Tutorial on Robust Interior Point Method

May 21, 2021

We give a short, self-contained proof of the interior point method and its robust version. Consider the primal linear program

$$\min_{\mathbf{A}x=b, x\in\mathbb{R}^n_{\geq 0}} c^\top x \tag{P}$$

and its dual

$$\max_{\mathbf{A}^\top y+s=c, s\in\mathbb{R}^n_{\geq 0}} b^\top y. \tag{D}$$

where $\mathbf{A} \in \mathbb{R}^{d\times n}$ and $\mathbb{R}_{\geq} = \{x \geq 0\}$. The feasible regions for the two programs are

$$\mathcal{P} = \{x \in \mathbb{R}^n_{\geq} : \mathbf{A}x = b\} \text{ and } \mathcal{D} = \{s \in \mathbb{R}^n_{\geq} : \mathbf{A}^\top y + s = c \text{ for some } y\}.$$

We define their interiors:

$$\mathcal{P}^\circ = \{x \in \mathbb{R}^n_{>} : \mathbf{A}x = b\} \text{ and } \mathcal{D}^\circ = \{s \in \mathbb{R}^n_{>} : \mathbf{A}^\top y + s = c \text{ for some } y\}.$$

To motivate the main idea of the interior point method, we recall the optimality condition for linear programs.

Theorem 1 (Complementary Slackness)**.** *Any $x \in \mathcal{P}$ and $s \in \mathcal{D}$ are optimal if and only if $x^\top s = 0$. Moreover, if both $\mathcal{P}$ and $\mathcal{D}$ are non-empty, there exist $x^* \in \mathcal{P}$ and $s^* \in \mathcal{D}$ such that $(x^*)^\top s^* = 0$ and $x^* + s^* > 0$.*

More generally, the quantity $x^\top s$ measures the duality gap of the feasible solution:

Lemma 2 (Duality Gap)**.** *For any $x \in \mathcal{P}$ and $s \in \mathcal{D}$, the duality gap $c^\top x - b^\top y = x^\top s$. In particular $c^\top x \leq \min_{x\in\mathcal{P}} c^\top x + x^\top s$.*

Proof. Using $\mathbf{A}x = b$ and $\mathbf{A}^\top y + s = c$, we can compute the duality gap as follows

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

By weak duality, we have

$$c^\top x = b^\top y + x^\top s \leq \max_{\mathbf{A}^\top y+s=c, s\in\mathbb{R}^n_+} b^\top y + x^\top s \leq \min_{x\in\mathcal{P}} c^\top x + x^\top s.$$

$\square$

The main implication of Lemma 2 is that any feasible (x, s) with small $x^\top s$ is a nearly optimal solution of the linear program. This leads us to the primal-dual algorithms in which we start with a feasible primal and dual solution (x, s) and iteratively update the solution to decrease the duality gap $x^\top s$.

1 Interior Point Method

In this section, we discuss the classical short-step interior point method.

Definition 3 (Central Path)**.** We define the central path $(x_t, s_t) \in \mathcal{P}^\circ \times \mathcal{D}^\circ$ by $x_t s_t = t$. We say x_t is on the central path of (P) at t.

The algorithm maintains a pair $(x, s) \in \mathcal{P}^\circ \times \mathcal{D}^\circ$ and a scalar $t > 0$ satisfying the invariant $\|\frac{xs}{t} - 1\|_2 \leq \frac{1}{4}$. Note that the deviation from the central path is measured in ℓ_2 norm. In each step, it decreases t by a factor of $1 - \Omega(n^{-1/2})$ while maintaining the invariant.

1.1 Basic Property of a Step

To see why there is a pair $(x,s)\in\mathcal{P}^{\circ}\times\mathcal{D}^{\circ}$ satisfying the invariant, we prove the following generalization.

Lemma 4 (Quadrant Representation of Primal-Dual)**.** *Suppose $\mathcal{P}$ is non-empty and bounded. For any positive vector $\mu\in\mathbb{R}_{>}^{n}$, there is an unique pair $(x_{\mu},s_{\mu})\in\mathcal{P}^{\circ}\times\mathcal{D}^{\circ}$ such that $x_{\mu}s_{\mu}=\mu$. Furthermore, $x_{\mu}=\min_{x\in\mathcal{P}}f_{\mu}(x)$ where*

$$f_{\mu}(x)=c^{\top}x-\sum_{i=1}^{n}\mu_{i}\ln x_{i}.$$

Proof. Fix $\mu\in\mathbb{R}_{>}^{n}$. We define $x_{\mu}=\arg\min_{x\in\mathcal{P}}f_{\mu}(x)$ and prove that $(x_{\mu},s_{\mu})\in\mathcal{P}^{\circ}\times\mathcal{D}^{\circ}$ with $x_{\mu}s_{\mu}=\mu$ for some s_{μ}. Since $\mathcal{P}$ is non-empty and bounded and since f_{μ} is strictly convex, such unique x exists. Furthermore, since $f_{\mu}(x)\rightarrow+\infty$ as $x_{i}\rightarrow0$ for any i, we have that $x_{\mu}\in\mathcal{P}^{\circ}$.

By the KKT optimality condition for f_{μ} , there is a vector y such that

$$c-\frac{\mu}{x}=\mathbf{A}^{\top}y.$$

Define $s_{\mu}=\frac{\mu}{x_{\mu}}$, then one can check that $s_{\mu}\in\mathcal{D}$ and $x_{\mu}s_{\mu}=\mu$.

For the uniqueness, if $(x,s)\in\mathcal{P}^{\circ}\times\mathcal{D}^{\circ}$ and $xs=\mu$, then x satisfies the optimality condition for f_{μ}. Since f_{μ} is strictly convex, such x is unique. $\square$

Lemma 4 shows that any point in $\mathcal{P}^{\circ}\times\mathcal{D}^{\circ}$ is uniquely represented by a positive vector μ. Interior point methods move μ uniformly to 0 while maintaining the corresponding x_{μ}. Now we discuss how to find (x_{μ},s_{μ}) given a nearby interior feasible point (x,s). Namely, how to move (x,s) to $(x+\delta_{x},s+\delta_{s})$ such that it satisfies the equation

$$\begin{aligned}(x+\delta_{x})(s+\delta_{s})&=\mu,\\ \mathbf{A}(x+\delta_{x})&=b,\\ \mathbf{A}^{\top}(y+\delta_{y})+(s+\delta_{s})&=c,\\ (x+\delta_{x},s+\delta_{s})&\in\mathbb{R}_{>0}^{2n}.\end{aligned}$$

Although the equation above involves y, our approximate solution does not need to know y. By ignoring the inequality constraint and the second-order term $\delta_{x}\delta_{s}$ in the first equation above, and using $\mathbf{A}x=0$ and $\mathbf{A}^{\top}y+s=c$ we can simplify the system:

$$\begin{aligned}xs+\mathbf{S}\delta_{x}+\mathbf{X}\delta_{s}&=\mu,\\ \mathbf{A}\delta_{x}&=0,\\ \mathbf{A}^{\top}\delta_{y}+\delta_{s}&=0,\end{aligned}\tag{1.1}$$

where $\mathbf{X}$ and $\mathbf{S}$ are the diagonal matrix with diagonal x and s. In the following Lemma, we show how to write the step above using a projection matrix ($P^{2}=P$).

Lemma 5. *Suppose that $\mathbf{A}$ has full row rank and $(x,s)\in\mathcal{P}^{\circ}\times\mathcal{D}^{\circ}$. Then, the unique solution for the linear system (1.1) is given by*

$$\begin{aligned}\mathbf{X}^{-1}\delta_{x}&=(\mathbf{I}-\mathbf{P})(\delta_{\mu}/\mu),\\ \mathbf{S}^{-1}\delta_{s}&=\mathbf{P}(\delta_{\mu}/\mu)\end{aligned}$$

where $\delta_{\mu}=\mu-xs$ and $\mathbf{P}=\mathbf{S}^{-1}\mathbf{A}^{\top}(\mathbf{A}\mathbf{S}^{-1}\mathbf{X}\mathbf{A}^{\top})^{-1}\mathbf{A}\mathbf{X}$.

Proof. Note that the step satisfies $\mathbf{S}\delta_{x}+\mathbf{X}\delta_{s}=\delta_{\mu}$. Multiply both sides by $\mathbf{A}\mathbf{S}^{-1}$ and using $\mathbf{A}\delta_{x}=0$, we have

$$\mathbf{A}\mathbf{S}^{-1}\mathbf{X}\delta_{s}=\mathbf{A}\mathbf{S}^{-1}\delta_{\mu}.$$

Now, we use that $\mathbf{A}^{\top}\delta_{y}+\delta_{s}=0$ and get

$$\mathbf{A}\mathbf{S}^{-1}\mathbf{X}\mathbf{A}^{\top}\delta_{y}=-\mathbf{A}\mathbf{S}^{-1}\delta_{\mu}.$$

Since $\mathbf{A}\in\mathbb{R}^{m\times n}$ has full row rank and $\mathbf{S}^{-1}\mathbf{X}$ is invertible, we have that $\mathbf{A}\mathbf{S}^{-1}\mathbf{X}\mathbf{A}^{\top}$ is invertible and $\delta_{y}=-(\mathbf{A}\mathbf{S}^{-1}\mathbf{X}\mathbf{A}^{\top})^{-1}\mathbf{A}\mathbf{S}^{-1}\delta_{\mu}$ and

$$\delta_{s}=\mathbf{A}^{\top}(\mathbf{A}\mathbf{S}^{-1}\mathbf{X}\mathbf{A}^{\top})^{-1}\mathbf{A}\mathbf{S}^{-1}\delta_{\mu}.$$

Putting it into $\mathbf{S}\delta_{x}+\mathbf{X}\delta_{s}=\delta_{\mu}$, we have

$$\delta_{x}=\mathbf{S}^{-1}\delta_{\mu}-\mathbf{S}^{-1}\mathbf{X}\mathbf{A}^{\top}(\mathbf{A}\mathbf{S}^{-1}\mathbf{X}\mathbf{A}^{\top})^{-1}\mathbf{A}\mathbf{S}^{-1}\delta_{\mu}.$$

The result follows from the definition of $\mathbf{P}$. $\square$

1.2 Lower Bounding Step Size

The efficiency of interior point methods depends on how large a step we can take while staying within the domain. We first study the step operators $(\mathbf{I}-\mathbf{P})$ and $\mathbf{P}$. The following lemma implies that $\mathbf{P}$ is a nearly orthogonal projection matrix when μ is close to a multiple of the all-ones vector. Hence, the relative changes of $\mathbf{X}^{-1}\delta_x$ and $\mathbf{S}^{-1}\delta_s$ are essentially the orthogonal decomposition of the relative step δ_μ/μ on μ.

