf{c}}_N^T\hat{\mathbf{x}}_N = \bar{\mathbf{c}}_N^T(\varepsilon\mathbf{e}_j) = \bar{c}_j\varepsilon \leq 0$$

since $\bar{c}_j < 0$ and $\varepsilon \geq 0$. But $\mathbf{x}_N = \mathbf{0}$, so $\bar{\mathbf{c}}^T\mathbf{x} = \bar{\mathbf{c}}_B^T\mathbf{x}_B + \bar{\mathbf{c}}_N^T\mathbf{x}_N = 0$. Thus, $\bar{\mathbf{c}}^T\hat{\mathbf{x}} \leq \bar{\mathbf{c}}^T\mathbf{x}$. □

Claim 19 (Decreasing objective for nondegenerate solutions) *If* $\mathbf{x}$ *is nondegenerate, then* $\mathbf{c}^T\hat{\mathbf{x}} < \mathbf{c}^T\mathbf{x}$.

Proof: Let $\mathbf{x}$ be nondegenerate, *i.e.*, let $\mathbf{x}_B > \mathbf{0}$. Since $\mathbf{x}_B = A_B^{-1}\mathbf{b} = \bar{\mathbf{b}}$, it's also true that $\bar{\mathbf{b}} > \mathbf{0}$. Thus, $b_i/\bar{a}_{ij} > 0$ whenever $\bar{a}_{ij} > 0$, so $\varepsilon > 0$. □

Claim 20 *The set* $\hat{B}$ *is a basis.*

Proof: Since $\hat{B} = B \cup \{j\} - \{i^*\}$, $A_{\hat{B}}$ and A_B differ only by one column. Without loss of generality, we can construct $A_{\hat{B}}$ by overwriting $\mathbf{a}_{i^*}^B$, the $(i^*)^{\text{th}}$ column of A_B, with $\mathbf{a}_j$, the j^{th} column of A_N:

$$\begin{aligned} A_{\hat{B}} &= \begin{bmatrix}\mathbf{a}_1^B & \dots & \mathbf{a}_{i^*-1}^B & \mathbf{a}_j & \mathbf{a}_{i^*+1}^B & \dots & \mathbf{a}_m^B\end{bmatrix} \\ &= A_B\begin{bmatrix}\mathbf{e}_1 & \dots & \mathbf{e}_{i^*-1} & A_B^{-1}\mathbf{a}_j & \mathbf{e}_{i^*+1} & \dots & \mathbf{e}_m\end{bmatrix} \\ &= A_B\begin{bmatrix}\mathbf{e}_1 & \dots & \mathbf{e}_{i^*-1} & \bar{\mathbf{a}}_j & \mathbf{e}_{i^*+1} & \dots & \mathbf{e}_m\end{bmatrix} \\ &= A_B E \end{aligned}$$

where $A_B^{-1}\mathbf{a}_j = \bar{\mathbf{a}}_j$ follows from $j \in N$ and $\bar{A} = A_B^{-1}A_N$. The matrix E is called an **eta matrix**.

Note that $(\bar{\mathbf{a}}_j)_{i^*} = \bar{a}_{i^*j} > 0$ from the ratio test, and $(\mathbf{e}_k)_i = 0$ for all $k \neq i$, so the set $\{\mathbf{e}_1, \dots, \mathbf{e}_{i^*-1}, \bar{\mathbf{a}}_j, \mathbf{e}_{i^*+1}, \dots, \mathbf{e}_m\}$ is linearly independent and E is full rank. Pre-multiplying by A_B, which is full rank since B is a basis, preserves rank. Thus, $A_{\hat{B}}$ is full rank and $\hat{B}$ is a basis. □

Claim 21 *The basic feasible solution corresponding to* $\hat{B}$ *is* $\hat{\mathbf{x}}$.

Because there are only $\binom{n}{m}$ possible bases, the previous four claims establish that – as long as no degenerate solutions are encountered – the simplex method converges to an optimum in finite time.

5.4 Finding an initial basic feasible solution

The algorithm in §5.2 requires an initial basic feasible solution and corresponding basis. To compute this solution and basis (or provide a certificate of infeasibiilty), we start by multiplying by -1 any row i of $A\mathbf{x} = \mathbf{b}$ such that $b_i < 0$. This ensures that $\mathbf{b} \geq \mathbf{0}$. We then introduce artificial variables $\mathbf{z} \in \mathbf{R}^m$ and consider the LP

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T \mathbf{z} \\ \text{subject to} & A\mathbf{x} + I\mathbf{z} = \mathbf{b} \\ & \mathbf{x}, \mathbf{z} \geq \mathbf{0} \end{array} \tag{8}$$

which can be written in standard form $\min\{\tilde{\mathbf{c}}^T \tilde{\mathbf{x}} \mid \tilde{A}\tilde{\mathbf{x}} = \tilde{\mathbf{b}},\ \tilde{\mathbf{x}} \geq \mathbf{0}\}$ by setting

$$\tilde{\mathbf{x}} = \begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix}, \quad \tilde{A} = \begin{bmatrix} A & I \end{bmatrix}, \quad \tilde{\mathbf{b}} = \mathbf{b}, \quad \tilde{\mathbf{c}} = \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix}.$$

An initial basis for (8) is $\tilde{A}_B = I$, $\tilde{A}_N = A$ with corresponding basic feasible solution $\tilde{\mathbf{x}}_N = \mathbf{0}$, $\tilde{\mathbf{x}}_B = \tilde{A}_B^{-1}\tilde{\mathbf{b}} = \tilde{\mathbf{b}} \geq \mathbf{0}$. We can therefore run the simplex method on (8), which will converge to an optimum $\tilde{\mathbf{x}}^*$. There are two possible outcomes:

1. $\tilde{\mathbf{c}}^T \tilde{\mathbf{x}}^* > 0$, so the original primal (5) is infeasible.

2. $\tilde{\mathbf{c}}^T \tilde{\mathbf{x}}^* = 0 \implies \mathbf{1}^T \mathbf{z}^* = 0$, so the original primal is feasible. There are two subcases:

   (a) All artificial variables are nonbasic, so $A_B = \tilde{A}_B$, $\mathbf{x}_N = \mathbf{0}$, and $\mathbf{x}_B = A_B^{-1}\mathbf{b}$ give an initial basis and corresponding basic feasible solution for (5).

   (b) Some artificial variables are basic. There are two sub-cases:

      i. For $i \in B$ corresponding to an artificial variable z_i, there exists a $j \in N$ corresponding to a real variable x_j such that $\bar{a}_{ij} \neq 0$. In this case, we pivot to $\hat{B} = B \cup \{i\} - \{j\}$. It can be shown that $\hat{B}$ is a basis with corresponding basic solution $\tilde{\mathbf{x}}$.

      ii. For $i \in B$ corresponding to an artificial variable z_i, $\bar{a}_{ij} = 0$ for all $j \in N$ corresponding to real variables x_j. This implies that A is not full (row) rank, so delete the linearly dependent row(s) and continue.

   Repeat until all artificial variables are nonbasic, at which point $A_B = \tilde{A}_{\hat{B}}$, $\mathbf{x}_N = \mathbf{0}$, and $\mathbf{x}_B = A_B^{-1}\mathbf{b}$ give an initial basis and corresponding basic feasible solution for (5).

Producing an initial basis and corresponding basic feasible solution is called **Phase 1** of the simplex method. Finding an optimal solution to the original LP is called **Phase 2**.

5.5 Complexity of a pivot

The operations associated with an iteration of Phase 2 of the simplex method are:

1. Solve $A_B\mathbf{x}_B = \mathbf{b}$ for $\mathbf{x}_B$.

2. Solve $A_B^{-T}\mathbf{y} = \bar{\mathbf{c}}$ for $\mathbf{y}$.

3. Compute $\bar{\mathbf{c}} = \mathbf{c} - A^T\mathbf{y}$, check whether $\bar{\mathbf{c}} \geq \mathbf{0}$ and if not, pick a j such that $\bar{c}_j < 0$.

4. Solve $A_B\bar{\mathbf{a}}_j = \mathbf{a}_j$ for $\bar{\mathbf{a}}_j$.

5. Find $\varepsilon = \min_{i:\bar{a}_{ij}>0} \bar{b}_i/\bar{a}_{ij}$ and the corresponding index i^*.

6. Update solution: $\hat{x}_j = \varepsilon$, $\hat{\mathbf{x}}_B = \mathbf{x}_B - \varepsilon\bar{\mathbf{a}}_j$.

Steps 1, 2 and 4 require solving m equations for m unknowns, which takes $\mathcal{O}(m^3)$ time if the coefficient matrix is dense, less if sparsity can be exploited. Steps 5 and 6 are checking or updating m-vectors, so each is $\mathcal{O}(m)$ time. For each nonbasic component of $\bar{\mathbf{c}}$ in step 3, an inner product must be taken between $\mathbf{y} \in \mathbf{R}^m$ and a column of A ($\mathcal{O}(m)$ time), along with a subtraction. Computing all $n - m$ nonbasic components of $\bar{\mathbf{c}}$ takes $\mathcal{O}(nm)$ time. Therefore, a naive implementation of the simplex method takes $\mathcal{O}(m^3 + nm)$ time per iteration.

Computation time can be cut down in steps 1, 2 and 4 by using the eta matrices defined in the proof of Claim 20. Let $B_0, \ldots, B_k$ be the bases at iterations $0, \ldots, k$ with corresponding eta matrices $E_1, \ldots, E_k$. Without loss of generality, suppose that $A_{B_0} = I$ (if not, we can perform elementary row operations on $A\mathbf{x} = \mathbf{b}$ until it is). Then

$$\begin{aligned} A_{B_1} &= A_{B_0}E_1 = E_1 \\ &\vdots \\ A_{B_k} &= A_{B_{k-1}}E_{k-1} = E_0 \cdots E_{k-1}. \end{aligned}$$

Solving $A_{B_1}\mathbf{x}_{B_1} = E_1\mathbf{x}_{B_1} = \mathbf{b}$ by back-substitution takes $\mathcal{O}(m)$ time. At iteration k, $A_{B_k}\mathbf{x}_{B_k} = E_1 \cdots E_k\mathbf{x}_{B_k} = \mathbf{b}$ can be solved by solving k equations of the form $E_i\mathbf{x}_{B_i} = \mathbf{b}$, which takes $\mathcal{O}(km)$ time in total. Steps 2 and 4 can be expedited similarly.

With this modification, an iteration of the simplex method takes $\mathcal{O}(km+nm)$ time. Once k gets large, common practice is to recompute $A_{B_k}^{-1}$ and row reduce $A\mathbf{x} = \mathbf{b}$ until $A_{B_k} = I$, effectively resetting k to zero.

5.6 Pivot rules

Some of the $\mathcal{O}(nm)$ time required to compute $\bar{\mathbf{c}}$ can be saved by using an efficient rule for choosing the j that leaves the basis. In general, choosing such a pivot rule requires balancing computational efficiency in the current iteration against progress of the overall algorithm.

Some candidate pivot rules:

1. Compute $\bar{\mathbf{c}}$ component-wise and stop at the first $\bar{c}_j < 0$.
2. Find the j that makes $\bar{c}_j$ most negative.
3. Find the j that gives the biggest decrease in the objective $\bar{\mathbf{c}}_N^T \mathbf{x}_N$.
4. Find the **steepest edge**, *i.e.*, the j that gives the biggest decrease in objective per change in basic solution:
$$\frac{\Delta(\text{objective})}{\Delta(\text{basic solution})} = \frac{\bar{\mathbf{c}}_N^T \hat{\mathbf{x}}_N - \bar{\mathbf{c}}_N^T \mathbf{x}_N}{\|\hat{\mathbf{x}}_B - \mathbf{x}_B\|_2} = \frac{\bar{\mathbf{c}}_N^T (\varepsilon \mathbf{e}_j)}{\|-\varepsilon \bar{\mathbf{a}}_j\|_2} = \frac{\bar{c}_j}{\|\bar{\mathbf{a}}_j\|_2}.$$

The steepest edge rule works very well in practice. Another common method is to maintain a **pivot pool**, *i.e.*, a list of indices that corresponded to negative reduced costs in recent iterations, and to search over these indices first. This heuristic is based on the idea that $\bar{\mathbf{c}}$ is fairly stable across iterations.