Lemma 6. *Under the assumption in Lemma 5, $\mathbf{P}$ is a projection matrix such that $\|\mathbf{P}v\|_\mu \leq \|v\|_\mu$ for any $v \in \mathbb{R}^n$. Similarly, we have that $\|(\mathbf{I}-\mathbf{P})v\|_\mu \leq \|v\|_\mu$.*

Proof. $\mathbf{P}$ is a projection because $\mathbf{P}^2 = \mathbf{P}$. Define the orthogonal projection

$$\mathbf{P}_{\mathrm{orth}} = \mathbf{S}^{-1/2}\mathbf{X}^{1/2}\mathbf{A}^\top(\mathbf{A}\mathbf{S}^{-1}\mathbf{X}\mathbf{A}^\top)^{-1}\mathbf{A}\mathbf{X}^{1/2}\mathbf{S}^{-1/2},$$

then we have

$$\begin{aligned}\|\mathbf{P}v\|_\mu^2 &= v^\top\mathbf{X}\mathbf{A}^\top(\mathbf{A}\mathbf{S}^{-1}\mathbf{X}\mathbf{A}^\top)^{-1}\mathbf{A}\mathbf{S}^{-1}\mathbf{X}\mathbf{S}\mathbf{S}^{-1}\mathbf{A}^\top(\mathbf{A}\mathbf{S}^{-1}\mathbf{X}\mathbf{A}^\top)^{-1}\mathbf{A}\mathbf{X}v\\ &= v^\top\mathbf{S}^{1/2}\mathbf{X}^{1/2}\mathbf{P}_{\mathrm{orth}}\mathbf{S}^{1/2}\mathbf{X}^{1/2}v\\ &\leq v^\top\mathbf{S}^{1/2}\mathbf{X}^{1/2}\mathbf{S}^{1/2}\mathbf{X}^{1/2}v = \|v\|_\mu^2.\end{aligned}$$

The calculation for $\|(\mathbf{I}-\mathbf{P})v\|_\mu$ is similar. $\square$

Next we give a lower bound on the largest feasible step size.

Lemma 7. *We have that $\|\mathbf{X}^{-1}\delta_x\|_\infty^2 \leq \frac{1}{\min_i \mu_i}\|\delta_\mu/\mu\|_\mu^2$ and $\|\mathbf{S}^{-1}\delta_s\|_\infty^2 \leq \frac{1}{\min_i \mu_i}\|\delta_\mu/\mu\|_\mu^2$. In particular, if $\|\delta_\mu/\mu\|_\mu^2 < \min_i \mu_i$, we have $(x+\delta_x, s+\delta_s) \in \mathcal{P}^\circ \times \mathcal{D}^\circ$*

Proof. For $\|\mathbf{X}^{-1}\delta_x\|_\infty$, we have $\min_i \mu_i \|\mathbf{X}^{-1}\delta_x\|_\infty^2 \leq \|\mathbf{X}^{-1}\delta_x\|_\mu^2$ and hence

$$\|\mathbf{X}^{-1}\delta_x\|_\infty^2 \leq \frac{1}{\min_i \mu_i}\|\mathbf{X}^{-1}\delta_x\|_\mu^2 = \frac{1}{\min_i \mu_i}\|(\mathbf{I}-\mathbf{P})(\delta_\mu/\mu)\|_\mu^2 \leq \frac{1}{\min_i \mu_i}\|\delta_\mu/\mu\|_\mu^2.$$

The proof for $\|\mathbf{S}^{-1}\delta_s\|_\infty$ is similar.

Hence, if $\|\delta_\mu/\mu\|_\mu^2 < \min_i \mu_i$, we have that $\|\mathbf{X}^{-1}\delta_x\|_\infty < 1$ and $\|\mathbf{S}^{-1}\delta_s\|_\infty < 1$. Therefore, $x+\delta_x$ and $s+\delta_s$ are feasible. $\square$

To decrease μ uniformly, we set $\delta_\mu = -h\mu$ for some step size h. To ensure the feasibility, we need $\|\delta_\mu/\mu\|_\mu^2 \leq \min_i \mu_i$ and this gives the maximum step size

$$h = \sqrt{\frac{\min_i \mu_i}{\sum_i \mu_i}}. \tag{1.2}$$

Note that the above quantity is maximized at $h = n^{-1/2}$ when μ has all equal coordinates.

1.3 Staying within small ℓ_2 distance

Since the step size (1.2) maximizes when μ is a constant vector. A natural approach is to control μ vector ℓ_2 close to a constant vector. This motivates the following algorithm:

Algorithm 1: `L2Step`$(\mathbf{A}, x, s, t_{\mathrm{start}}, t_{\mathrm{end}})$

Define $\mathbf{P}_{x,s} = \mathbf{S}^{-1}\mathbf{A}^\top(\mathbf{A}\mathbf{S}^{-1}\mathbf{X}\mathbf{A}^\top)^{-1}\mathbf{A}\mathbf{X}$.
Invariant: $(x,s) \in \mathcal{P}^\circ \times \mathcal{D}^\circ$ and $\|xs - t\|_2 \leq \frac{t}{4}$.
Let $t = t_{\mathrm{start}}$, $h = 1/(16\sqrt{n})$ and n is the number of columns in $\mathbf{A}$.
repeat
  Let $t' = \max(t/(1+h), t_{\mathrm{end}})$
  Let $\mu = xs$ and $\delta_\mu = t' - \mu$.
  Let $\delta_x = \mathbf{X}(\mathbf{I}-\mathbf{P}_{x,s})(\delta_\mu/\mu)$ and $\delta_s = \mathbf{S}\mathbf{P}_{x,s}(\delta_\mu/\mu)$.
  Set $x \leftarrow x + \delta_x$, $s \leftarrow s + \delta_s$ and $t \leftarrow t'$.
until $t \neq t_{\mathrm{end}}$;
Return (x, s)

Note that the algorithm requires some initial point (x, s) close to the central path and we will show how to get it by changing the linear program temporarily. First, we show that the invariant is maintained in each step. The conclusion distance less than $t/6$ is needed in Section A where we call `L2Step` on a modified LP, then prove the result is close to central path for the original LP.

Lemma 8. *Suppose that the input satisfies $(x, s) \in \mathcal{P}^\circ \times \mathcal{D}^\circ$ and $\|xs - t_{\text{start}}\|_2 \leq \frac{t_{\text{start}}}{4}$. Then, the algorithm* `L2Step` *maintains $(x, s) \in \mathcal{P}^\circ \times \mathcal{D}^\circ$ and t such that $\|xs - t\|_2 \leq \frac{t}{6}$.*

Proof. We prove by induction that $\|xs - t\|_2 \leq \frac{t}{6}$ after each step. Note that the input satisfies $\|xs - t\|_2 \leq \frac{t}{4}$.

Let $x' = x + \delta_x$, $s' = s + \delta_s$ and t' defined in the algorithm. Note that

$$\begin{aligned} x's' - t' =& (x + \delta_x)(s + \delta_s) - t' \\ =& \mu + \mathbf{S}\delta_x + \mathbf{X}\delta_s + \delta_x\delta_s - t'. \end{aligned}$$

Lemma 5 shows that $\mathbf{S}\delta_x + \mathbf{X}\delta_s = t' - \mu$. Hence, we have

$$x's' - t' = \delta_x\delta_s = \mathbf{X}^{-1}\delta_x \cdot \mathbf{S}^{-1}\delta_s \cdot \mu$$

Using this, we have

$$\begin{aligned} \|x's' - t'\|_2 &\leq \|\mu^{1/2}\mathbf{X}^{-1}\delta_x\|_2 \|\mu^{1/2}\mathbf{S}^{-1}\delta_s\|_2 \\ &= \|\mathbf{X}^{-1}\delta_x\|_\mu \|\mathbf{S}^{-1}\delta_s\|_\mu \\ &\leq \|\delta_\mu/\mu\|_\mu^2 \end{aligned}$$

where we used Lemma 6 at the end.

Using $t' - \mu = \frac{t'}{t}(t - \mu) + (\frac{t'}{t} - 1)\mu$, we have

$$\begin{aligned} \|\delta_\mu/\mu\|_\mu &= \|\frac{t'}{t}\frac{t - \mu}{\mu} + (\frac{t'}{t} - 1)\|_\mu \\ &\leq \frac{t'}{t}\|xs - t\|_{\mu^{-1}} + \|\frac{t'}{t} - 1\|_\mu. \end{aligned}$$

Since $\|\mu - t\|_2 \leq \frac{t}{4}$, we have $\min_i \mu_i \geq \frac{3t}{4}$ and $\max_i \mu_i \leq \frac{5}{4}t$. Using $|\frac{t'}{t} - 1| \leq h = \frac{16}{\sqrt{n}}$, we have

$$\|\delta_\mu/\mu\|_\mu \leq \frac{t'}{t}\sqrt{\frac{4}{3t}}\|xs - t\|_2 + h\sqrt{\frac{5}{4}t} \leq \sqrt{\frac{t}{12}} + h\sqrt{\frac{5}{4}t} \leq 0.38\sqrt{t}.$$

Hence, we have $\|x's' - t'\|_2 \leq \|\delta_\mu/\mu\|_\mu^2 \leq 0.15t \leq t'/6$. Furthermore, $\|\delta_\mu/\mu\|_\mu^2 < \min_i \mu_i$ which implies (x, s) is feasible (Lemma 7). □

The following theorem only concludes the output is close to central path. To upper bound the error, we can apply Lemma 2 which shows the duality gap is equal to $x^\top s$.

1.4 Solving LP Approximately and Exactly

Now we discuss how to get the interior point by modifying . The runtime of interior point method depends on how degenerate the linear program is.

Definition 9. We define the following parameters for the linear program $\min_{\mathbf{A}x=b, x\geq 0} c^\top x$:

1. Inner radius r: There exists a $x \in \mathcal{P}$ such that $x_i \geq r$ for all i.
2. Outer radius R: For any $x \geq 0$ with $\mathbf{A}x = b$, we have that $\|x\|_2 \leq R$.
3. Lipschitz constant L: $\|c\|_2 \leq L$.

Since `L2Step` requires an explicit central path, we modify the linear program to make it happen. To satisfy the constraint $\mathbf{A}x = b$, we start the algorithm by taking a least square solution of the constraint $\mathbf{A}x = b$. Since it can be negative, we write the variable $x = x^+ - x^-$ with both $x^+, x^- \geq 0$. We put a large cost vector on x^- to ensure the solution is roughly the same. The crux of the proof is that if we optimize this new program well enough, we will have $x^+ - x^- > 0$ and hence this gives a good starting point of the original program. Due to technical reasons, we need to put an extra constraint $1^\top x^+ \leq \Lambda$ for some Λ to ensure the problem is bounded. The precise formulation of the modified linear program is as follows:

Definition 10 (Modified Linear Program)**.** Given a linear program $\min_{\mathbf{A}x=b,x\in\mathbb{R}^n_{\geq}} c^\top x$ with inner radius r, outer radius R and Lipschitz constant L. For any $\overline{R} \geq 10R$, $t \geq 8L\overline{R}$, we define the modified primal linear program by

$$\min_{(x^+,x^-,\theta)\in\mathcal{P}_{t,\overline{R}}} c^\top x^+ + d^\top x^-$$

where $\mathcal{P}_{t,\overline{R}} = \{(x^+,x^-,\theta) \in \mathbb{R}^{2n+1}_{\geq 0} : \mathbf{A}(x^+ - x^-) = b, \sum_{i=1}^n x_i^+ + \theta = \Lambda\}$, $d = t/x_c^-$, $x_c^- = x_c^+ - \mathbf{A}^\top(\mathbf{A}\mathbf{A}^\top)^{-1}b$, $x_c^+ = \frac{t}{c+t/\overline{R}}$, $\Lambda = \sum_i x_{c,i}^+ + \overline{R}$. We define the corresponding dual polytope by

$$\mathcal{D}_{t,\overline{R}} = \{(s^+,s^-,s^\theta) \in \mathbb{R}^{2n+1}_{\geq 0} : \mathbf{A}^\top y + \lambda + s^+ = c, \mathbf{A}^\top y + s^- = d, \lambda + s^\theta = 0 \text{ for some } y \in \mathbb{R}^d \text{ and } \lambda \in \mathbb{R}\}.$$

The main result about the modified program is the following.

Theorem 11. *Given a linear program $\min_{\mathbf{A}x=b,x\in\mathbb{R}^n_{\geq}} c^\top x$ with inner radius r, outer radius R and Lipschitz constant L. For any $0 \leq \epsilon \leq \frac{1}{2}$, the modified linear program* **??** *with $\overline{R} = \frac{5}{\epsilon}R$, $t = 2^{16}\epsilon^{-3}n^2\frac{R}{r}\cdot LR$ has the following properties:*

- *The point $(x_c^+, x_c^-, \overline{R})$ is on the central path of the modified program at t.*
- *For any primal $x \stackrel{\text{def}}{=} (x^+,x^-,\theta) \in \mathcal{P}_{t,\overline{R}}$ and dual $s \stackrel{\text{def}}{=} (s^+,s^-,s^\theta) \in \mathcal{D}_{t,\overline{R}}$ such that $\frac{5}{6}LR \leq x_i s_i \leq \frac{7}{6}LR$, we have that*
$$(x^+ - x^-, s^+ - s^\theta) \in \mathcal{P}\times\mathcal{D}$$
and that $x_i^- \leq \epsilon x_i^+$ and $s^\theta \leq \epsilon s_i^+$ for all i.

Proof. Since the proof is a bit complicated and not illuminating, we defer the proof to Appendix A (Lemma 23 and Lemma 29). □

Now, we state our main algorithm:

Algorithm 2: `L2LPApproximate`$(\mathbf{A}, b, c, x^{(0)}, \epsilon)$

Assume the linear program has inner radius r, outer radius R and Lipschitz constant L.
Let $\epsilon = 1/(100\sqrt{n})$, $\overline{R} = \frac{5}{\epsilon}R$, $t = 2^{16}\epsilon^{-3}n^2\frac{R}{r}\cdot LR$.
`// Define the modified program` $\min_{\overline{\mathbf{A}}x=\overline{b}} \overline{c}^\top x$ `by Definition 10 with parameters` $\overline{R}$ `and` t`.`
Let $\overline{\mathbf{A}} = \begin{bmatrix} \mathbf{A} & -\mathbf{A} & 0 \\ 1 & 0 & 0 \end{bmatrix}$, $\overline{c} = [c, d]$, $\overline{b} = [b; \Lambda]$ where d and Λ are defined in Definition 10.
`// Write down the central path at` t `for modified linear program using Lemma 23.`
$\overline{x} = (x_c^+, x_c^-, \overline{R})$. $\overline{s} = x/t$.
$(\overline{x}, \overline{s}) =$ `L2Step`$(\overline{\mathbf{A}}, \overline{x}, \overline{s}, t, LR)$.
$(x, s) = (x^+ - x^-, s^+ - s^\theta)$ where $\overline{x} = (x^+, x^-, \theta)$ and $\overline{s} = (s^+, s^-, s^\theta)$.
$(x_{\text{end}}, s_{\text{end}}) =$ `L2Step`$(\mathbf{A}, x^+ - x^-, s^+ - s^\theta, LR, t_{\text{end}})$ with $t_{\text{end}} = \epsilon LR/(2n)$.
Return x_{end}.

Theorem 12. *Consider a linear program $\min_{\mathbf{A}x=b,x\geq 0} c^\top x$ with n variables and d constraints. Assume the linear program has inner radius r, outer radius R and Lipschitz constant L (See Definition 9),* `L2LPApproximate` *outputs x such that*

$$\begin{aligned} c^\top x &\leq \min_{\mathbf{A}x=b,x\geq 0} c^\top x + \epsilon LR, \\ \mathbf{A}x &= b, \\ x &\geq 0. \end{aligned}$$

The algorithm takes $O(\sqrt{n}\log(nR/(\epsilon r)))$ Newton steps (defined in (1.1)).

If we further assume that the solution $x^* = \arg\min_{\mathbf{A}x=b,x\geq 0} c^\top x$ is unique and that $c^\top x \geq c^\top x^* + \eta LR$ for any other vertex x of $\{\mathbf{A}x = b, x \geq 0\}$, then we have that $\|x - x^*\|_2 \leq \frac{2\epsilon R}{\eta}$.

Proof. By Theorem 11, $(x_c^+, x_c^-, \overline{R})$ is on the central path of the modified program at t. After the first call of `L2Step`, Lemma 8 shows that `L2Step` returns $(\overline{x}, \overline{s})$ such that $\|\overline{x}\overline{s} - t\|_2 \leq \frac{t}{6}$ with $t = LR$.

Theorem 11 shows that $(x, s) = (x^+ - x^-, s^+ - s^\theta) \in \mathcal{P}\times\mathcal{D}$ and that $x = (1 \pm \epsilon)x^+$ and $s = (1 \pm \epsilon)s^+$. Since $\epsilon = \frac{1}{100\sqrt{n}}$ and $\|x^+s^+ - t\|_2 \leq \frac{t}{6}$, we have that $\|xs - t\|_2 \leq \frac{t}{6}$. This satisfies the condition for the second call of `L2Step`.

After the second call of `L2Step`, Lemma 8 shows that `L2Step` returns $(x_{\text{end}}, s_{\text{end}})$ such that $\|x_{\text{end}} s_{\text{end}} - t_{\text{end}}\|_2 \leq \frac{t}{6}$ with $t_{\text{end}} = \epsilon LR/(2n)$. Hence, Lemma 2 shows that

$$c^\top x_{\text{end}} \leq \min_{\mathbf{A}x=b, x\geq 0} c^\top x + x_{\text{end}}^\top s_{\text{end}} \leq \min_{\mathbf{A}x=b, x\geq 0} c^\top x + 2t_{\text{end}} n \leq \min_{\mathbf{A}x=b, x\geq 0} c^\top x + \epsilon LR.$$

For the runtime, note that `L2Center` decrease t by $1 - \Omega(n^{-1/2})$ factor each step. Hence, the first call takes $O(\sqrt{n}\log(nR/r))$ Newton steps and second call takes $O(\sqrt{n}\log(n/\epsilon))$ Newton steps.