5.7 Degeneracy and cycling

In §5.3, we showed that if no degenerate solutions are encountered, then the simplex method converges in finite time. If degenerate solutions are encountered, however, this guarantee is lost: the algorithm can **cycle** repeatedly through a sequence of bases.

One anti-cycling technique used in some commercial codes is perturbation: the algorithm uses an efficient pivot rule until a cycle is encountered, then perturbs $\mathbf{b}$ by some small amount in order to force nondegeneracy. Another technique is to switch to Bland's rule when a cycle is encountered.

Definition 17 (Bland's rule) *Under Bland's rule, the smallest index in* $\{j \in N \mid \bar{c}_j < 0\}$ *leaves the basis and the smallest index in* $\left\{i \in B \mid \bar{a}_{ij} > 0,\ \bar{b}_i/\bar{a}_{ij} = \varepsilon\right\}$ *enters.*

Theorem 22 (Termination with Bland's rule) *If the simplex method uses Bland's rule, then it converges to an optimum in finite time (with no cycling).*

5.8 Number of pivots

Using Bland's rule, the simplex method is guaranteed to converge in no more than $\binom{n}{m}$ pivots. This factorial upper bound is very loose.

In 1973, Victor Klee and George Minty showed that the worst-case number of pivots under the "minimum $\bar{c}_j$" pivot rule is at least exponential in n. Using an n-dimensional **Klee-Minty cube** (a particular perturbation to the unit hypercube) as the feasible region, they showed that the simplex method visits all 2^n vertices. Thus, a lower bound on the worst-case number of pivots with this pivot rule is $2^n - 1$. As of 2014, no other pivot rule has improved upon this lower bound.

Another approach to finding a lower bound on the number of pivots is to bound the diameter of a polytope $P \subseteq \mathbf{R}^n$ with m constraints. Let $d(\mathbf{x}, \mathbf{y})$ be the minimum number of nondegenerate pivots to move between vertices $\mathbf{x}$ and $\mathbf{y}$ of P. Define the diameter of P to be the maximum $d(\mathbf{x}, \mathbf{y})$ over all vertices $\mathbf{x}, \mathbf{y}$ of P. Let $\Delta(n, m)$ be the maximum diameter over all polytopes $P \subseteq \mathbf{R}^n$ with m constraints. By construction, the simplex method with any pivot rule, started from the "worst vertex" of the "worst polytope" in $\mathbf{R}^n$, will take at least $\Delta(n, m)$ pivots. In 2014, Michael Todd showed that $\Delta(n, m) \leq (m - n)^{\log_2 n}$.

In practice, the simplex method seems to take only $\mathcal{O}(m)$ pivots. Explaining this observed convergence rate, which is much faster than the best theoretical bound, is still an active research area.

5.9 Varieties of the simplex method

- The **revised simplex method** is the one described in §5.2. It maintains primal feasibility and complementary slackness while working toward dual feasibility.

- The **standard simplex method** is similar to the revised simplex method, but it also maintains a tableau that facilitates solving small problems by hand.

- The **capacitated simplex method** applies to problems with constraints of the form $\boldsymbol{\ell} \leq \mathbf{x} \leq \mathbf{u}$. Although these problems can be written in standard form, they can be solved more efficiently by modifying the simplex method. The capacitated simplex method keeps track of the basis B, the set L of indices of variables set to their lower bound, and the set U of indices of variables set to their upper bound.

- The **dual simplex method** maintains dual feasibility and complementary slackness while working toward primal feasibility. The dual simplex method is the most common implementation of the simplex algorithm, for three main reasons:

  1. It's often easier to produce an initial basic feasible solution for the dual than for the primal. This is because $\mathbf{c}$ is often nonnegative, so $\mathbf{y} = \mathbf{0}$ is feasible.
  2. The dual simplex method seems to encounter degenerate solutions less frequently.
  3. The dual simplex method is useful for **warm starts**, where an LP is initialized with the optimal solution to a related LP. The primal simplex method can be warm started if $\mathbf{c}$ is changed. The dual simplex method can be warm started if $\mathbf{b}$ is changed or if a new constraint $\boldsymbol{\alpha}_{m+1}^T \mathbf{x} = b_{m+1}$ is added, in which case setting the additional dual variable y_{m+1} to zero produces an initial basic feasible solution. This incremental addition of constraints is common practice when solving large-scale LPs or integer programs.

5.10 Sensitivity analysis

How do the solutions $\mathbf{x}$ and $\mathbf{y}$ to the standard form primal (5) and dual (6) respond to perturbations in the data A, $\mathbf{b}$, and $\mathbf{c}$?

5.10.1 Changes to b

Suppose $\mathbf{b} \to \mathbf{b} + \delta \mathbf{e}_i$. Then $\mathbf{y} = A_B^{-T}\mathbf{c}_B$ is unchanged, so dual feasibility is preserved. Setting $\mathbf{x}_N = \mathbf{0}$ and $\mathbf{x}_B = A_B^{-1}(\mathbf{b} + \delta \mathbf{e}_i)$ also preserves complementary slackness: for all $j \in N$, $x_j = 0$, and for all $j \in B$, $\mathbf{a}_j^T \mathbf{y} = c_j$:

$$A^T \mathbf{y} = \begin{bmatrix} A_B^T \\ A_N^T \end{bmatrix} A_B^{-T} \mathbf{c}_B = \begin{bmatrix} \mathbf{c}_B \\ \bar{A}^T \mathbf{c}_B \end{bmatrix}.$$

If $\mathbf{x}_B \geq \mathbf{0}$, then primal feasibility is also preserved, and $\mathbf{x}$ and $\mathbf{y}$ are optimal. In this case, the change in objective is

$$\Delta(\mathbf{c}^T \mathbf{x}) = \mathbf{c}_B^T \Delta \mathbf{x}_B = \mathbf{c}_B^T (\delta A_B^{-1} \mathbf{e}_i) = \delta (A_B^{-T} \mathbf{c}_B)^T \mathbf{e}_i = \delta \mathbf{y}^T \mathbf{e}_i = \delta y_i.$$

This gives the interesting interpretation of $y_i = \Delta(\mathbf{c}^T\mathbf{x})/\delta$ as the change in objective per unit change in the i^{th} constraint. This is why in the economics literature, the dual variables are called **shadow prices** or **marginal costs**.

5.10.2 Changes to c

Suppose $\mathbf{c} \to \mathbf{c} + \delta \mathbf{e}_j$. Then $\mathbf{x}_N = \mathbf{0}$ and $\mathbf{x}_B = A_B^{-1}\mathbf{b}$ are unchanged, so primal feasibility is preserved. To check complementary slackness and dual feasibility, consider two cases:

1. If $j \in N$, then $\mathbf{c}_B$ and $\mathbf{y} = A_B^{-T}\mathbf{c}_B$ are unchanged, so we still have $x_i = 0$ for all $i \in B$ and $y_i = c_i$ for all $i \in N$, *i.e.*, complementary slackness is preserved. If $\mathbf{a}_j^T \mathbf{y} \leq c_j$, then dual feasibility is also preserved, and $\mathbf{x}$ and $\mathbf{y}$ are optimal. The objective is unchanged since $j \in N \implies x_j = 0$.

2. If $j \in B$, then $\mathbf{y} \to A_B^{-T}(\mathbf{c}_B + \delta \mathbf{e}_j) = \mathbf{y} + \delta A_B^{-T} \mathbf{e}_j$. Both dual feasibility and complementary slackness need to be checked. The objective changes by δx_j.

5.10.3 Changes to A

Suppose $\mathbf{a}_j \to \mathbf{a}_j + \delta \mathbf{e}_i$. Consider two cases:

1. If $j \in N$, then $\mathbf{x}_N = \mathbf{0}$, $\mathbf{x}_B = A_B^{-1}\mathbf{b}$ and $\mathbf{y} = A_B^{-T}\mathbf{c}_B$ are unchanged, so primal feasibility and complementary slackness are preserved and the objective is unchanged. Dual feasibility is preserved if $\mathbf{a}_j^T \mathbf{y} + \delta y_i \leq c_j$.

2. If $j \in B$, then $A_B \to A_B + \delta \mathbf{e}_i \mathbf{e}_j^T$, so we need to check whether A_B becomes singular. If not, then both $\mathbf{x}_B = A_B^{-1}\mathbf{b}$ and $\mathbf{y} = A_B^{-T}\mathbf{c}_B$ change, so both primal and dual feasibility need to be checked. If $\mathbf{x}$ and $\mathbf{y}$ are feasible, then they're optimal because complementary slackness is preserved.

5.11 Solving large-scale linear programs

Some linear programs have too many variables to be solved directly with the simplex method in a reasonable amount of time. A common method for such problems is to decompose them into a master problem and a number of subproblems. At each iteration of the simplex method on the master problem, the checks for optimality and unboundedness and the ratio test are conducted by solving the subproblems. This method is called **column generation**.

5.11.1 The cutting stock problem

A classic example of a problem amenable to column generation, called the **cutting stock problem**, comes from the paper industry. The problem is to cut smaller rolls ("finals") from a big roll of paper (a "raw") of width W. For each $i \in \{1, \ldots, m\}$, there is a demand for b_i finals of width $s_i \leq W$. The objective is to cut the smallest number of raws while satisfying demand.

With the proper formulation, this integer program can be solved by LP relaxation and column generation. Define the **pattern** $\mathbf{a}_j \in \mathbf{R}^m$ such that $(\mathbf{a}_j)_i = a_{ij}$ is the number of finals of width s_i cut from a raw. The patterns of interest in this problem are feasible (they satisfy $\mathbf{a}_j \geq \mathbf{0}$ and $\mathbf{1}^T\mathbf{a}_j \leq W$) and maximal (their sum is as close as possible to W).

As an example, consider problem data $W = 10$, $m = 3$, $s_1 = 6$, $s_2 = 3$, $s_3 = 2$. The maximal, feasible patterns are

$$\mathbf{a}_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \ \mathbf{a}_2 = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix}, \ \mathbf{a}_3 = \begin{bmatrix} 0 \\ 3 \\ 0 \end{bmatrix}, \ \mathbf{a}_4 = \begin{bmatrix} 0 \\ 2 \\ 2 \end{bmatrix}, \ \mathbf{a}_5 = \begin{bmatrix} 0 \\ 1 \\ 3 \end{bmatrix}, \ \mathbf{a}_6 = \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix}.$$

Let n be the number of maximal patterns and x_j be the number of raws cut into pattern $\mathbf{a}_j$. Then the cutting stock problem can be written as

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T\mathbf{x} \\ \text{subject to} & A\mathbf{x} \geq \mathbf{b} \\ & \mathbf{x} \in \mathbf{Z}_+^n \end{array} \tag{9}$$

where $A = \begin{bmatrix} \mathbf{a}_1 & \ldots & \mathbf{a}_n \end{bmatrix}$, $\mathbf{x} = \begin{bmatrix} x_1 & \ldots & x_n \end{bmatrix}^T$, $\mathbf{b} = \begin{bmatrix} b_1 & \ldots & b_m \end{bmatrix}^T$ and $\mathbf{Z}_+^n$ is the set of n-vectors with nonnegative, integer components.

The LP relaxation of (9) is

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T\mathbf{x} \\ \text{subject to} & A\mathbf{x} \geq \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{array}$$

which can be solved by column generation. Let $\mathbf{y} = A_B^{-T}\mathbf{c}_B$ be the verifying $\mathbf{y}$ at some iteration of the simplex method on the master problem. The subproblem that verifies optimality or produces a negative reduced cost is

$$\begin{array}{ll} \text{maximize} & \mathbf{y}^T\mathbf{a} \\ \text{subject to} & \mathbf{s}^T\mathbf{a} \leq W \\ & \mathbf{a} \in \mathbf{Z}_+^m \end{array}$$

where $\mathbf{s} = \begin{bmatrix} s_1 & \dots & s_m \end{bmatrix}^T$. This is called the **knapsack problem**. It can be solved by dynamic programming.

5.11.2 The Dantzig-Wolfe decomposition

A common application of column generation is the **Dantzig-Wolfe decomposition**, which applies to LPs of the form

$$\begin{array}{ll} \text{minimize} & \mathbf{c}_1^T\mathbf{x}_1 + \cdots + \mathbf{c}_m^T\mathbf{x}_m \\ \text{subject to} & \begin{bmatrix} A_{01} & \dots & A_{0m} \\ A_{11} & & \\ & \ddots & \\ & & A_{mm} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_m \end{bmatrix} = \begin{bmatrix} \mathbf{b}_0 \\ \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_m \end{bmatrix} \\ & \mathbf{x}_i \geq \mathbf{0}, \quad i \in \{1, \dots, m\} \end{array} \tag{10}$$

where $A_{ij} \in \mathbf{R}^{m_i \times n_j}$, $\mathbf{x}_j \in \mathbf{R}^{n_j}$, and $\mathbf{b}_i \in \mathbf{R}^{m_i}$. The method also requires that the subproblem

$$\begin{array}{ll} \text{minimize} & \tilde{\mathbf{c}}_i^T\mathbf{x}_i \\ \text{subject to} & A_{ii}\mathbf{x}_i = \mathbf{b}_i \\ & \mathbf{x}_i \geq \mathbf{0} \end{array} \tag{11}$$

be easy to solve for any $i \in \{1, \dots, m\}$ and any cost $\tilde{\mathbf{c}}_i \in \mathbf{R}^{n_i}$.

We assume that each polyhedron $P_i = \{\mathbf{x}_i \in \mathbf{R}^{n_i} \mid A_{ii}\mathbf{x}_i = \mathbf{b}_i, \mathbf{x}_i \geq \mathbf{0}\}$ is bounded. This assumption is unnecessary, but makes life easier by allowing any $\mathbf{x}_i \in P_i$ to be written as a convex combination of the vertices $\mathbf{v}_{i1}, \dots, \mathbf{v}_{iN_i}$ of P_i. That is, for any $\mathbf{x}_i \in P_i$, there exists a $\boldsymbol{\lambda}_i = \begin{bmatrix} \lambda_{i1} & \dots & \lambda_{iN_i} \end{bmatrix}^T$ satisfying $\mathbf{1}^T\boldsymbol{\lambda}_i = 1$ and $\boldsymbol{\lambda}_i \geq \mathbf{0}$ such that $\mathbf{x}_i = \sum_{j=1}^{N_i} \lambda_{ij}\mathbf{v}_{ij}$.

The last assumption allows LP (10) to be written as

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m \sum_{j=1}^{N_i} \lambda_{ij}\mathbf{c}_i^T\mathbf{v}_{ij} \\ \text{subject to} & \sum_{i=1}^m \sum_{j=1}^{N_i} \lambda_{ij}A_{0i}\mathbf{v}_{ij} = \mathbf{b}_0 \\ & \mathbf{1}^T\boldsymbol{\lambda}_i = 1, \quad i \in \{1, \dots, m\} \\ & \boldsymbol{\lambda}_i \geq \mathbf{0}, \quad i \in \{1, \dots, m\} \end{array}$$

or, equivalently,

$$\begin{array}{ll} \text{minimize} & \mathbf{c}^T\boldsymbol{\lambda} \\ \text{subject to} & \begin{bmatrix} A_{01}V_1 & \dots & A_{0m}V_m \\ \mathbf{1}_{N_1}^T & & \\ & \ddots & \\ & & \mathbf{1}_{N_m}^T \end{bmatrix} \boldsymbol{\lambda} = \begin{bmatrix} \mathbf{b}_0 \\ \mathbf{1}_m \end{bmatrix} \\ & \boldsymbol{\lambda} \geq \mathbf{0} \end{array} \tag{12}$$

where we define $N = \sum_{i=1}^{m} N_i$ and

$$\boldsymbol{\lambda} = \begin{bmatrix} \boldsymbol{\lambda}_1 \\ \vdots \\ \boldsymbol{\lambda}_m \end{bmatrix} \in \mathbf{R}^N, \quad V_i = \begin{bmatrix} \mathbf{v}_{i1} & \dots & \mathbf{v}_{iN_i} \end{bmatrix} \in \mathbf{R}^{n_i \times N_i}, \quad \mathbf{c} = \begin{bmatrix} V_1^T \mathbf{c}_1 \\ \vdots \\ V_m^T \mathbf{c}_m \end{bmatrix} \in \mathbf{R}^N.$$

Given a basis for LP (12), the verifying dual variables $\mathbf{y} \in \mathbf{R}^{m_0}$ and $\mathbf{z} \in \mathbf{R}^m$ can be computed. For each $(i, j) \in \{1, \dots, m\} \times \{1, \dots, N_i\}$, the reduced cost with respect to $\mathbf{y}$ and $\mathbf{z}$ is $\bar{c}_{ij} = \tilde{\mathbf{c}}_i^T \mathbf{v}_{ij} - z_i$, where $\tilde{\mathbf{c}}_i = \mathbf{c}_i - A_{0i}^T \mathbf{y}$. Thus, the reduced cost $\bar{\mathbf{c}}_i$ of the i^{th} subproblem is nonnegative if and only if $\tilde{\mathbf{c}}_i^T \mathbf{v}_{ij} \geq z_i$ for all $j \in \{1, \dots, N_i\}$.

Since $P_i = \{\mathbf{x}_i \in \mathbf{R}^{n_i} \mid A_{ii}\mathbf{x}_i = \mathbf{b}_i, \mathbf{x}_i \geq \mathbf{0}\}$ is a polytope with vertices $\mathbf{v}_{i1}, \dots, \mathbf{v}_{iN_i}$, by Lemma 5 at least one of the $\mathbf{v}_{ij}$ will satisfy $\tilde{\mathbf{c}}_i^T \mathbf{v}_{ij} \leq \tilde{\mathbf{c}}_i^T \mathbf{x}_i$ for all $\mathbf{x}_i \in P_i$. Thus, the i^{th} optimality condition $\bar{\mathbf{c}}_i \geq \mathbf{0}$ can be checked by solving subproblem (11).

Let $\mathbf{x}_i^*$ be the optimal solution to LP (11). If $\tilde{\mathbf{c}}_i^T \mathbf{x}_i^* \geq z_i$, then $\bar{\mathbf{c}}_i \geq \mathbf{0}$. This condition can be checked for each i. If $\bar{\mathbf{c}}_i \geq \mathbf{0}$ for all $i \in \{1, \dots, m\}$, then $\bar{\mathbf{c}} \geq \mathbf{0}$, so $\boldsymbol{\lambda}$ is optimal for the original LP (12). Otherwise, there exist an $i \in \{1, \dots, m\}$ and $k \in \{1, \dots, N_i\}$ such that $\tilde{\mathbf{c}}_i^T \mathbf{v}_{ik} > z_i$, so we add λ_{ik} to the basis, perform the ratio test to find a λ_{rs} that leaves the basis, and continue.

6 Good algorithms

In a 1965 paper, Jack Edmonds first discussed the difference between *finite* and *good* algorithms, *i.e.*, between algorithms that converge eventually and ones that converge "quickly." This distinction spurred the development of a formal theory of complexity. This section is gives a very brief introduction to that theory.

Definition 18 (Size) *The size of $\alpha \in \mathbf{R}$ is the number of bits needed to encode α in binary.*

- An upper bound on the size of $\alpha \in \mathbf{Z}$ is $\text{size}(\alpha) \leq \lceil \log_2(\alpha+1) \rceil + 1$.
- The size of a vector or matrix is the sum of the sizes of its components.

Definition 19 (Good) *An algorithm is good if its runtime can be bounded by a polynomial in the size of its inputs.*

By this definition, the simplex method is not a good algorithm: no known pivot rule guarantees fewer than $2^n - 1$ pivots in the worst case (see §5.8). Good algorithms for linear programming do exist, however. One is the ellipsoid method discussed in §7. Another is the family of interior-point methods discussed in §8.

Note that the size of the inputs to the LP $\min\{\mathbf{c}^T\mathbf{x} \mid A\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ with $A \in \mathbf{R}^{m \times n}$ is *not* directly a function of m and n. Rather, it's $\text{size}(A) + \text{size}(\mathbf{b}) + \text{size}(\mathbf{c})$, which depends on the sizes of the entries in A, $\mathbf{b}$ and $\mathbf{c}$, as well as on m and n. A LP algorithm with runtime bounded by a polynomial in m and n would be called **strongly polynomial**. No one knows whether a strongly polynomial LP algorithm exists.

The next two claims bound the input and output size of the LP

$$\begin{array}{ll} \text{minimize} & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & A\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{array} \tag{13}$$

where A, $\mathbf{b}$ and $\mathbf{c}$ are assumed to have integer entries. Both claims use the definition $U = \max(\max_{i,j} \text{size}(a_{ij}), \max_i \text{size}(b_i), \max_j \text{size}(c_j))$.

Claim 23 *The size of the inputs to LP (13) is $\mathcal{O}(nmU)$.*

Claim 24 *If $\mathbf{x}$ solves LP (13), then for any $j \in \{1, \ldots, n\}$, $\text{size}(x_j) = \mathcal{O}(n \log_2 n + nU)$.*

An optimal $\mathbf{x}$ has at most m nonzero entries, so $\text{size}(\mathbf{x}) = \mathcal{O}(m(n \log_2 n + nU))$. Thus, the number of bits required to encode the output of an LP algorithm is bounded by a polynomial in the size of the inputs to the LP. (If this weren't true, then there'd be no hope for finding a good LP algorithm.)

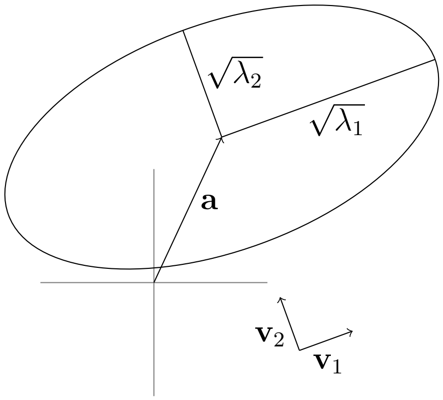

Figure 2: the ellipsoid $\mathcal{E}(\mathbf{a}, A)$. The matrix A has eigenvalues λ_1 and λ_2 and eigenvectors $\mathbf{v}_1$ and $\mathbf{v}_2$.

7 Ellipsoid methods

In 1979, the Russian mathematician Leonid Khachiyan used an ellipsoid method to give the first proof that linear programs can be solved in polynomial time. Ellipsoid methods are not generally regarded as practical algorithms, but they are good algorithms according to Definition 19. Ellipsoid methods can also be extended to other classes of optimization problems.

The basic idea of ellipsoid methods is to start with an ellipsoid large enough to contain the entire feasible region, cut the ellipsoid in half with a "well-chosen" hyperplane through its center, draw a new ellipsoid around the "good" half, and repeat. If "well-chosen" and "good" are defined appropriately, then the algorithm will converge to the desired result (typically, either a feasible point or an optimum) in polynomial time.

We begin with some geometry, then discuss two ellipsoid methods for linear programming.

7.1 Ellipsoid geometry

Definition 20 (Ellipsoid) *The set* $\mathcal{E}(\mathbf{a}, A) = \left\{\mathbf{x} \in \mathbf{R}^n \mid (\mathbf{x}-\mathbf{a})^T A^{-1}(\mathbf{x}-\mathbf{a}) \leq 1\right\}$*, where* $A \succ 0$*, is an ellipsoid.*

The ellipsoid $\mathcal{E}(\mathbf{a}, A)$ is centered at $\mathbf{a}$, and its size and shape are determined by A. The eigenvectors of A determine the ellipsoid's orientation. The square roots of the eigenvalues of A are the lengths of the ellipsoid's semi-axes (see Figure 2). The unit sphere is $\mathcal{E}(\mathbf{0}, I)$.

Claim 25 *The volume of the ellipsoid* $\mathcal{E}(\mathbf{a}, A) \subset \mathbf{R}^n$ *is* $\beta_n \sqrt{\det A}$*, where* $\beta_n = \frac{\pi^{n/2}}{\Gamma(1+n/2)}$ *is the volume of the unit hypersphere in* $\mathbf{R}^n$*.*

The next claim involves cutting an ellipsoid with a hyperplane through its center, as shown in Figure 3.