For the last conclusion, we assume $\epsilon \leq \eta$ and let $\mathcal{P}_t = \mathcal{P} \cap \{c^\top x \leq c^\top x^* + tLR\}$. Note that $\mathcal{P}_\eta$ is a cone at x^* (because there is no vertex except x^* with value less than $c^\top x^* + tLR$). Hence, we have $\mathcal{P}_\epsilon - x^* = \frac{\epsilon}{\eta}(\mathcal{P}_\eta - x^*)$. Since $x \in \mathcal{P}_\epsilon$, we have that

$$\|x - x^*\|_2 \leq \frac{\epsilon}{\eta}\text{diameter}(\mathcal{P}_\eta - x^*) \leq \frac{2\epsilon R}{\eta}.$$

□

If we know the solution of the linear program is integral or rational with some bound on the number of bits, then getting a solution close enough to x^* allows us to round the solution to integral. Therefore, the last conclusion of last theorem gives us an exact linear program algorithm assuming $\mathbf{A}, b, c$ are integral and bounded. The uniqueness assumption can be achieved by perturbing the cost vector by a random vector (e.g., using the "isolation" lemma [1, Lemma 4]).

2 Robust Interior Point Method

To improve the interior point method, one can either improve the number of steps $\widetilde{O}(\sqrt{n})$ or the cost per step. In this note, we focus on the later question. Recall from (1.1) that the linear system we solve in each step is of the form

$$\begin{aligned}
\mathbf{S}\delta_x + \mathbf{X}\delta_s &= \delta_\mu, \\
\mathbf{A}\delta_x &= 0, \\
\mathbf{A}^\top \delta_y + \delta_s &= 0.
\end{aligned} \tag{2.1}$$

In each step, x, s and δ_μ in the equation above changes relatively by a vector with bounded ℓ_2 norm. So, only few coordinates change a lot in each step. To take advantage of this, the robust interior point method contains two new components: 1) Analyze the convergence when we only solve the linear system approximately (Section 2.1). 2) Show how to maintain the solution throughout the iteration (Section 2.3).

2.1 Staying within small ℓ_∞ distance

In the above description and analysis, we assumed that we computed each step of the interior point method precisely. But one can imagine that it suffices to compute steps approximately, since our goal is only to stay close to the central path. This could have significant computational advantages.

To make the interior point method robust to noise in the updates to x and s, we need the interior point method to work under a larger neighborhood than that given by the Euclidean norm ($\|xs - t\|_2 \leq \frac{t}{4}$). One natural choice of distance and potential would be a higher norm, $\|xs - t\|_q^q$. However, analyzing the step δ_μ that minimizes $\|\mu + \delta_\mu - t\|_q^q$ involves many cases. Instead, we pick the potential

$$\Phi(v) = \sum_{i=1}^n \cosh(\lambda v_i) = \sum_{i=1}^n \frac{(e^{\lambda v_i} + e^{-\lambda v_i})}{2}. \tag{2.2}$$

for a scalar $\lambda > 0$. This potential induces the following algorithm.

Algorithm 3: $\texttt{RobustStep}(\mathbf{A}, x, s, t_{\text{start}}, t_{\text{end}})$

Define $r = (xs - t)/t$ and Φ according to (2.2) with $\lambda = 16 \log 40n$.
Invariant: $(x, s) \in \mathcal{P}^\circ \times \mathcal{D}^\circ$ and $\Phi(r) \leq 16n$.
Let $t = t_{\text{start}}$, $h = 1/(128\lambda\sqrt{n})$ and n is the number of columns in $\mathbf{A}$.
repeat

Pick $\overline{x}$, $\overline{s}$ and $\overline{r}$ such that $\|\ln \overline{x} - \ln x\|_\infty \leq \frac{1}{48}$, $\|\ln \overline{s} - \ln s\|_\infty \leq \frac{1}{48}$ and $\|\overline{r} - r\|_\infty \leq \frac{1}{48\lambda}$.
Let $t' = \max(t/(1+h), t_{\text{end}})$, $\overline{\delta}_\mu = -\frac{t'}{32\lambda}\frac{\overline{g}}{\|\overline{g}\|_2}$, $\overline{g} = \nabla\Phi(\overline{r})$.
Find δ_x, δ_s such that

$$\begin{aligned} \overline{\mathbf{S}}\delta_x + \overline{\mathbf{X}}\delta_s &= \overline{\delta}_\mu, \\ \mathbf{A}\delta_x &= 0, \\ \mathbf{A}^\top \delta_y + \delta_s &= 0. \end{aligned} \tag{2.3}$$

Set $x \leftarrow x + \delta_x$, $s \leftarrow s + \delta_s$ and $t \leftarrow t'$.

until $t \neq t_{\text{end}}$;
Return (x, s)

Here, we prove the key facts we use about Φ:

Lemma 13. *Define Φ according to (2.2). For any $v \in \mathbb{R}^n$, we have that $\|v\|_\infty \leq \frac{\log 2\Phi(v)}{\lambda}$ and $\|\nabla\Phi(v)\|_2 \geq \frac{\lambda}{\sqrt{n}}(\Phi(v) - n)$. Moreover, if $\Phi(v) \geq 4n$ and $\|\delta\|_\infty \leq \frac{1}{5\lambda}$, we have*

$$\|\nabla\Phi(v + \delta) - \nabla\Phi(v)\|_2 \leq \frac{1}{3}\|\nabla\Phi(v)\|_2.$$

Proof. We have $\Phi(v) \geq \frac{1}{2}\min_i e^{\lambda|v_i|}$ and hence $\|v\|_\infty \leq \frac{\log 2\Phi(v)}{\lambda}$.

For the second claim, we have

$$\begin{aligned} \|\nabla\Phi(v)\|_2 &= \lambda\sqrt{\sum_{i=1}^n \sinh^2(\lambda v_i)} = \lambda\sqrt{\sum_{i=1}^n (\cosh^2(\lambda v_i) - 1)} \\ &\geq \frac{\lambda}{\sqrt{n}}\sum_{i=1}^n \sqrt{\cosh^2(\lambda v_i) - 1} \geq \frac{\lambda}{\sqrt{n}}\sum_{i=1}^n (\cosh(\lambda v_i) - 1) \\ &= \frac{\lambda}{\sqrt{n}}(\Phi(v) - n). \end{aligned}$$

For the last claim, using $\sinh(v + \delta) = \sinh v \cosh \delta + \cosh v \sinh \delta$ and $|\cosh v - \sinh v| \leq 1$, for $|\delta| \leq \frac{1}{5}$, we have

$$\begin{aligned} |\sinh(v + \delta) - \sinh(v)| &\leq |\cosh\delta - 1| \cdot |\sinh v| + |\sinh\delta| \cdot \cosh v \\ &\leq (|\cosh\delta - 1| + |\sinh\delta|) \cdot |\sinh v| + |\sinh\delta| \\ &\leq \frac{1}{4}|\sinh v| + \frac{1}{4}. \end{aligned}$$

Using that $\nabla\Phi(v) = \sum_{i=1}^n \lambda \sinh(\lambda v_i)$, for $\|\delta\|_\infty \leq \frac{1}{5\lambda}$, we have

$$\|\nabla\Phi(v + \delta) - \nabla\Phi(v)\|_2 \leq \frac{1}{4}\|\nabla\Phi(v)\|_2 + \frac{\sqrt{n}\lambda}{4}. \tag{2.4}$$

Since $\Phi(v) \geq 4n$, we have that $\|\nabla\Phi(v)\|_2 \geq 3\sqrt{n}\lambda$ and hence (2.4) shows that

$$\|\nabla\Phi(v + \delta) - \nabla\Phi(v)\|_2 \leq (\frac{1}{4} + \frac{1}{12})\|\nabla\Phi(v)\|_2 = \frac{1}{3}\|\nabla\Phi(v)\|_2.$$

$\square$

We collect some basic bounds on the step in the following Lemma:

Lemma 14. *Using the notation in* `RobustStep` *(Algorithm 3). Under the invariant* $\Phi((xs-t)/t) \leq 16n$*, we have* $\|xs-t\|_\infty \leq \frac{t}{16}$*,* $\|\delta_x/x\|_2 \leq \frac{1}{16\lambda}$*, and* $\|\delta_s/s\|_2 \leq \frac{1}{16\lambda}$*.*

Proof. Using $\Phi((xs-t)/t) \leq 16n$ and Lemma 13, we have

$$\|xs-t\|_\infty \leq \frac{t \log 32n}{\lambda} \leq \frac{t}{16}$$

By Lemma 5, we have

$$\mathbf{X}^{-1}\delta_x = (\mathbf{I}-\mathbf{P})(\overline{\delta}_\mu/\overline{\mu})$$

where $\overline{\mu} = \overline{xs}$ and $\mathbf{P} = \overline{\mathbf{S}}^{-1}\mathbf{A}^\top(\mathbf{A}\overline{\mathbf{S}}^{-1}\overline{\mathbf{X}}\mathbf{A}^\top)^{-1}\mathbf{A}\overline{\mathbf{X}}$. By Lemma 6, we have

$$\|\delta_x/x\|_{\overline{\mu}} = \|(\mathbf{I}-\mathbf{P})v\|_{\overline{\mu}} \leq \|\overline{\delta}_\mu/\overline{\mu}\|_{\overline{\mu}}.$$

Using that $\|xs-t\|_\infty \leq \frac{t}{16}$, $\|\ln\overline{x} - \ln x\|_\infty \leq \frac{1}{48}$, $\|\ln\overline{s} - \ln s\|_\infty \leq \frac{1}{48}$, we have $\overline{\mu} \geq \frac{10}{11}t$ and hence

$$\|\delta_x/x\|_2 \leq \sqrt{\frac{11}{10t}}\|\delta_x/x\|_{\overline{\mu}} \leq \sqrt{\frac{11}{10t}}\|\overline{\delta}_\mu\|_{\overline{\mu}^{-1}} \leq \frac{11}{10t}\|\overline{\delta}_\mu\|_2$$

Using the formula $\overline{\delta}_\mu = t' - t - \frac{t'}{32\lambda}\frac{\overline{g}}{\|\overline{g}\|_2}$ and $|t'-t| \leq \frac{t'}{128\lambda\sqrt{n}}$, we have

$$\|\delta_x/x\|_2 \leq \frac{11}{10}\left(\|\frac{t'-t}{t}\|_2 + \frac{t'}{100\lambda t}\right) \leq \frac{1}{16\lambda}.$$

Same proof gives $\|\delta_s/s\|_2 \leq \frac{1}{16\lambda}$. $\square$

Using this, we prove the algorithm `RobustStep` satisfies the invariant on the distance.

Now, we are ready to prove the main theorem.

Lemma 15. *Suppose that the input satisfies* $(x,s) \in \mathcal{P}^\circ \times \mathcal{D}^\circ$ *and* $\Phi((xs - t_{\text{start}})/t_{\text{start}}) \leq 16n$*. Let* $x^{(k)}, s^{(k)}, t^{(k)}$ *be the* x, s, t *computed in the* `RobustStep` *after the* k*-th step. Let* $\Phi^{(k)} = \Phi((x^{(k)}s^{(k)} - t^{(k)})/t^{(k)})$*. Then, we have*

$$\Phi^{(k+1)} \leq \begin{cases} 12n & \text{if } \Phi^{(k)} \leq 8n \\ \Phi^{(k)} & \text{otherwise} \end{cases}.$$

Furthermore, we have that $\|r^{(k+1)} - r^{(k)}\|_2 \leq \frac{1}{5\lambda}$ *where* $r^{(k)} = (x^{(k)}s^{(k)} - t^{(k)})/t^{(k)}$*.*

Proof. Fix some iteration k. Let $x = x^{(k)}, s = s^{(k)}, t = t^{(k)}, x' = x^{(k+1)}, s' = s^{(k+1)}$ and $t' = t^{(k+1)}$. We define $r = (xs-t)/t$ and $r' = (x's'-t')/t'$.

By the definition of δ_x and δ_s, we have

$$\overline{\mathbf{S}}\delta_x + \overline{\mathbf{X}}\delta_s = \overline{\delta}_\mu = t' - t - \frac{t'}{32\lambda}\frac{\overline{g}}{\|\overline{g}\|_2}$$

and hence

$$\begin{aligned}
\frac{x's'-t'}{t'} =& \frac{(x+\delta_x)(s+\delta_s)-t'}{t'} \\
=& \frac{xs + s\delta + x\delta_s + \delta_x\delta_s - t'}{t'} \\
=& \frac{xs - t' + s\delta_x + x\delta_s + (s-\overline{s})\delta_x + (x-\overline{x})\delta_s + \delta_x\delta_s}{t'} \\
=& \frac{xs-t}{t} - \frac{1}{32\lambda}\cdot\frac{\overline{g}}{\|\overline{g}\|_2} + \eta
\end{aligned} \tag{2.5}$$

where the error term

$$\eta = \frac{t-t'}{t} + (\frac{t}{t'} - 1)\frac{xs-t}{t} + \frac{(s-\overline{s})\delta_x + (x-\overline{x})\delta_s + \delta_x\delta_s}{t'}.$$

Now, we bound the error term η. Using Lemma 14 ($\|\delta_x/x\|_2 \leq \frac{1}{16\lambda}$, $\|\delta_s/s\|_2 \leq \frac{1}{16\lambda}$, $\|xs-t\|_\infty \leq \frac{t}{16}$) and the definition of the algorithm ($\lambda \geq 16$, $|t'-t| \leq \frac{t'}{128\lambda\sqrt{n}} \leq \frac{t}{128\lambda\sqrt{n}}$, $\|\ln \overline{x} - \ln x\|_\infty \leq \frac{1}{48}$, $\|\ln \overline{s} - \ln s\|_\infty \leq \frac{1}{48}$), we have

$$\begin{aligned}\|\eta\|_2 \leq & |\frac{t'}{t} - 1|\sqrt{n} + |\frac{t}{t'} - 1|\|\frac{xs-t}{t}\|_\infty\sqrt{n} + \|\frac{xs}{t'}\|_\infty\|\frac{s-\overline{s}}{s}\|_\infty\|\frac{\delta_x}{x}\|_2 \\ & + \|\frac{xs}{t'}\|_\infty\|\frac{x-\overline{x}}{x}\|_\infty\|\frac{\delta_s}{s}\|_2 + \|\frac{xs}{t'}\|_\infty\|\frac{\delta_x}{x}\|_2\|\frac{\delta_s}{s}\|_2 \\ \leq & \frac{1}{128\lambda} + \frac{1}{128\lambda}\frac{1}{16} + \frac{9}{8}(e^{1/48}-1)(\frac{1}{16\lambda} + \frac{1}{16\lambda}) + \frac{9}{8}(\frac{1}{16\lambda}\frac{1}{16\lambda}) \leq \frac{1}{80\lambda}. \end{aligned} \tag{2.6}$$

In particular, we use (2.5) and (2.6) to get

$$\|r - r'\|_2 \leq \frac{1}{32\lambda} + \|\eta\|_2 \leq \frac{1}{16\lambda}.$$

This proves the conclusion about r.

Case 1: $\Phi(r) \leq 8n$.

The definition of Φ together with the fact $\|r - r'\|_2 \leq \frac{1}{16\lambda}$ implies that $\Phi(r') \leq \frac{3}{2}\Phi(r) \leq 12n$.

Case 2: $\Phi(r) \geq 8n$.

Mean value theorem shows there is $\widetilde{r}$ between r and r' such that

$$\begin{aligned}\Phi(r') &= \Phi(r) + \langle \nabla\Phi(\widetilde{r}), r' - r\rangle \\ &= \Phi(r) + \left\langle \nabla\Phi(\widetilde{r}), -\frac{1}{32\lambda}\frac{\overline{g}}{\|\overline{g}\|_2} + \eta \right\rangle\end{aligned}$$

where we used (2.5) at the end. Using Lemma 14, we have $\|r - r'\|_\infty \leq \frac{1}{6\lambda}$ and by assumption, $\|\overline{r} - r\|_\infty \leq \frac{1}{48\lambda}$. Hence, we have $\|\overline{r} - \widetilde{r}\|_\infty \leq \frac{1}{5\lambda}$. Since $\Phi(r) \geq 8n$, we have $\Phi(\overline{r}) \geq 4n$ and hence Lemma 13 shows that

$$\|\nabla\Phi(\widetilde{r}) - \nabla\Phi(\overline{r})\|_2 \leq \frac{1}{3}\|\nabla\Phi(\overline{r})\|_2.$$

Using $\overline{g} = \nabla\Phi(\overline{r})$ and letting $\eta_2 = \nabla\Phi(\widetilde{r}) - \nabla\Phi(\overline{r})$, we have

$$\begin{aligned}\Phi(r') - \Phi(r) &= \left\langle \overline{g} + \eta_2, -\frac{1}{32\lambda}\frac{\overline{g}}{\|\overline{g}\|_2} + \eta \right\rangle \\ &= -\frac{1}{32\lambda}\|\overline{g}\|_2 - \frac{1}{32\lambda}\eta_2^\top\frac{\overline{g}}{\|\overline{g}\|_2} + \overline{g}^\top\eta + \eta_2^\top\eta.\end{aligned}$$

Using $\|\eta_2\|_2 \leq \frac{1}{3}\|\overline{g}\|_2$ and $\|\eta\|_2 \leq \frac{1}{80\lambda}$ (2.6), we have

$$\begin{aligned}\Phi(r') - \Phi(r) &\leq -\frac{1}{32\lambda}\|\overline{g}\|_2 + \frac{1}{32\lambda}\cdot\frac{1}{3}\|\overline{g}\|_2 + \|\overline{g}\|_2\cdot\frac{1}{80\lambda} + \frac{1}{3}\|\overline{g}\|_2\cdot\frac{1}{80\lambda} \\ &\leq -\frac{1}{270\lambda}\|\overline{g}\|_2\end{aligned}$$

Using Lemma 13, we have $\|\overline{g}\|_2 \geq \frac{\lambda}{\sqrt{n}}(\Phi(\overline{r}) - n) \geq 3\lambda\sqrt{n}$. Hence, we have

$$\Phi(r') \leq \Phi(r) - \frac{\sqrt{n}}{40} < 16n. \tag{2.7}$$

The potential actually decreases in this case. So, in both cases, we have that $\Phi(r') \leq 16n$. Suppose that the input satisfies $(x, s) \in \mathcal{P}^\circ \times \mathcal{D}^\circ$ and $\Phi((xs - t_{\text{start}})/t_{\text{start}}) \leq 16n$. Then the algorithm `RobustStep` (Algorithm 3) outputs x and s such that $\Phi((xs - t_{\text{end}})/t_{\text{end}}) \leq 12n$. Furthermore, `RobustStep` takes $O(\sqrt{n}\log n \log(t_{\text{start}}/t_{\text{end}}))$ Newton steps (defined in (1.1)) and that each step satisfies $\|r' - r\|_2 \leq \frac{1}{16\lambda}$ where $r' = ((x+\delta_x)(s+\delta_s) - t')/t'$.

The guarantee of the output follows from Lemma 15. $\square$

Note that `RobustStep` decrease t by a $\Omega(1/(\sqrt{n}\log n))$ factor. So, if we implement the Newton step in the same way as `L2Step`, this new algorithm is slower.