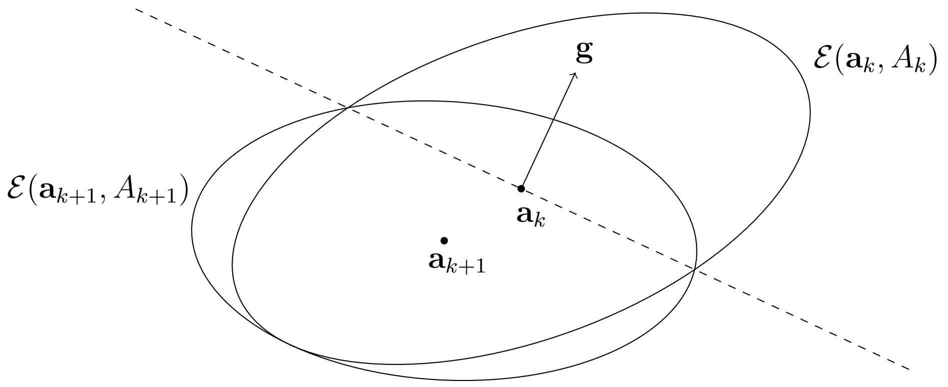
$\mathbf{g}$
$\mathcal{E}(\mathbf{a}_k, A_k)$
$\mathcal{E}(\mathbf{a}_{k+1}, A_{k+1})$
$\mathbf{a}_k$
$\mathbf{a}_{k+1}$

Figure 3: $\mathcal{E}(\mathbf{a}_{k+1}, A_{k+1})$ is the minimum volume ellipsoid containing the half-ellipsoid $\mathcal{E}(\mathbf{a}_k, A_k) \cap \{\mathbf{x} \mid \mathbf{g}^T(\mathbf{x} - \mathbf{a}_k) \leq 0\}$.

Claim 26 (Minimum volume ellipsoid) *Let $\mathcal{E}(\mathbf{a}_k, A_k) \subset \mathbf{R}^n$. For any $\mathbf{g} \in \mathbf{R}^n$, the minimum volume ellipsoid containing the half-ellipsoid $\mathcal{E}(\mathbf{a}_k, A_k) \cap \{\mathbf{x} \in \mathbf{R}^n \mid \mathbf{g}^T(\mathbf{x} - \mathbf{a}_k) \leq 0\}$ is $\mathcal{E}(\mathbf{a}_{k+1}, A_{k+1})$, where*

$$\begin{aligned} \mathbf{a}_{k+1} &= \mathbf{a}_k - \frac{1}{n+1} A_k \tilde{\mathbf{g}} \\ A_{k+1} &= \frac{n^2}{n^2-1}\left(A_k - \frac{2}{n+1} A_k \tilde{\mathbf{g}} \tilde{\mathbf{g}}^T A_k\right) \\ \tilde{\mathbf{g}} &= \frac{1}{\sqrt{\mathbf{g}^T A_k \mathbf{g}}}\mathbf{g}. \end{aligned} \tag{14}$$

Claim 27 *Let $\mathcal{E}(\mathbf{a}_k, A_k)$ and $\mathcal{E}(\mathbf{a}_{k+1}, A_{k+1})$ be defined as in Claim 26. Then*

$$\frac{\operatorname{vol}(\mathcal{E}(\mathbf{a}_{k+1}, A_{k+1}))}{\operatorname{vol}(\mathcal{E}(\mathbf{a}_k, A_k))} = \frac{\sqrt{\det(A_{k+1})}}{\sqrt{\det(A_k)}} \leq e^{-\frac{1}{2(n+1)}}.$$

7.2 The basic ellipsoid method

Suppose we had an oracle that took a polytope Q and either returned an $\mathbf{x} \in Q$ or certified that $Q = \emptyset$.

How could we use such an oracle to solve an LP?

Consider the primal-dual pair

$$\begin{array}{llcll} \text{minimize} & \mathbf{c}^T\mathbf{x} & \qquad & \text{maximize} & \mathbf{b}^T\mathbf{y} \\ \text{subject to} & A\mathbf{x} \leq \mathbf{b} & & \text{subject to} & A^T\mathbf{y} \leq \mathbf{c} \\ & \mathbf{x} \geq \mathbf{0} & & & \mathbf{y} \geq \mathbf{0} \end{array} \qquad (15, 16)$$

with feasible regions $\mathcal{F}(P) = \{\mathbf{x} \in \mathbf{R}^n \mid A\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ and $\mathcal{F}(D) = \{\mathbf{y} \in \mathbf{R}^m \mid A^T\mathbf{y} \leq \mathbf{c}, \mathbf{y} \geq \mathbf{0}\}$ both bounded.

We can solve the LPs (15) and (16) with three calls to our oracle:

1. Check $\mathcal{F}(P)$. If $\mathcal{F}(P) = \emptyset$, the primal is infeasible.
2. If $\mathcal{F}(P) \neq \emptyset$, check $\mathcal{F}(D)$. If $\mathcal{F}(D) = \emptyset$, the primal is unbounded.
3. If $\mathcal{F}(P) \neq \emptyset$ and $\mathcal{F}(D) \neq \emptyset$, check $\left\{(\mathbf{x}, \mathbf{y}) \in \mathcal{F}(P) \times \mathcal{F}(D) \mid \mathbf{c}^T\mathbf{x} \leq \mathbf{b}^T\mathbf{y}\right\}$. By weak duality, $\mathbf{c}^T\mathbf{x} \geq \mathbf{b}^T\mathbf{y}$ for all $\mathbf{x} \in \mathcal{F}(P)$ and $\mathbf{y} \in \mathcal{F}(D)$, so the $(\mathbf{x}, \mathbf{y})$ returned by the method satisfy $\mathbf{c}^T\mathbf{x} = \mathbf{b}^T\mathbf{y}$. By strong duality, therefore, $\mathbf{x}$ and $\mathbf{y}$ are optimal.

If the oracle runs in polynomial time in the sizes of its inputs, then so does this algorithm.

7.2.1 Explaining the oracle

Given a polytope Q, how can we use ellipsoids to find an $\mathbf{x} \in Q$ or certify that $Q = \emptyset$?

Consider a general polytope

$$Q = \{\mathbf{x} \in \mathbf{R}^n \mid B\mathbf{x} \leq \mathbf{d}\}$$

where

$$B = \begin{bmatrix} \boldsymbol{\beta}_1^T \\ \vdots \\ \boldsymbol{\beta}_m^T \end{bmatrix} \in \mathbf{R}^{m \times n}.$$

Assume for now that we have a subroutine that checks whether a polytope is empty. Given such a subroutine and a $Q \neq \emptyset$, the following algorithm produces an $\mathbf{x} \in Q$ ($\mathbf{x}$ happens to be a vertex).

1. Set $S = \emptyset$.
2. For $i \in \{1, \ldots, m\}$,
   if $\left\{\mathbf{x} \in \mathbf{R}^n \mid \boldsymbol{\beta}_j^T\mathbf{x} \leq d_j \ \forall\ j \notin S,\ \boldsymbol{\beta}_j^T\mathbf{x} = d_j \ \forall\ j \in S \cup \{i\}\right\} \neq \emptyset$, set $S = S \cup \{i\}$.
3. Solve the system $\left\{\boldsymbol{\beta}_j^T\mathbf{x} = d_j \ \forall\ j \in S\right\}$ for $\mathbf{x}$.

This algorithm's runtime is polynomial in the sizes of B and $\mathbf{d}$, provided that the subroutine runs in polynomial time.

7.2.2 Explaining the subroutine

Given a polytope Q, how can we use ellipsoids to check whether $Q = \emptyset$?

Consider the polytope

$$\tilde{Q} = \left\{ \mathbf{x} \in \mathbf{R}^n \;\middle|\; B\mathbf{x} \leq \mathbf{d} + \frac{2^{-L}}{n+2}\mathbf{1} \right\}.$$

Let L denote the number of bits required to encode B and $\mathbf{d}$, *i.e.*, $L = \operatorname{size}(B) + \operatorname{size}(\mathbf{d})$. Let U denote maximum number of bits required to encode any entry in B or $\mathbf{d}$, *i.e.*, $U = \max\{\max_{i,j} \operatorname{size}(b_{ij}), \max_i \operatorname{size}(d_i)\}$.

Claim 28 *$Q \subseteq \mathcal{E}(\mathbf{0}, 2^L I)$.*

Proof: By Claim 24, any element x_j of a vertex $\mathbf{x}$ of Q requires at most $\mathcal{O}(nU + n\log_2 n)$ bits to encode, so $|x_j| \leq 2^{nU + n\log_2 n}$. By Claim 23, $L = \mathcal{O}(nmU)$, so $|x_j| \leq 2^L$ for all $j \in \{1, \ldots, n\}$. □

In other words: a sphere of radius 2^L centered at the origin is big enough to contain Q. This gives a starting point for the ellipsoid method.

Lemma 29 *If $Q = \emptyset$, then $\tilde{Q} = \emptyset$.*

By contraposition, this lemma implies that if $\tilde{Q} \neq \emptyset$, then $Q \neq \emptyset$.

Lemma 30 *If $Q \neq \emptyset$, then there exists an $\hat{\mathbf{x}} \in \tilde{Q}$ such that*

$$\mathcal{E}\left(\hat{\mathbf{x}}, \frac{2^{-2L}}{n+2} I\right) \subseteq \tilde{Q}.$$

By contraposition, this lemma implies that if $\operatorname{vol}(\tilde{Q}) < \operatorname{vol}\left(\mathcal{E}\left(\hat{\mathbf{x}}, \frac{2^{-2L}}{n+2} I\right)\right)$, then $Q = \emptyset$.

These lemmas allow us to run the following algorithm on $\tilde{Q}$ in order to check whether Q is empty.

1. Set $k = 0$, $\mathbf{a}_k = \mathbf{0}$, $A_k = 2^L I$.
2. If $\mathbf{a}_k \in \tilde{Q}$, return $Q \neq \emptyset$.
3. If $\operatorname{vol}(\mathcal{E}(\mathbf{a}_k, A_k)) < \operatorname{vol}\left(\mathcal{E}\left(\mathbf{0}, \frac{2^{-2L}}{n+2} I\right)\right)$, return $Q = \emptyset$.
4. Pick an i such that $\boldsymbol{\beta}_i^T \mathbf{a}_k > d_i$. Set $\mathbf{g} = \boldsymbol{\beta}_i$ and update $\mathbf{a}_k$ and A_k according to the minimum volume ellipsoid equations (14).
5. Set $k = k + 1$ and go to Step 2.

Claim 31 *After $\mathcal{O}(n)$ iterations, the ellipsoid volume shrinks by at least a factor of two.*

In symbols, this claim says that

$$\operatorname{vol}(\mathcal{E}(\mathbf{a}_{k+\mathcal{O}(n)}, A_{k+\mathcal{O}(n)})) = \frac{1}{2}\operatorname{vol}(\mathcal{E}(\mathbf{a}_k, A_k)).$$

The proof uses Claim 27.

Claim 32 *The algorithm terminates in $\mathcal{O}(n^2 L)$ iterations.*

Proof: By Claim 25, the volume of $\mathcal{E}(\mathbf{a}_0, A_0) = \mathcal{E}(\mathbf{0}, 2^L I)$ is $2^{\mathcal{O}(nL)}$. By Claim 30, if $Q \neq \emptyset$, then $\operatorname{vol}(\tilde{Q}) \geq \operatorname{vol}\left(\mathcal{E}\left(\hat{\mathbf{x}}, \frac{2^{-2L}}{n+2} I\right)\right) = 2^{\Omega(nL)}$. So after $\mathcal{O}(n^2 L)$ iterations ($\mathcal{O}(n)$ iterations per $\mathcal{O}(nL)$ volume reduction), the algorithm will either have found an $\mathbf{a}_k \in \tilde{Q}$ or certified that $Q = \emptyset$. □

Since L is polynomial in the inputs to the LP (see Claim 23), so is the algorithm's runtime.

7.3 The ellipsoid method with objective function cuts

The basic ellipsoid method relies primarily on information about the feasible region, making little reference to the objective function. We now describe an ellipsoid method that directly uses information about the objective function.

Consider again the LP

$$\begin{array}{ll} \text{minimize} & \mathbf{c}^T \mathbf{x} \\ \text{subject to} & A\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{array}$$

where

$$A = \begin{bmatrix} \boldsymbol{\alpha}_1^T \\ \vdots \\ \boldsymbol{\alpha}_m^T \end{bmatrix} \in \mathbf{R}^{m \times n}$$

and $\mathcal{F}(P) = \{\mathbf{x} \in \mathbf{R}^n \mid A\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ is bounded. Let $L = \operatorname{size}(A) + \operatorname{size}(\mathbf{b})$.

The method uses a mix of feasibility cuts, where the cutting hyperplane is normal to the gradient of a violated constraint, and objective function cuts, where the cutting hyperplane is normal to the gradient of the objective function.