2.2 Selecting $\overline{x}, \overline{s}$ and $\overline{r}$

Each step of the robust interior point method solves the linear system

$$\begin{pmatrix} \overline{\mathbf{S}} & \overline{\mathbf{X}} & \mathbf{0} \\ \mathbf{A} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{A}^\top \end{pmatrix} \begin{pmatrix} \delta_x \\ \delta_s \\ \delta_y \end{pmatrix} = \begin{pmatrix} \nabla\Phi(\overline{r}) \\ 0 \\ 0 \end{pmatrix}$$

for some vectors $\overline{x}, \overline{s}, \overline{r}$ such that $\|\ln\overline{x} - \ln x\|_\infty \leq \frac{1}{48}$, $\|\ln\overline{s} - \ln s\|_\infty \leq \frac{1}{48}$, $\|\overline{r} - r\|_\infty \leq \frac{1}{48\lambda}$. The key observation is that only few coordinate of x, s and r_i changes significant each step and hence we can maintain the solution of the linear system instead of computing from scratch. In this section, we discuss how to select $\overline{x}, \overline{s}, \overline{r}$ such that there are only as few updates to $\overline{x}, \overline{s}, \overline{r}$ as possible while maintaining the invariants.

First, we observe that $\ln x, \ln s$ and r are changed by $O(1)$ in ℓ_2 norm in each step.

Lemma 16. *Define $x^{(k)}, s^{(k)}, r^{(k)}$ according to Lemma 15. Then, $\|\ln x^{(k+1)} - \ln x^{(k)}\|_2$, $\|\ln s^{(k+1)} - \ln s^{(k)}\|_2$ and $\|r^{(k+1)} - r^{(k)}\|_2$ are all bounded by $1/(5\lambda)$.*

Proof. Lemma 6 shows that

$$\|(x^{(k+1)} - x^{(k)})/x^{(k)}\|_{\mu^{(k)}} \leq \|\overline{\delta}_\mu/\mu^{(k)}\|_{\mu^{(k)}} \tag{2.8}$$

where $\mu^{(k)} = \overline{x}^{(k)}\overline{s}^{(k)}$ and $\overline{x}^{(k)}, \overline{s}^{(k)}$ are the $\overline{x}, \overline{s}$ used on the k-th step.

To bound $\mu^{(k)}$, Lemma 15 shows that the invariant $\Phi(r^{(k)})$ holds and hence Lemma 13 shows that $\|(x^{(k)}s^{(k)} - t^{(k)})/t^{(k)}\|_\infty = \|r^{(k)}\|_\infty \leq \frac{\log 32n}{\lambda} \leq \frac{1}{16}$ (recall $\lambda = 16\log 40n$). Together with the fact that $\|\frac{\overline{x}^{(k)} - x^{(k)}}{x^{(k)}}\|_\infty \leq \frac{1}{48}$, $\|\frac{\overline{s}^{(k)} - s^{(k)}}{s^{(k)}}\|_\infty \leq \frac{1}{48}$, we have

$$\|\frac{\overline{x}^{(k)}\overline{s}^{(k)} - t^{(k)}}{t^{(k)}}\|_\infty \leq \frac{1}{8}.$$

Using it on (2.8) gives $\|(x^{(k+1)} - x^{(k)})/x^{(k)}\|_2 \leq \frac{8}{7}\frac{1}{t^{(k)}}\|\overline{\delta}_\mu\|_2$. Using $\overline{\delta}_\mu = -\frac{t'}{32\lambda}\frac{\overline{g}}{\|\overline{g}\|_2}$, we have

$$\|(x^{(k+1)} - x^{(k)})/x^{(k)}\|_2 \leq \frac{1}{28\lambda}.$$

To translate the bound to log scale, we note that $|\ln(1+t) - t| \leq 2t$ for all $|t| \leq \frac{1}{2}$ and hence

$$\|\ln x^{(k+1)} - \ln x^{(k)}\|_2 = \|\ln(1 + \frac{x^{(k+1)} - x^{(k)}}{x^{(k)}})\|_2 \leq \frac{1}{14\lambda}.$$

The bound for $\|\ln s^{(k+1)} - \ln s^{(k)}\|_2$ is similar.

The bound for $\|r^{(k+1)} - r^{(k)}\|_\infty$ follows from Lemma 15. $\square$

Now, the question is how to select $\ln\overline{x}, \ln\overline{s}$ and $\overline{r}$ such that they are close to $\ln x, \ln s$ and r in ℓ_∞ norm. If the cost of updating the inverse of a matrix is linear to the rank of update, then we can simply update any coordinate of $\overline{x}, \overline{s}$ and $\overline{r}$ whenever they violate the condition. However, due to fast matrix multiplication, the average cost (per rank) of update is lower when the rank of update is large. Therefore, it is beneficial to update coordinate preemptively.

Now, we state the algorithm for selecting $\overline{x}, \overline{s}$ and $\overline{r}$. This is a general algorithm for selecting an approximate vector.

Algorithm 4: `SelectVector`$(\overline{v}, v^{(0)}, v^{(1)}, \cdots, v^{(k)}, \delta)$

Let $I = \{\}$ be the set of updating coordinates.
for $\ell = 0, 1, \cdots, \lceil\log n\rceil$ **do**
  if $k = 0 \mod 2^\ell$ **then**
    if $\ell = \lceil\log n\rceil$ **then**
      $I = [n]$.
    else
      $I = I \cup \{i : |v_i^{(k)} - v_i^{(k-2^\ell)}| \geq \delta/(2\lceil\log n\rceil)\}$.
    end
  end
end
$\overline{v}_i \leftarrow v_i^{(k)}$ for all $i \in I$
Return $\overline{v}$

Lemma 17. *Given vectors $v^{(0)}, v^{(1)}, v^{(2)}, \cdots$ in stream. Suppose that $\|v^{(k+1)} - v^{(k)}\|_2 \le \beta$ for all k. For any $\frac{1}{2} > \delta > 0$, we define the approximation vector $\overline{v}^{(0)} = v^{(0)}$ and $\overline{v}^{(k)} = \texttt{SelectVector}(\overline{v}^{(k-1)}, v^{(0)}, v^{(1)}, \cdots, v^{(k)}, \delta)$. Then, we have that*

- $\|\overline{v}^{(k)} - v^{(k)}\|_\infty \le \delta$ *for all k.*
- $\|\overline{v}^{(k)} - \overline{v}^{(k-1)}\|_0 \le O(2^{2\ell_k}(\beta/\delta)^2 \log^2 n)$ *where ℓ_k is the largest integer ℓ with $k = 0 \mod 2^\ell$.*

Proof. For the accuracy result, we first fix some coordinate $i \in [n]$. Note that $\overline{v}_i^{(k)} = v_i^{(k')}$ for some $k - 2^{\lceil \log n \rceil} < k' < k$ because we set $\overline{v} \leftarrow v$ every $2^{\lceil \log n \rceil}$ steps. We can write $k = k_0 > k_1 > k_2 > \cdots > k_s = k'$ such that $k_i - k_{i+1}$ is a power of 2 and that $|s| \le 2\lceil \log n \rceil$. Hence, we have that

$$v_i^{(k)} - \overline{v}_i^{(k)} = v_i^{(k_0)} - v_i^{(k_s)} = \sum_{j=1}^{s} (v_i^{(k_{j-1})} - v_i^{(k_j)}).$$

The fact that $\overline{v}_i$ is not updated after step k_s implies that $|v_i^{(k_{j-1})} - v_i^{(k_j)}| \le \delta/(2\lceil \log n \rceil)$ and hence $|v_i^{(k)} - \overline{v}_i^{(k)}| \le \delta$. Since this holds for every i, we have that $\|\overline{v}^{(k)} - v^{(k)}\|_\infty \le \delta$.

For the sparsity result, we first bound the size of the set $I_\ell \stackrel{\text{def}}{=} \{i : |v_i^{(k)} - v_i^{(k-2^\ell)}| \ge \delta/(2\lceil \log n \rceil)\}$. Note that

$$\Omega(|I_\ell| \cdot \frac{\delta^2}{\log^2 n}) \le \sum_{i=1}^{n} |v_i^{(k)} - v_i^{(k-2^\ell)}|^2 = 2^\ell \sum_{i=1}^{n} \sum_{t=k-2^\ell}^{k-1} |v_i^{(t+1)} - v_i^{(t)}|^2 \le 2^{2\ell}\beta^2$$

where we used $\|v^{(t+1)} - v^{(t)}\|_2 \le \beta$ at the end. Hence, we have $|I_\ell| = O(2^{2\ell}(\beta/\delta)^2 \log^2 n)$. Hence, $|I| \le \sum_{\ell=0}^{\ell_k} |I_\ell| \le O(2^{2\ell_k}(\beta/\delta)^2 \log^2 n)$. $\square$

2.3 Inverse Maintenance

In this section, we discuss how to maintain the solution of the Newton step. Although both the matrix and the vector changes during the algorithm, we can simplify it by moving the vector inside the matrix.

Fact 18. *For any invertible matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$ and any vector $v \in \mathbb{R}^n$, we have*

$$\begin{bmatrix} \mathbf{M} & v \\ 0 & -1 \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{M}^{-1} & \mathbf{M}^{-1}v \\ 0 & -1 \end{bmatrix}.$$

Hence, the question of maintain the solution reduces to the problem of maintaining a column of the inverse of the matrix

$$\mathbf{M}_{\overline{x},\overline{s},\overline{r}} \stackrel{\text{def}}{=} \begin{pmatrix} \overline{\mathbf{S}} & \overline{\mathbf{X}} & \mathbf{0} & \nabla\Phi(\overline{r}) \\ \mathbf{A} & \mathbf{0} & \mathbf{0} & 0 \\ \mathbf{0} & \mathbf{I} & \mathbf{A}^\top & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}. \tag{2.9}$$

Now, when we update $\overline{x}, \overline{s}, \overline{r}$ to $\overline{x} + \delta_{\overline{x}}, \overline{s} + \delta_{\overline{s}}, \overline{r} + \delta_{\overline{r}}$ with q coordinates changed in total, there are only q coordinates in $\mathbf{M}$ is changed. Hence, we can compute the update the inverse of $\mathbf{M}$ using the Woodbury matrix identity.