1. Set $k = 0$, $\mathbf{a}_k = \mathbf{0}$, $A_k = 2^L I$.

2. If $\operatorname{vol}(\mathcal{E}(\mathbf{a}_k, A_k))$ is sufficiently small that $\mathcal{E}(\mathbf{a}_k, A_k)$ contains exactly one vertex, enter a rounding routine to find and return the optimal vertex $\mathbf{x}^*$.

3. If $\operatorname{vol}(\mathcal{E}(\mathbf{a}_k, A_k)) < \operatorname{vol}\left(\mathcal{E}\left(\mathbf{0}, \frac{2^{-2L}}{n+2} I\right)\right)$, return infeasible.

4. If $\mathbf{a}_k \notin \mathcal{F}(P)$, pick an i such that $\boldsymbol{\alpha}_i^T \mathbf{a}_k > d_i$ and set $\mathbf{g} = \boldsymbol{\alpha}_i$.

5. If $\mathbf{a}_k \in \mathcal{F}(P)$, set $\mathbf{g} = \mathbf{c}$.

6. Update $\mathbf{a}_k$ and A_k according to the minimum volume ellipsoid equations (14).

7. Set $k = k + 1$ and go to step 2.

7.4 Separation oracles

Definition 21 (Separation oracle) *Given a polytope Q and a vector $\mathbf{x}$, a separation oracle checks whether $\mathbf{x} \in Q$ or returns a violated constraint.*

Given a polynomial-time separation oracle for the feasible region, any LP can be solved in polynomial time. The ellipsoid method is an important example of a separation oracle.

8 Interior-point methods

While the ellipsoid method is much slower than the simplex method in practice, it proved the existence of a polynomial-time algorithm for linear programming. Interior-point methods (IPMs) for linear programming, first developed by Narendra Karmakar in 1984, are provably polynomial-time and compete with the simplex method in practice. In fact, some commercial LP solvers run the simplex method and an IPM in parallel on separate cores and return the solution of whichever algorithm finishes first. IPMs are also interesting because they readily extend to more general optimization problems, such as conic programming (see §9).

There are many varieties of IPMs. One variety, called **path-following**, has an upper bound of $\mathcal{O}(n \ln(1/\epsilon))$ iterations to come within a factor of $(1+\epsilon)$ of the optimal value. Another variety, called **potential-reduction**, has a tighter theoretical bound of $\mathcal{O}(\sqrt{n} \ln(1/\epsilon))$. Path-following methods tend to be faster in practice, however. On a typical LP, a good IPM gets within $1 + 10^{-8}$ of the optimal value in 10-50 (costly) iterations.

We develop IPMs in the context of the primal-dual pair

$$\begin{array}{ll} \text{minimize} & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{array} \qquad \begin{array}{ll} \text{maximize} & \mathbf{b}^T\mathbf{y} \\ \text{subject to} & A^T\mathbf{y} + \mathbf{s} = \mathbf{c} \\ & \mathbf{s} \geq \mathbf{0} \end{array} \tag{17, 18}$$

where $A \in \mathbf{R}^{m \times n}$ and $\text{rank}(A) = m$. Denote the feasible regions and their interiors by

$$\begin{aligned} \mathcal{F}(P) &= \{\mathbf{x} \in \mathbf{R}^n \mid A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\} \\ \mathcal{F}(D) &= \left\{(\mathbf{y}, \mathbf{s}) \in \mathbf{R}^m \times \mathbf{R}^n \mid A^T\mathbf{y} + \mathbf{s} = \mathbf{c}, \mathbf{s} \geq \mathbf{0}\right\} \\ \mathcal{F}^\circ(P) &= \{\mathbf{x} \in \mathbf{R}^n \mid A\mathbf{x} = \mathbf{b}, \mathbf{x} > \mathbf{0}\} \\ \mathcal{F}^\circ(D) &= \left\{(\mathbf{y}, \mathbf{s}) \in \mathbf{R}^m \times \mathbf{R}^n \mid A^T\mathbf{y} + \mathbf{s} = \mathbf{c}, \mathbf{s} > \mathbf{0}\right\}. \end{aligned}$$

At a high level, any IPM can be viewed as a way to generate a sequence of points in either $\mathcal{F}^\circ(P)$ or $\mathcal{F}^\circ(P) \times \mathcal{F}^\circ(D)$ that converges to an optimum.

8.1 Finding a descent direction that preserves feasibility

Given an initial feasible point $\bar{\mathbf{x}} \in \mathcal{F}^\circ(P)$, we can begin generating this sequence by finding a search direction $\bar{\mathbf{d}}$ that improves the objective and preserves feasibility. Without loss of generality, we require $\left\|\bar{\mathbf{d}}\right\|_2 \leq 1$ and define $\mathbf{x} = \bar{\mathbf{x}} + \alpha\bar{\mathbf{d}}$, where $\alpha > 0$ is the step size. Since $\bar{\mathbf{x}} > \mathbf{0}$, α can always be chosen small enough that $\mathbf{x} > \mathbf{0}$. For $\mathbf{x}$ to remain feasible, then, the only requirement is that $A\mathbf{x} = \mathbf{b}$. This implies that that $\bar{\mathbf{d}} \in \text{null}(A)$.

The problem of finding the feasibility-preserving direction that yields the best improvement in the objective function can be written as

$$\begin{array}{ll} \text{minimize} & \mathbf{c}^T\mathbf{d} \\ \text{subject to} & A\mathbf{d} = \mathbf{0} \\ & \|\mathbf{d}\|_2 \leq 1. \end{array} \tag{19}$$

Lemma 33 *If no* $\mathbf{y} \in \mathbf{R}^m$ *satisfies* $A^T\mathbf{y} = \mathbf{c}$*, then the solution to* (19) *is*

$$\bar{\mathbf{d}} = -\frac{P_A\mathbf{c}}{\|P_A\mathbf{c}\|_2}$$

where $P_A = I - A^T(AA^T)^{-1}A$*.*

Since P_A is symmetric and idempotent, it's a projection matrix. Notice that $AP_A = 0$, so for any $\mathbf{z} \in \mathbf{R}^n$, $P_A\mathbf{z} \in \text{null}(A)$. This gives the geometric interpretation of $\bar{\mathbf{d}}$ as the (negated, normalized) projection of $\mathbf{c}$ onto the null space of A.

If no $\mathbf{y} \in \mathbf{R}^m$ satisfies $A^T\mathbf{y} = \mathbf{c}$, then $\bar{\mathbf{d}}$ strictly improves the objective:

$$\begin{aligned}
\mathbf{c}^T(\bar{\mathbf{x}} + \alpha\bar{\mathbf{d}}) &= \mathbf{c}^T\bar{\mathbf{x}} - \frac{\alpha}{\|P_A\mathbf{c}\|_2}\mathbf{c}^TP_A\mathbf{c} \\
&= \mathbf{c}^T\bar{\mathbf{x}} - \frac{\alpha}{\|P_A\mathbf{c}\|_2}\mathbf{c}^TP_A^TP_A\mathbf{c} \\
&= \mathbf{c}^T\bar{\mathbf{x}} - \alpha\,\|P_A\mathbf{c}\|_2 \\
&< \mathbf{c}^T\bar{\mathbf{x}}
\end{aligned}$$

where the last line follows from the facts that $\alpha > 0$ and $\|P_A\mathbf{c}\|_2 > 0$. To see that the latter is true, observe that $P_A\mathbf{c} = \mathbf{0}$ if and only if $A^T(AA^T)^{-1}A\mathbf{c} = \mathbf{c}$. But this is equivalent to $A^T\mathbf{y} = \mathbf{c}$, where $\mathbf{y} = (AA^T)^{-1}A\mathbf{c}$. This is prohibited by assumption, so $P_A\mathbf{c} \neq \mathbf{0}$ and $\|P_A\mathbf{c}\|_2 > 0$.

8.2 The affine-scaling direction

In §8.1, we showed that if no $\mathbf{y} \in \mathbf{R}^m$ satisfies $A^T\mathbf{y} = \mathbf{c}$, then $\bar{\mathbf{d}} = -(1/\|P_A\mathbf{c}\|_2)P_A\mathbf{c}$ solves (19). This value of $\bar{\mathbf{d}}$ is the steepest feasibility-preserving descent direction for *any* $\mathbf{x} \in \mathcal{F}^\circ(P)$, whether $\mathbf{x}$ is near or far from the boundary. This poses an implementation problem: if $\mathbf{x}$ is near the boundary of the feasible region, then $\mathbf{x}+\alpha\bar{\mathbf{d}}$ may remain an interior point only for very small values of α. Finding a "small enough" step size is nontrivial.

A common way to address this difficulty is to transform the problem such that $\mathbf{x}$ is some standardized distance from the boundary of the feasible region, find the optimal search direction in the transformed space, and then invert the transformation. The simplest such transformation is $\hat{\mathbf{x}} = X^{-1}\mathbf{x}$, where

$$X = \text{diag}(\mathbf{x}) = \begin{bmatrix} x_1 & & \\ & \ddots & \\ & & x_n \end{bmatrix} \quad\Longrightarrow\quad X^{-1} = \begin{bmatrix} 1/x_1 & & \\ & \ddots & \\ & & 1/x_n \end{bmatrix}.$$

This transformation maps $\mathbf{x}$ into $\hat{\mathbf{x}} = \mathbf{1}$. Under this transformation, the primal (17) becomes

$$\begin{aligned}
\text{minimize}\quad & \hat{\mathbf{c}}^T\hat{\mathbf{x}} \\
\text{subject to}\quad & \hat{A}\hat{\mathbf{x}} = \mathbf{b} \\
& \hat{\mathbf{x}} \geq \mathbf{0}
\end{aligned} \tag{20}$$

where $\hat{\mathbf{c}} = X\mathbf{c}$ and $\hat{A} = AX$.

In the transformed space, we once again seek a descent direction $\mathbf{d}$ that preserves feasibility. By the same arguments used in §8.1, the best such $\mathbf{d}$ solves

$$\begin{array}{ll} \text{minimize} & \hat{\mathbf{c}}^T\mathbf{d} \\ \text{subject to} & \hat{A}\mathbf{d} = \mathbf{0} \\ & \|\mathbf{d}\|_2 \leq 1. \end{array} \tag{21}$$

By Lemma 33, if no $\mathbf{y} \in \mathbf{R}^m$ satisfies $\hat{A}^T\mathbf{y} = \hat{\mathbf{c}}$, then the solution to (21) is

$$\hat{\mathbf{d}} = -\frac{P_{\hat{A}}\hat{\mathbf{c}}}{\|P_{\hat{A}}\hat{\mathbf{c}}\|_2}$$

where $P_{\hat{A}} = I - \hat{A}^T(\hat{A}\hat{A}^T)^{-1}\hat{A}$. Inverting the transformation gives the **affine-scaling direction**,

$$\tilde{\mathbf{d}} = X\hat{\mathbf{d}} = -\frac{XP_{\hat{A}}\hat{\mathbf{c}}}{\|P_{\hat{A}}\hat{\mathbf{c}}\|_2}.$$

Note that the affine-scaling direction $\tilde{\mathbf{d}}$ solves

$$\begin{array}{ll} \text{minimize} & \mathbf{c}^T\mathbf{d} \\ \text{subject to} & A\mathbf{d} = \mathbf{0} \\ & \|X^{-1}\mathbf{d}\|_2 \leq 1. \end{array}$$

The norm constraint in the above provides a natural interpretation of the affine-scaling direction. If $\mathbf{x}$ is very near the boundary of the feasible region, then some x_i is nearly zero. In order for $\|X^{-1}\mathbf{d}\|_2 = \sqrt{(d_1/x_1)^2 + \cdots + (d_n/x_n)^2}$ to be less than one, therefore, the corresponding d_i must be even closer to zero. Thus, the affine-scaling direction can be thought of as a force that repels $\mathbf{x}$ from the boundary; the closer $\mathbf{x}$ gets, the harder it pushes.

Although the affine-scaling direction is used in some practical IPM implementations, it hasn't been proven to yield a polynomial-time LP algorithm. To generate a provably polynomial-time IPM, we formalize the notion of "repelling $\mathbf{x}$ from the boundary" by introducing the logarithmic barrier function.