Lemma 19. *Given vectors $\overline{x}, \overline{s}, \overline{r} \in \mathbb{R}^n$ and the update $\delta_{\overline{x}}, \delta_{\overline{s}}, \delta_{\overline{r}} \in \mathbb{R}^n$. Let $q = \|\delta_{\overline{x}}\|_0 + \|\delta_{\overline{s}}\|_0 + \|\delta_{\overline{r}}\|_0$ and $T_{m,n,\ell}$ be the cost of multiplying $m \times n$ and $n \times \ell$ matrices. Then, we have*

- *Given $\mathbf{M}_{\overline{x},\overline{s},\overline{r}}^{-1}$, we can compute $\mathbf{M}_{\overline{x}+\delta_{\overline{x}},\overline{s}+\delta_{\overline{s}},\overline{r}+\delta_{\overline{r}}}^{-1}$ in time $O(T_{n,n,q})$.*
- *Given $\mathbf{M}_{\overline{x},\overline{s},\overline{r}}^{-1}$ and $\mathbf{M}_{\overline{x},\overline{s},\overline{r}}^{-1}b$, we can compute $\mathbf{M}_{\overline{x}+\delta_{\overline{x}},\overline{s}+\delta_{\overline{s}},\overline{r}+\delta_{\overline{r}}}^{-1}b$ in time $O(T_{q,q,q} + nq)$.*

Proof. We write $\mathbf{M}_0 = \mathbf{M}_{\overline{x},\overline{s},\overline{r}}$ and $\mathbf{M}_1 = \mathbf{M}_{\overline{x}+\delta_{\overline{x}},\overline{s}+\delta_{\overline{s}},\overline{r}+\delta_{\overline{r}}}$. Note that $\mathbf{M}_1$ and $\mathbf{M}_0$ is off by just q entries. Hence, we can write

$$\mathbf{M}_1 = \mathbf{M}_0 + \mathbf{UCV}$$

where $\mathbf{U}$ is q columns of identity matrix, $\mathbf{C}$ is a $q \times q$ matrix and $\mathbf{V}$ is q rows of identity matrix. Hence, Woodbury matrix identity shows that

$$\mathbf{M}_1^{-1} = (\mathbf{M}_0 + \mathbf{UCV})^{-1} = \mathbf{M}_0^{-1} - \mathbf{M}_0^{-1}\mathbf{U}(\mathbf{C}^{-1} + \mathbf{V}\mathbf{M}_0^{-1}\mathbf{U})^{-1}\mathbf{V}\mathbf{M}_0^{-1}.$$

Note that $\mathbf{M}_0^{-1}\mathbf{U}$, $\mathbf{V}\mathbf{M}_0^{-1}\mathbf{U}$, $\mathbf{V}\mathbf{M}_0^{-1}$ are just blocks of $\mathbf{M}_0^{-1}$ and no computation is needed. Hence, we can compute $(\mathbf{C}^{-1}+\mathbf{V}\mathbf{M}_0^{-1}\mathbf{U})^{-1}$ in the time of inverting two $q\times q$ matrices, which is $O(T_{q,q,q})$. The rest of the formula can be computed in $O(T_{n,n,q}+T_{n,q,q})=O(T_{n,n,q})$ time. In total, the runtime is $O(T_{n,n,q})$.

For computing $\mathbf{M}_1^{-1}b$, we note that

$$\mathbf{M}_1^{-1}b=\mathbf{M}_0^{-1}b-\mathbf{M}_0^{-1}\mathbf{U}(\mathbf{C}^{-1}+\mathbf{V}\mathbf{M}_0^{-1}\mathbf{U})^{-1}\mathbf{V}\mathbf{M}_0^{-1}b.$$

Since $\mathbf{M}_0^{-1}b$ is given, the above formula can be computed in $O(T_{q,q,q}+nq)$ time where the $O(nq)$ term comes from multiplying $\mathbf{M}_0^{-1}\mathbf{U}$ with a vector. $\square$

In this note, we only use the following estimate for $T_{n,n,r}$.

Lemma 20. *For $r\leq n$, we have $T_{n,n,r}=n^2+n^{\omega-\frac{\omega-2}{1-\alpha}}r^{\frac{\omega-2}{1-\alpha}}$ where $\omega\sim 2.37$ is the exponent of matrix multiplication and $\alpha\sim 0.31$ is the dual exponent of matrix multiplication.*

Algorithm 5: `FastRobustStep`$(\mathbf{A},x,s,t_{\text{start}},t_{\text{end}})$

Assume the restart frequency satisfies $2^{2\ell_*}\leq n^{\alpha}$**.**
Define $r=(xs-t)/t$ and Φ according to (2.2) with $\lambda=16\log 40n$.
Invariant: $(x,s)\in\mathcal{P}^{\circ}\times\mathcal{D}^{\circ}$ and $\Phi(r)\leq 16n$.
Let $t=t_{\text{start}}$, $h=1/(128\lambda\sqrt{n})$ and n is the number of columns in $\mathbf{A}$.
Let $x^{(0)}=\overline{x}^{(0)}=x$, $s^{(0)}=\overline{s}^{(0)}=s$, $r^{(0)}=\overline{r}^{(0)}=(xs-t)/t$.
Let $\mathbf{T}=\mathbf{M}^{-1}_{\overline{x}^{(0)},\overline{s}^{(0)},\overline{r}^{(0)}}$ (defined in (2.9)) and $u=\mathbf{T}e_{2n+d+1}$.
repeat
- Let $t'=\max(t/(1+h),t_{\text{end}})$, $\overline{\delta}_{\mu}=-\frac{t'}{32\lambda}\frac{\overline{g}}{\|\overline{g}\|_2}$, $\overline{g}=\nabla\Phi(\overline{r})$.
- **if** $k=0 \mod 2^{\ell_*}$ **then**
  - Update $\mathbf{T}$ to $\mathbf{M}^{-1}_{\overline{x}^{(k)},\overline{s}^{(k)},\overline{r}^{(k)}}$ using Lemma 19.
  - $u\leftarrow\mathbf{T}e_{2n+d+1}$, $v\leftarrow u$.
- **else**
  - Update v to $\mathbf{M}^{-1}_{\overline{x}^{(k)},\overline{s}^{(k)},\overline{r}^{(k)}}e_{2n+d+1}$ using vector u and Lemma 19.
- **end**
- Let (δ_x,δ_s) be the first $2n$ coordinates of v.
- Let $x^{(k+1)}=x^{(k)}+\delta_x$, $s^{(k+1)}=s^{(k)}+\delta_s$ and $t\leftarrow t'$.
- Let $r^{(k+1)}=(x^{(k+1)}s^{(k+1)}-t)/t$.
- $\ln\overline{x}^{(k+1)}=$ `SelectVector`$(\ln\overline{x}^{(k)},\ln x^{(0)},\ln x^{(1)},\cdots,\ln x^{(k+1)},1/48)$.
- $\ln\overline{s}^{(k+1)}=$ `SelectVector`$(\ln\overline{s}^{(k)},\ln s^{(0)},\ln s^{(1)},\cdots,\ln s^{(k+1)},1/48)$.
- $\overline{r}^{(k+1)}=$ `SelectVector`$(\overline{r}^{(k)},r^{(0)},r^{(1)},\cdots,r^{(k+1)},1/(48\lambda))$.
- Set $k\leftarrow k+1$.

until $t\neq t_{\text{end}}$;
Return (x,s)

Combining the previous Lemma with Lemma 17, we have the following:

Lemma 21. *With $2^{2\ell_*}=\min(n^{\alpha},n^{2/3})$,* `FastRobustStep` *(algorithm 5) takes*

$$\widetilde{O}((n^{\omega}+n^{2+1/6}+n^{5/2-\alpha/2})\log(t_{\text{end}}/t_{\text{start}}))$$

time.

Proof. The bottleneck of `FastRobustStep` is the step for updating v and updating $\mathbf{T}$. It depends on the number of coordinates updates in $\overline{x},\overline{s},\overline{r}$. Lemma 16 shows that $\ln x,\ln s$ and r changes by $\alpha\stackrel{\text{def}}{=}1/(5\lambda)$ in ℓ_2 norm per step. Since we set the error of `SelectVector` to be $\delta\stackrel{\text{def}}{=}1/(48\lambda)$, Lemma 17 shows that $q_k\stackrel{\text{def}}{=}O(2^{2\ell_k}\log^2 n)$ coordinates in $\overline{x},\overline{s},\overline{r}$ are updated at the k-th step where ℓ_k is the largest integer ℓ with $k=0 \mod 2^{\ell}$.

Cost of updating v: We update u whenever $k=0 \mod 2^{\ell_*}$. Within that 2^{ℓ_*} steps, the number of coordinates updated in $\overline{x},\overline{s},\overline{r}$ is bounded by

$$q\stackrel{\text{def}}{=}\sum_{k=1}^{2^{\ell_*}-1}q_k=\sum_{k=1}^{2^{\ell_*}-1}O(2^{2\ell_k}\log^2 n)=O(2^{2\ell_*}\log^2 n).$$

Therefore, $\mathbf{M}_{\overline{x}^{(k)},\overline{s}^{(k)},\overline{r}^{(k)}}$ and $\mathbf{T}$ are off by q coordinates. Lemma 19 shows that it takes

$$O(T_{q,q,q}+nq)=\widetilde{O}(2^{2\ell_{*}\omega}+n2^{2\ell_{*}})$$

time to compute $v=\mathbf{M}_{\overline{x}^{(k)},\overline{s}^{(k)},\overline{r}^{(k)}}^{-1}e_{2n+d+1}$ using $u=\mathbf{T}e_{2n+d+1}$.