8.3 The logarithmic barrier function

It can be shown (*e.g.*, by Lagrange multipliers) that $-XP_{\hat{A}}\hat{\mathbf{c}}$, the direction of $\tilde{\mathbf{d}}$, solves

$$\begin{array}{ll} \text{minimize} & \mathbf{c}^T\mathbf{d} + \frac{1}{2}\|X^{-1}\mathbf{d}\|_2^2 \\ \text{subject to} & A\mathbf{d} = \mathbf{0}. \end{array}$$

This problem can be viewed as a trade-off between making progress toward the objective (by making $\mathbf{c}^T\mathbf{d}$ small) and keeping $\mathbf{x}$ away from the boundary of the feasible region (by keeping $\|X^{-1}\mathbf{d}\|_2^2$ small).

The weighted, squared 2-norm $\|X^{-1}\mathbf{d}\|_2^2$ is a fairly arbitrary measure of proximity to the boundary of the feasible region. It turns out that a better measure is the **logarithmic barrier function**,

$$F(\mathbf{x}) = -\sum_{j=1}^{n} \ln(x_j).$$

The minimizer of $F(\mathbf{x})$ over $\mathcal{F}^\circ(P)$, which is also the maximizer of $\prod_{j=1}^n x_j$ over $\mathcal{F}^\circ(P)$, is called the **analytic center** of $\mathcal{F}^\circ(P)$.

To negotiate the trade-off between progress and centrality, we minimize over $\mathcal{F}^\circ(P)$ the function

$$B_\gamma(\mathbf{x}) = \mathbf{c}^T\mathbf{x} + \gamma F(\mathbf{x})$$

parameterized by γ. For large values of γ, the minimizer of $B_\gamma(\mathbf{x})$ is nearly the analytic center of $\mathcal{F}^\circ(P)$. As $\gamma \to 0$, the minimizer of $B_\gamma(\mathbf{x})$ tends toward an optimum of the original primal (17).

Theorem 34 (Existence of a minimizer) *There exists a minimizer of $B_\gamma(\mathbf{x})$ on $\mathcal{F}^\circ(P)$ if and only if $\mathcal{F}^\circ(P)$ and $\mathcal{F}^\circ(D)$ are nonempty.*

Theorem 35 (Uniqueness of the minimizer) *Let $\mathcal{F}^\circ(P)$ and $\mathcal{F}^\circ(D)$ be nonempty. A vector $\mathbf{x}$ is the unique minimizer of $B_\gamma(\mathbf{x})$ on $\mathcal{F}^\circ(P)$ if and only if there exists a $(\mathbf{y}, \mathbf{s}) \in \mathcal{F}^\circ(D)$ such that*

$$\begin{aligned} A^T\mathbf{y} + \mathbf{s} &= \mathbf{c} \\ A\mathbf{x} &= \mathbf{b} \\ XS\mathbf{1} &= \gamma\mathbf{1} \end{aligned} \tag{22}$$

where $X = \operatorname{diag}(\mathbf{x})$ and $S = \operatorname{diag}(\mathbf{s})$.

8.4 A primal-dual path-following method

Note that the last equation in system (22) reduces to complementary slackness for $\gamma = 0$. In this limit, the system enforces primal feasibility, dual feasibility and complementary slackness. As $\gamma \to 0$, therefore, the solutions to system (22) tend toward optimality. Path-following methods use this idea to generate a sequence of points in either $\mathcal{F}^\circ(P)$ or $\mathcal{F}^\circ(P) \times \mathcal{F}^\circ(D)$ that follow a **central path** toward optimality. There are many varieties of path-following methods. This section describes a primal-dual variety.

Definition 22 (Central path) *Let $\mathbf{x}(\gamma)$, $\mathbf{y}(\gamma)$ and $\mathbf{s}(\gamma)$ solve system (22) for a fixed value of γ. We call $\{\mathbf{x}(\gamma) \in \mathcal{F}^\circ(P) \mid \gamma > 0\}$ the primal central path and $\{(\mathbf{x}(\gamma), \mathbf{y}(\gamma), \mathbf{s}(\gamma)) \in \mathcal{F}^\circ(P) \times \mathcal{F}^\circ(D) \mid \gamma > 0\}$ the primal-dual central path.*

System (22) involves $n + m$ linear equations coupled through n nonlinear equations. Solving it means finding a zero of the function

$$\mathbf{f}_\gamma(\mathbf{x}, \mathbf{y}, \mathbf{s}) = \begin{bmatrix} A^T\mathbf{y} + \mathbf{s} - \mathbf{c} \\ A\mathbf{x} - \mathbf{b} \\ XS\mathbf{1} - \gamma\mathbf{1} \end{bmatrix}$$

that satisfies $\mathbf{x}, \mathbf{s} \geq \mathbf{0}$. The classical algorithm for such nonlinear root-finding problems is Newton's method, which we will use with minor variations.

8.4.1 Choosing γ

Recall that γ governs the trade-off between making progress toward optimality and staying away from the boundaries of the feasible regions, as discussed in §8.3. There are several common choices of γ, each of which defines a different primal-dual path-following IPM. These methods all set $\gamma = \sigma\mu$ for some $\sigma \in [0, 1]$, where $\mu = \mathbf{x}^T\mathbf{s}/n$ is a scaled measure of the optimality gap.

The methods differ in their choice of σ. If $\sigma = 0$, the step will be toward optimality (an **affine-scaling step**). If $\sigma = 1$, the step will be toward the analytic centers of $\mathcal{F}^\circ(P)$ and $\mathcal{F}^\circ(D)$ (a **centering step**). A value of $\sigma \in (0, 1)$ trades off between the two.

Long-step methods use values of σ nearly zero, **short-step** use values of σ nearly one, and **predictor-corrector** methods alternate between centering and affine-scaling steps. Short-step and predictor-corrector methods have the tightest theoretical runtime bounds, both guaranteeing $\mu < \epsilon$ within $\mathcal{O}(\sqrt{n} \ln(1/\epsilon))$ iterations. Long-step methods have a looser $\mathcal{O}(n \ln(1/\epsilon))$ bound, but tend to out-perform short-step methods in practice. Many commercial solvers implement the predictor-corrector method.

8.4.2 Newton direction

Suppose that we're given an $\mathbf{x} \in \mathcal{F}^\circ(P)$ and $(\mathbf{y}, \mathbf{s}) \in \mathcal{F}^\circ(D)$ and that we've chosen γ somehow. We can follow Newton's method by seeking a direction $(\Delta\mathbf{x}, \Delta\mathbf{y}, \Delta\mathbf{s})$ that satisfies

$$D_\mathbf{f}(\mathbf{x}, \mathbf{y}, \mathbf{s}) \begin{bmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{y} \\ \Delta\mathbf{s} \end{bmatrix} + \mathbf{f}_\gamma(\mathbf{x}, \mathbf{y}, \mathbf{s}) = \mathbf{0} \tag{23}$$

where $D_\mathbf{f}(\mathbf{x}, \mathbf{y}, \mathbf{s})$ is the Jacobian of $\mathbf{f}_\gamma(\mathbf{x}, \mathbf{y}, \mathbf{s})$,

$$D_\mathbf{f}(\mathbf{x}, \mathbf{y}, \mathbf{s}) = \begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix}.$$

When solving (23) given a feasible $\mathbf{x}$ and $(\mathbf{y}, \mathbf{s})$, we have $A\mathbf{x} = \mathbf{b}$ and $A^T\mathbf{y} + \mathbf{s} = \mathbf{c}$, so the system reduces to

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{y} \\ \Delta\mathbf{s} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ -XS\mathbf{1} + \gamma\mathbf{1} \end{bmatrix}.$$

A direction $(\Delta\mathbf{x}, \Delta\mathbf{y}, \Delta\mathbf{s})$ satisfying (23) is called a **Newton direction.**

Once we've found a Newton direction, we update the iterates according to

$$\begin{bmatrix} \mathbf{x}^+ \\ \mathbf{y}^+ \\ \mathbf{s}^+ \end{bmatrix} = \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{s} \end{bmatrix} + \alpha \begin{bmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{y} \\ \Delta\mathbf{s} \end{bmatrix}$$

where the step size α is chosen such that $\mathbf{x}^+ \in \mathcal{F}^\circ(P)$ and $(\mathbf{y}^+, \mathbf{s}^+) \in \mathcal{F}^\circ(D)$. We then compute a new γ and repeat.

8.5 A potential-reduction method

In potential-reduction methods, we define a potential function that is small only if $\mathbf{x}$ and $(\mathbf{y}, \mathbf{s})$ are nearly optimal. At each iteration, we choose a step that reduces the potential function, driving the iterates toward optimality. There are many varieties of potential-reduction methods. This section describes one.

What form should we choose for the potential function? Recall that for $\mathbf{x} \in \mathcal{F}^\circ(P)$ and $(\mathbf{y}, \mathbf{s}) \in \mathcal{F}^\circ(D)$,

$$\mathbf{x}^T\mathbf{s} = \mathbf{x}^T(\mathbf{c} - A^T\mathbf{y}) = \mathbf{c}^T\mathbf{x} - (A\mathbf{x})^T\mathbf{y} = \mathbf{c}^T\mathbf{x} - \mathbf{b}^T\mathbf{y}.$$

In other words, $\mathbf{x}^T\mathbf{s}$ is the optimality gap: if $\mathbf{x}^T\mathbf{s} = 0$, then $\mathbf{c}^T\mathbf{x} = \mathbf{b}^T\mathbf{y}$. This motivates defining the potential function

$$G_q(\mathbf{x}, \mathbf{s}) = q \ln(\mathbf{x}^T\mathbf{s}) + F(\mathbf{x}) + F(\mathbf{s}).$$

Minimizing $G_q(\mathbf{x}, \mathbf{s})$ can be viewed as negotiating a trade-off between decreasing the optimality gap and keeping $\mathbf{x}$ and $(\mathbf{y}, \mathbf{s})$ near the analytic centers of $\mathcal{F}^\circ(P)$ and $\mathcal{F}^\circ(D)$, respectively. This trade-off is parameterized by q.

What's a good choice of q? It turns out that $q = n + \sqrt{n}$ is a good one.

Claim 36 *If $G_{n+\sqrt{n}}(\mathbf{x}, \mathbf{s}) \le -\sqrt{n}\ln(1/\epsilon)$, then $\mathbf{x}^T\mathbf{s} \le \epsilon$.*

Proof: By definition, for any value of q, $G_q(\mathbf{x}, \mathbf{s}) - G_n(\mathbf{x}, \mathbf{s}) = (q - n)\ln(\mathbf{x}^T\mathbf{s})$. In particular, for $q = n + \sqrt{n}$, we have $G_{n+\sqrt{n}}(\mathbf{x}, \mathbf{s}) = G_n(\mathbf{x}, \mathbf{s}) + \sqrt{n}\ln(\mathbf{x}^T\mathbf{s})$.

Note that $G_n(\mathbf{x}, \mathbf{s}) \ge n\ln(n)$:

$$\begin{aligned} G_n(\mathbf{x}, \mathbf{s}) &= n\ln(\mathbf{x}^T\mathbf{s}) + F(\mathbf{x}) + F(\mathbf{s}) \\ &= n\ln\left(\sum_{j=1}^n x_j s_j\right) - \sum_{j=1}^n \ln(x_j) - \sum_{j=1}^n \ln(s_j) \\ &= n\ln\left(\sum_{j=1}^n x_j s_j\right) - \sum_{j=1}^n \ln(x_j s_j) \\ &\ge n\ln(n). \end{aligned}$$

The last line follows from the fact that the geometric mean lower bounds the arithmetic mean:

$$\frac{1}{n}\sum_{j=1}^{n} t_j \geq \left(\prod_{j=1}^{n} t_j\right)^{1/n}.$$

(Take logs and substitute $x_j s_j$ for t_j.)

This gives the inequality

$$\begin{aligned} & G_{n+\sqrt{n}}(\mathbf{x},\mathbf{s}) \geq n\ln(n) + \sqrt{n}\ln(\mathbf{x}^T\mathbf{s}) \\ \iff \quad & \ln(\mathbf{x}^T\mathbf{s}) \leq \frac{1}{\sqrt{n}} G_{n+\sqrt{n}}(\mathbf{x},\mathbf{s}) - \sqrt{n}\ln(n). \end{aligned}$$

Suppose that $G_{n+\sqrt{n}}(\mathbf{x},\mathbf{s}) \leq -\sqrt{n}\ln(1/\epsilon)$. Then

$$\begin{aligned} \ln(\mathbf{x}^T\mathbf{s}) &\leq -\ln(1/\epsilon) - \sqrt{n}\ln(n) \\ &= \ln(\epsilon) - \sqrt{n}\ln(n) \\ &\leq \ln(\epsilon). \end{aligned}$$

Taking exponentials on both sides gives $\mathbf{x}^T\mathbf{s} \leq \epsilon$. □

Claim 36 implies that, given an initial $\mathbf{x} \in \mathcal{F}^\circ(P)$ and $(\mathbf{y},\mathbf{s}) \in \mathcal{F}^\circ(D)$ and a method for reducing $G_{n+\sqrt{n}}$ by some fixed amount per iteration, the objective value will be within ϵ of optimal in $\mathcal{O}(\sqrt{n}\ln(1/\epsilon))$ iterations.

9 Conic programming

So far we've restricted our attention to linear programming, *i.e.*, to minimizing a linear objective function subject to linear constraints. Much LP theory extends to more general optimization problems, however. In fact, the standard form LP

$$\begin{array}{ll} \text{minimize} & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{array}$$

can be viewed as a special case of the conic programming problem,

$$\begin{array}{ll} \text{minimize} & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \in K \end{array} \tag{24}$$

where K is a convex cone.

Definition 23 (Convex cone) *A set K is a convex cone if for all $\mathbf{x}, \mathbf{y} \in K$ and for all $\lambda, \mu \geq 0$, $\lambda\mathbf{x} + \mu\mathbf{y} \in K$.*

Examples:

- The nonnegative orthant,

  $$\{\mathbf{x} \in \mathbf{R}^n \mid \mathbf{x} \geq \mathbf{0}\}.$$

  If K is the nonnegative orthant, then (24) reduces to the standard form primal LP.

- The second-order cone,

  $$\left\{\mathbf{x} \in \mathbf{R}^n \,\middle|\, \mathbf{x}^T \begin{bmatrix} I & 0 \\ 0 & -1 \end{bmatrix} \mathbf{x} \leq 0,\ x_n \geq 0 \right\}.$$

  The second-order cone is also known as the quadratic cone, the Lorentz cone, or the ice-cream cone. The second-order cone in $\mathbf{R}^3$ is

  $$\left\{ \begin{bmatrix} x \\ y \\ z \end{bmatrix} \,\middle|\, x^2 + y^2 \leq z^2,\ z \geq 0 \right\}.$$

  Figure 4 shows the boundary of this set.

- The positive semidefinite cone,

  $$\{X \in \mathbf{R}^{n \times n} \mid X \succeq 0\}.$$

  The notation $X \succeq 0$ means that X is a symmetric, positive semidefinite matrix, *i.e.*, $X^T = X$ and for all $\mathbf{v} \in \mathbf{R}^n$, $\mathbf{v}^T X \mathbf{v} \geq 0$. Equivalently, X is symmetric and all of its eigenvalues are nonnegative.

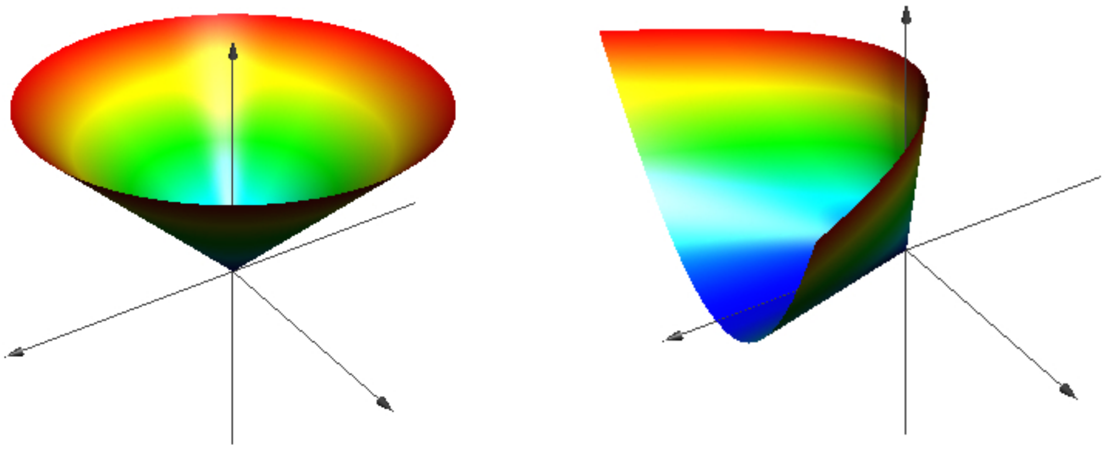

Figure 4: the boundaries of the second-order cone (left) and positive semidefinite cone (right). The second-order cone is also called the ice-cream cone. The cone of 2×2 positive semidefinite matrices is a subset of $\mathbf{R}^3$.

Note that the elements of the positive semidefinite cone are $n \times n$ symmetric matrices. Any such matrix is determined by $n(n+1)/2$ numbers. For example, the cone of 2×2 positive semidefinite matrices is a subset of $\mathbf{R}^3$:

$$\left\{ \begin{bmatrix} x & y \\ y & z \end{bmatrix} \middle| \, x, z \geq 0, \; y^2 \geq xz \right\}.$$

(To see that this is the positive semidefinite cone for $n = 2$, apply the nonnegative eigenvalue condition.) Figure 4 shows the boundary of this set.

The constraint and objective functions in problem (24) can be generalized to accommodate the matrix-valued optimization variable X (see §9.4).

9.1 Weak duality

In LP duality, we define the dual variables $(\mathbf{y}, \mathbf{s})$ such that $A^T\mathbf{y} + \mathbf{s} = \mathbf{c}$ and $\mathbf{s} \geq \mathbf{0}$. Recall the proof of weak duality: for any $\mathbf{x} \geq \mathbf{0}$ satisfying $A\mathbf{x} = \mathbf{b}$,

$$\mathbf{c}^T\mathbf{x} = (A^T\mathbf{y} + \mathbf{s})^T\mathbf{x} = (A\mathbf{x})^T\mathbf{y} + \mathbf{s}^T\mathbf{x} = \mathbf{b}^T\mathbf{y} + \mathbf{s}^T\mathbf{x} \geq \mathbf{b}^T\mathbf{y}$$

where the last line follows from $\mathbf{s} \geq \mathbf{0} \implies \mathbf{s}^T\mathbf{x} \geq 0$ for all $\mathbf{x} \geq \mathbf{0}$.

How to define the conic dual variables such that weak duality holds? Suppose $\mathbf{x} \in K$ satisfies $A\mathbf{x} = \mathbf{b}$. For the proof of weak duality to follow through, rather than requiring that $\mathbf{s}$ satisfy $\mathbf{s}^T\mathbf{x} \geq 0$ for all $\mathbf{x} \geq \mathbf{0}$, we need $\mathbf{s}$ to satisfy $\mathbf{s}^T\mathbf{x} \geq \mathbf{0}$ for all $\mathbf{x} \in K$. This motivates defining the **dual cone** of K,

$$K^* = \left\{ \mathbf{s} \in \mathbf{R}^n \mid \mathbf{s}^T\mathbf{x} \geq 0 \text{ for all } \mathbf{x} \in K \right\}.$$

We define the dual variables $(\mathbf{y}, \mathbf{s})$ such that $A^T\mathbf{y} + \mathbf{s} = \mathbf{c}$ and $\mathbf{s} \in K^*$. With this definition, weak duality holds: for all feasible $\mathbf{x}$, $\mathbf{c}^T\mathbf{x} \geq \mathbf{b}^T\mathbf{y}$.

The dual conic program is the problem of finding the greatest lower bound on the primal value:

$$\begin{array}{ll} \text{maximize} & \mathbf{b}^T\mathbf{y} \\ \text{subject to} & A^T\mathbf{y} + \mathbf{s} = \mathbf{c} \\ & \mathbf{s} \in K^*. \end{array} \tag{25}$$

Definition 24 (Self-dual) *A convex cone K is self-dual if $K^* = K$.*

Claim 37 *The nonnegative orthant, second-order cone and positive semidefinite cone are all self-dual.*

9.2 Infimum vs. minimum

If an LP and its dual are both feasible, then strong duality guarantees that some primal-feasible point is optimal. Conic programming provides no such guarantee. For example, consider the primal-dual pair

$$\begin{array}{ll} \text{minimize} & x_2 + x_3 \\ \text{subject to} & x_1 = 1 \\ & \mathbf{x} \in K \end{array} \qquad \begin{array}{ll} \text{maximize} & y \\ \text{subject to} & y + s_1 = 0, \;\; s_2 = s_3 = 1 \\ & \mathbf{s} \in K^* \end{array} \tag{26, 27}$$

where $K = K^*$ is the second-order cone in $\mathbf{R}^3$. The primal and dual feasible regions are

$$\begin{aligned} \mathcal{F}(P) &= \left\{\mathbf{x} \in \mathbf{R}^3 \mid x_1 = 1, \;\; x_1^2 + x_2^2 \leq x_3^2, \;\; x_3 \geq 0\right\} \\ &= \left\{\mathbf{x} \in \mathbf{R}^3 \mid 1 + x_2^2 \leq x_3^2, \;\; x_3 \geq 0\right\} \\ \mathcal{F}(D) &= \left\{(y, \mathbf{s}) \in \mathbf{R} \times \mathbf{R}^3 \mid y + s_1 = 0, \;\; s_2 = s_3 = 1, \;\; s_1^2 + s_2^2 \leq s_3^2, \;\; s_3 \geq 0\right\} \\ &= \left\{\left(0, \begin{bmatrix} 0 & 1 & 1 \end{bmatrix}^T\right)\right\} \end{aligned}$$

so both the primal and dual are feasible.

Observe that any $\mathbf{x} \in \mathcal{F}(P)$ satisfies

$$1 + x_2^2 \leq x_3^2 \quad \iff \quad (x_2 + x_3)(x_3 - x_2) \geq 1$$

so no $\mathbf{x}$ with objective value $x_2 + x_3 = 0$ is feasible. However, consider the solution $x_1 = 1$, $x_2 = \frac{1}{2}(\epsilon - \frac{1}{\epsilon})$, $x_3 = \frac{1}{2}(\epsilon + \frac{1}{\epsilon})$. This solution is feasible for any $\epsilon > 0$, since $x_3 > 0$ and

$$(x_2 + x_3)(x_3 - x_2) = (\epsilon)\,(1/\epsilon) = 1.$$

The objective value of this solution is $x_2 + x_3 = \epsilon$, which can be made arbitrarily close to zero while preserving feasibility. Thus, $\inf\{x_2 + x_3 \mid \mathbf{x} \in \mathcal{F}(P)\} = 0$, but no $\mathbf{x} \in \mathcal{F}(P)$ attains this value.

This example motivates defining the optimal value of the conic primal (24) as $\inf\{\mathbf{c}^T\mathbf{x} \mid A\mathbf{x} = \mathbf{b}, \;\; \mathbf{x} \in K\}$, rather than $\min\{\mathbf{c}^T\mathbf{x} \mid A\mathbf{x} = \mathbf{b}, \;\; \mathbf{x} \in K\}$. Similarly, we replace the maximum with the supremum when defining the optimal value of the conic dual (25).

This example also suggests that odd things can happen when we try to apply strong duality to conic programming.

9.3 Strong duality

Unlike weak duality, strong duality does not hold for a general conic program. Strong duality can be violated in various ways: the primal can have a finite value while the dual is infeasible, or the primal and dual can both have finite, but different, values. The weak link in the derivation of strong duality for conic programming is Farkas' lemma (see §4.1), which does not extend to general conic constraints.

We'd like to find special cases of conic programming where strong duality holds. We start by defining asymptotic feasibility.

Definition 25 (Asymptotically feasible) *The system $A\mathbf{x} = \mathbf{b}$, $\mathbf{x} \in K$ is asymptotically feasible if for all $\epsilon > 0$, there exists a $\Delta\mathbf{b}$ such that $\|\Delta\mathbf{b}\| < \epsilon$ and the system $A\mathbf{x} = \mathbf{b} + \Delta\mathbf{b}$, $\mathbf{x} \in K$ is feasible.*

In other words, an asymptotically feasible system can be made feasible by perturbing $\mathbf{b}$ by a tiny amount. The dual system $A^T\mathbf{y} + \mathbf{s} = \mathbf{c}$, $\mathbf{s} \in K^*$ is asymptotically feasible if for all $\epsilon > 0$, there exists a $\Delta\mathbf{c}$ such that $\|\Delta\mathbf{c}\| < \epsilon$ and the system $A^T\mathbf{y} + \mathbf{s} = \mathbf{c} + \Delta\mathbf{c}$, $\mathbf{s} \in K^*$ is feasible.

Theorem 38 (Asymptotic Farkas' lemma) *Let $A \in \mathbf{R}^{m\times n}$, $\mathbf{b} \in \mathbf{R}^m$, and K be a convex cone. Exactly one of the following holds:*

(1) *The system $A\mathbf{x} = \mathbf{b}$, $\mathbf{x} \in K$ is asymptotically feasible.*

(2) *The system $-A^T\mathbf{y} \in K^*$, $\mathbf{b}^T\mathbf{y} > 0$ is feasible.*

Definition 26 (Asymptotically optimal value) *If the primal (24) is asymptotically feasible, then its asymptotically optimal value is*

$$\texttt{a-opt} = \lim_{\epsilon\to 0} \inf_{\|\Delta\mathbf{b}\|<\epsilon} \inf \left\{\mathbf{c}^T\mathbf{x} \mid A\mathbf{x} = \mathbf{b} + \Delta\mathbf{b},\ \mathbf{x} \in K\right\}.$$

If the dual (25) is asymptotically feasible, then its asymptotically optimal value is

$$\texttt{a-dual-opt} = \lim_{\epsilon\to 0} \sup_{\|\Delta\mathbf{c}\|<\epsilon} \sup \left\{\mathbf{b}^T\mathbf{y} \mid \mathbf{c} + \Delta\mathbf{c} - A^T\mathbf{y} \in K^*\right\}.$$

Theorem 39 *If the primal is asymptotically feasible, then the optimal value of the dual equals* `a-opt`*. If the dual is asymptotically feasible, then the optimal value of the primal equals* `a-dual-opt`*.*

By weak duality and Theorem 39, if the primal and dual are asymptotically feasible, then

$$\texttt{a-opt} = \texttt{dual-opt} \le \texttt{opt} = \texttt{a-dual-opt}$$

where `opt` is the optimal value of the primal and `dual-opt` is the optimal value of the dual.

To find a case where all four values are equal, we define strong feasibility.

Definition 27 (Strongly feasible) *The system $A\mathbf{x} = \mathbf{b}$, $\mathbf{x} \in K$ is strongly feasible if there exists an $\epsilon > 0$ such that for all $\Delta\mathbf{b}$ with $\|\Delta\mathbf{b}\| < \epsilon$, the system $A\mathbf{x} = \mathbf{b} + \Delta\mathbf{b}$, $\mathbf{x} \in K$ is feasible.*

In other words, a strongly feasible system remains feasible under any sufficiently small perturbation of $\mathbf{b}$. The dual system $A^T\mathbf{y} + \mathbf{s} = \mathbf{c}$, $\mathbf{s} \in K^*$ is strongly feasible if there exists an $\epsilon > 0$ such that for all $\Delta\mathbf{c}$ with $\|\Delta\mathbf{c}\| < \epsilon$, the system $A^T\mathbf{y} + \mathbf{s} = \mathbf{c} + \Delta\mathbf{c}$, $\mathbf{s} \in K^*$ is feasible.

Note that if any $\mathbf{x} \in \operatorname{int}(K)$ satisfies $A\mathbf{x} = \mathbf{b}$, then the primal system is strongly feasible. Similarly, if any $\mathbf{s} \in \operatorname{int}(K^*)$ satisfies $A^T\mathbf{y} + \mathbf{s} = \mathbf{c}$, then the dual system is strongly feasible.

Theorem 40 *If either the primal or dual is strongly feasible, then strong duality holds.*

Corollary 41 *If the primal is feasible and the dual is strongly feasible, then the primal has an optimal solution.*

9.4 Semidefinite programming

Semidefinite programming is a class of conic programming that is widely used in control theory, statistics, and other fields. We discuss semidefinite programming in the context of the primal-dual pair

$$\begin{array}{ll} \text{minimize} & \langle C, X\rangle \\ \text{subject to} & \langle A_k, X\rangle = b_k,\ \ k \in \{1,\dots,m\} \\ & X \succeq 0 \end{array} \qquad \begin{array}{ll} \text{maximize} & \mathbf{b}^T\mathbf{y} \\ \text{subject to} & \sum_{k=1}^m y_k A_k + S = C \\ & S \succeq 0 \end{array} \tag{28, 29}$$

where $X^T = X \in \mathbf{R}^{n\times n}$ and $\langle C, X\rangle$ is the standard inner product on $\mathbf{R}^{n\times n}$,

$$\langle C, X\rangle = \sum_{i=1}^n \sum_{j=1}^n c_{ij}x_{ij} = \operatorname{tr}(C^T X).$$

We will develop a primal-dual path-following method for semidefinite programming, analogous to the LP algorithm in §8.4.

We begin by defining the feasible regions and their interiors,

$$\begin{aligned} \mathcal{F}(P) &= \left\{X \in \mathbf{R}^{n\times n} \mid A_k X = b_k,\ \ k \in \{1,\dots,m\},\ X \succeq 0\right\} \\ \mathcal{F}(D) &= \left\{(\mathbf{y}, S) \in \mathbf{R}^m \times \mathbf{R}^{n\times n} \,\middle|\, \sum_{k=1}^m y_k A_k + S = C,\ \ S \succeq 0\right\} \\ \mathcal{F}^\circ(P) &= \left\{X \in \mathbf{R}^{n\times n} \mid A_k X = b_k,\ \ k \in \{1,\dots,m\},\ X \succ 0\right\} \\ \mathcal{F}^\circ(D) &= \left\{(\mathbf{y}, S) \in \mathbf{R}^m \times \mathbf{R}^{n\times n} \,\middle|\, \sum_{k=1}^m y_k A_k + S = C,\ \ S \succ 0\right\}. \end{aligned}$$

where the notation $X \succ 0$ means that X is symmetric and positive definite, *i.e.*, $X^T = X$ and for all nonzero $\mathbf{v} \in \mathbf{R}^n$, $\mathbf{v}^T X \mathbf{v} > 0$. Equivalently, X is symmetric and all of its eigenvalues are positive.

We define the logarithmic barrier function for semidefinite programming as

$$F(X) = -\ln(\det(X)).$$

Why this definition? Recall that $\det(X) = \prod_{i=1}^n \lambda_i$, where $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of X. Since positive definite matrices have only positive eigenvalues, $\det(X) > 0$ for any $X \in \mathcal{F}^\circ(P)$. Any X on the boundary of $\mathcal{F}(P)$, however, is positive semidefinite but not positive definite. Hence, it must have at least one eigenvalue equal to zero, so its determinant is also zero. Thus, $F(X) \to \infty$ as X approaches the boundary of the feasible region. This is exactly the "repelling" behavior we want.

To negotiate the trade-off between making progress in the objective function and staying in the interior of the feasible region, we minimize over $\mathcal{F}^\circ(P)$ the function

$$B_\gamma(X) = \langle C, X \rangle + \gamma F(X)$$

parameterized by γ. For large values of γ, the minimizer of $B_\gamma(X)$ is near the analytic center of $\mathcal{F}^\circ(P)$. As $\gamma \to 0$, the minimizer of $B_\gamma(X)$ tends toward an optimal solution of (28).

Theorem 42 (Existence of a minimizer) *There exists a minimizer of $B_\gamma(X)$ on $\mathcal{F}^\circ(P)$ if and only if $\mathcal{F}^\circ(P)$ and $\mathcal{F}^\circ(D)$ are nonempty.*

Theorem 43 (Uniqueness of the minimizer) *Let $\mathcal{F}^\circ(P)$ and $\mathcal{F}^\circ(D)$ be nonempty. A matrix X is the unique minimizer of $B_\gamma(X)$ on $\mathcal{F}^\circ(P)$ if and only if there exists a $(\mathbf{y}, S) \in \mathcal{F}^\circ(D)$ such that*

$$\begin{aligned} &\sum_{k=1}^m y_k A_k + S = C \\ &\langle A_k, X \rangle = b_k, \ \ k \in \{1, \ldots, m\} \\ &XS = \gamma I. \end{aligned} \tag{30}$$

Solving system (30) amounts to finding a zero of the nonlinear function $\mathbf{f}_\gamma : \mathbf{R}^{n\times n} \times \mathbf{R}^m \times \mathbf{R}^{n\times n} \to \mathbf{R}^{2n^2+m}$, with the zero satisfying $X, S \succeq 0$. We can obtain $\mathbf{f}_\gamma(X, \mathbf{y}, S)$ by stacking the columns of the matrix equations in (30):

$$\mathbf{f}_\gamma(X, \mathbf{y}, S) = \begin{bmatrix} \sum_{k=1}^m y_k \mathbf{a}_1^k + \mathbf{s}_1 - \mathbf{c}_1 \\ \vdots \\ \sum_{k=1}^m y_k \mathbf{a}_n^k + \mathbf{s}_n - \mathbf{c}_n \\ \langle A_1 X \rangle - b_1 \\ \vdots \\ \langle A_m X \rangle - b_m \\ X\mathbf{s}_1 - \gamma \mathbf{e}_1 \\ \vdots \\ X\mathbf{s}_n - \gamma \mathbf{e}_n \end{bmatrix}$$

where $\mathbf{a}_i^k \in \mathbf{R}^n$ is the i^{th} column of A_k, $\mathbf{s}_i \in \mathbf{R}^n$ is the i^{th} column of S, and $\mathbf{c}_i \in \mathbf{R}^n$ is the i^{th} column of C:

$$A_k = \begin{bmatrix} \mathbf{a}_1^k & \dots & \mathbf{a}_n^k \end{bmatrix}, \quad S = \begin{bmatrix} \mathbf{s}_1 & \dots & \mathbf{s}_n \end{bmatrix}, \quad C = \begin{bmatrix} \mathbf{c}_1 & \dots & \mathbf{c}_n \end{bmatrix}.$$

Suppose that we're given an $X \in \mathcal{F}^\circ(P)$ and $(\mathbf{y}, S) \in \mathcal{F}^\circ(D)$ and that we've chosen $\sigma \in [0,1]$ somehow (see §8.4.1), computed $\mu = \frac{1}{n}\langle X, S\rangle$, and set $\gamma = \sigma\mu$. We can follow Newton's method by seeking a Newton direction $(\Delta X, \Delta\mathbf{y}, \Delta S)$, *i.e.*, a direction that satisfies

$$D_{\mathbf{f}}(X, \mathbf{y}, S) \begin{bmatrix} \Delta\mathbf{x}_1 \\ \vdots \\ \Delta\mathbf{x}_n \\ \Delta\mathbf{y} \\ \Delta\mathbf{s}_1 \\ \vdots \\ \Delta\mathbf{s}_n \end{bmatrix} + \mathbf{f}_\gamma(X, \mathbf{y}, S) = \mathbf{0} \tag{31}$$

where

$$\Delta X = \begin{bmatrix} \Delta\mathbf{x}_1 & \dots & \Delta\mathbf{x}_n \end{bmatrix}, \quad \Delta S = \begin{bmatrix} \Delta\mathbf{s}_1 & \dots & \Delta\mathbf{s}_n \end{bmatrix}$$

and $D_{\mathbf{f}}(X, \mathbf{y}, S)$ is the Jacobian of $\mathbf{f}_\gamma(X, \mathbf{y}, S)$. By computing the Jacobian and noting that $\mathbf{f}_\gamma$ simplifies significantly for feasible X and $(\mathbf{y}, S)$, one can show that system (31) is equivalent to

$$\begin{aligned} &\sum_{k=1}^{m} \Delta y_k A_k + \Delta S = 0 \\ &\langle A_k, \Delta X\rangle = 0, \ \ k \in \{1, \dots, m\} \\ &S\Delta X + X\Delta S = -SX + \gamma I. \end{aligned}$$

Once we've found a Newton direction, we update the iterates according to

$$\begin{aligned} X^+ &= X + \alpha\Delta X \\ \mathbf{y}^+ &= \mathbf{y} + \alpha\Delta\mathbf{y} \\ S^+ &= S + \alpha\Delta S. \end{aligned}$$

The step size α is chosen such that $X^+ \in \mathcal{F}^\circ(P)$ and $(\mathbf{y}^+, S^+) \in \mathcal{F}^\circ(D)$. We then compute a new γ and repeat.

This idea leads to an algorithm that takes $\mathcal{O}(\sqrt{n}\ln(c/\epsilon))$ iterations to reduce the duality gap from c to ϵ.