Cost of updating $\mathbf{T}$: For the k-th step that updates $\mathbf{T}$, the number of coordinates updated in $\overline{x},\overline{s},\overline{r}$ is bounded by $q+O(2^{2\ell_{k}}\log^{2}n)=O(2^{2\ell_{k}}\log^{2}n)$ where the first term is the delayed update and the second term is the update at that step. Lemma 19 shows that it takes $\widetilde{O}(T_{n,n,2^{2\ell_{k}}})$ time to update $\mathbf{T}$. Since $2^{2\ell_{k}}$ update happens every $2^{\ell_{k}}$ iterations, the amortized cost is

$$\begin{aligned}\widetilde{O}(\sum_{\ell=\ell_{*}}^{\frac{1}{2}\log n}2^{-\ell}T_{n,n,2^{2\ell}}) &= \widetilde{O}(\sum_{\ell=\ell_{*}}^{\frac{1}{2}\log n}(n^{\omega-\frac{\omega-2}{1-\alpha}}2^{2\ell\cdot\frac{\omega-2}{1-\alpha}-\ell}+n^{2}2^{-\ell}))\\ &= \widetilde{O}(\sum_{\ell=\ell_{*}}^{\frac{1}{2}\log n}n^{\omega-\frac{\omega-2}{1-\alpha}}2^{2\ell\cdot\frac{\omega-2}{1-\alpha}-\ell}+n^{2}2^{-\ell_{*}})\end{aligned}$$

where we used Lemma 20. The sum above is dominated by either the term at $\ell=0$ or the term at $\ell=\frac{1}{2}\log n$. Hence, the amortized cost of updating $\mathbf{T}$ is

$$\widetilde{O}(n^{\omega-\frac{\omega-2}{1-\alpha}}2^{2\ell_{*}\cdot\frac{\omega-2}{1-\alpha}-\ell_{*}}+n^{\omega-\frac{1}{2}}+n^{2}2^{-\ell_{*}})=\widetilde{O}(n^{\omega-\frac{1}{2}}+n^{2}2^{-\ell_{*}})$$

where we used $n^{\omega-\frac{\omega-2}{1-\alpha}}2^{2\ell_{*}\cdot\frac{\omega-2}{1-\alpha}}\leq n^{2}$ since $2^{2\ell_{*}}\leq n^{\alpha}$.

Cost of initialize $\mathbf{T}$ and u: $O(n^{\omega})$.

Since there are $\sqrt{n}\log(t_{\text{end}}/t_{\text{start}})$ steps, the total cost is

$$\begin{aligned}&\widetilde{O}(n^{\omega}+\sqrt{n}\log(t_{\text{end}}/t_{\text{start}})(2^{2\ell_{*}\omega}+n2^{2\ell_{*}}+n^{\omega-\frac{1}{2}}+n^{2}2^{-\ell_{*}}))\\ =&\widetilde{O}(\sqrt{n}\log(t_{\text{end}}/t_{\text{start}})(n2^{2\ell_{*}}+n^{\omega-\frac{1}{2}}+n^{2}2^{-\ell_{*}}))\end{aligned}$$

where we used $2^{2\ell_{*}\omega}\leq n^{\alpha\omega}\leq n$. Putting $2^{2\ell_{*}}=\min(n^{\alpha},n^{2/3})$, we have

$$\widetilde{O}((n^{\omega}+n^{2+1/6}+n^{5/2-\alpha/2})\log(t_{\text{end}}/t_{\text{start}})).$$

□

Following Section 1.4, we have the following Theorem

Theorem 22. *Consider a linear program* $\min_{\mathbf{A}x=b,x\geq0}c^{\top}x$ *with* n *variables and* d *constraints. Assume the linear program has inner radius* r*, outer radius* R *and Lipschitz constant* L *(See Definition 9), we can find* x *such that*

$$\begin{aligned}c^{\top}x &\leq \min_{\mathbf{A}x=b,x\geq0}c^{\top}x+\epsilon LR,\\ \mathbf{A}x &= b,\\ x &\geq 0.\end{aligned}$$

in time

$$\widetilde{O}((n^{\omega}+n^{2+1/6}+n^{5/2-\alpha/2})\log(nR/(\epsilon r))).$$

If we further assume that the solution $x^{*}=\arg\min_{\mathbf{A}x=b,x\geq0}c^{\top}x$ *is unique and that* $c^{\top}x\geq c^{\top}x^{*}+\eta LR$ *for any other vertex* x *of* $\{\mathbf{A}x=b,x\geq0\}$*, then we have that* $\|x-x^{*}\|_{2}\leq\frac{2\epsilon R}{\eta}$.

Proof. The algorithm for find x is same as `L2LPApproximate` except the function `L2Step` is replaced by the function `FastRobustStep`. The runtime of `FastRobustStep` is analyzed in Lemma 21. Since `FastRobustStep` is a instantiation of `RobustStep`, the property of the output is analyzed in Lemma 15. □

References

[1] Adam R Klivans and Daniel Spielman. Randomness efficient identity testing of multivariate polynomials. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 216–223, 2001.

A Finding a Point on Central Path

Continue from the discussion in Section 1.4.

First, we show that $x^{(0)}$ defined in Theorem 11 is indeed on the central path of the modified linear program.

Lemma 23. *The modified linear program **??** has an explicit central path point $x^{(0)} = (x_c^+, x_c^-, \overline{R})$ at t.*

Proof. Recall that we say (x^+, x^-, θ) is on the central path at t if x^+, x^-, θ are positive and it satisfies the following equation

$$\begin{aligned} \mathbf{A}x^+ - \mathbf{A}x^- &= b, \\ \sum_{i=1}^n x_i^+ + \theta &= \Lambda, \\ \mathbf{A}^\top y + \lambda + s^+ &= c, \\ \mathbf{A}^\top y + s^- &= d, \\ \lambda + s^\theta &= 0, \end{aligned} \tag{A.1}$$

for some $s^+, s^- \in \mathbb{R}^n_{>0}$, $s^\theta > 0$, $y \in \mathbb{R}^d$ and $\lambda \in \mathbb{R}$.

Now, we verify the solution $x^+ = x_c^+ = \frac{t}{c+t/\overline{R}}$, $x^- = x_c^- = \frac{t}{c+t/\overline{R}} - x_\circ$, $\theta = \overline{R}$, $x_\circ = \mathbf{A}^\top(\mathbf{A}\mathbf{A}^\top)^{-1}b$, $y = 0$, $s^+ = \frac{t}{x^+}$, $s^- = \frac{t}{x^-}$, $s^\theta = \frac{t}{\overline{R}}$, $\lambda = -s_3$. Using $\mathbf{A}x_\circ = b$, one can check it satisfies all the equality constraints above.

For the inequality constraints, using $\|c\|_\infty \le L$ and $t \ge 8L\overline{R}$, we have

$$\frac{3}{4}\overline{R} \le \frac{t}{L + t/\overline{R}} \le x_{c,i}^+ \le \frac{t}{-L + t/\overline{R}} \le \frac{3}{2}\overline{R} \tag{A.2}$$

and hence $x_c^+ > 0$ and so is s^+. Since $\|x_\circ\|_2 \le R \le \frac{\overline{R}}{2}$ and $x_{c,i}^+ \ge \frac{3}{4}\overline{R}$ for all i, we have $x_{c,i}^- \ge 0$ for all i. Hence, x_c^- and s^- are positive. Finally, θ and s^θ are positive. This proves that $(x_c^+, x_c^-, \overline{R})$ is the central path point at t. $\square$

Next, we show that the near central path point (x, s) at $t = LR$ is far from the constraints $x^+ \ge 0$ and is close to the constraints $x^- \ge 0$. The proof for both involves the same idea: use the optimality condition of x. Throughout the rest of the section, we are given $(x, s) \in \mathcal{P}_{t,\overline{R}} \times \mathcal{D}_{t,\overline{R}}$ such that $\mu = xs$ satisfies

$$\frac{5}{6}LR \le \mu \le \frac{7}{6}LR.$$

We write μ into three parts $(\mu^+, \mu^-, \mu^\theta)$. By Lemma 4, we have that $x \stackrel{\text{def}}{=} (x^+, x^-, \theta)$ minimizes the function

$$f(x^+, x^-, \theta) \stackrel{\text{def}}{=} c^\top x^+ + d^\top x^- - \sum_{i=1}^n \mu_i^+ \log x_i^+ - \sum_{i=1}^n \mu_i^- \log x_i^- - \mu^\theta \log \theta$$

over the domain $\mathcal{P}_{t,\overline{R}}$. The gradient of f is a bit complicated and we only need to consider the directional derivative at x on the direction "$v - x$" where v is the point such that $\mathbf{A}v = b$ and $v \ge r$. Since our domain is in $\mathcal{P}_{t,\overline{R}} \subset \mathbb{R}^{2n+1}$, we need to lift v to higher dimension. Now, we define the point

$$\begin{aligned} v^- &= \min(x^-, \frac{8L\overline{R}}{t} \cdot R), \\ v^+ &= v + v^-, \\ v^\theta &= \Lambda - \sum_{i=1}^n v_i^+. \end{aligned}$$

First, we need to get some basic bounds on Λ and d.

Lemma 24. *We have that $\frac{3}{4}n\overline{R} \le \Lambda \le 3n\overline{R}$ and $d_i \ge t/(2\overline{R})$ for all i.*

Proof. By (A.2), we have $\frac{3}{4}\overline{R} \leq x_{c,i}^{+} \leq \frac{3}{2}\overline{R}$. By the definition of Λ, we have

$$\Lambda = \sum_i x_i^{+} + \overline{R} \leq \frac{3}{2}n\overline{R} + \overline{R} \leq 3n\overline{R}.$$

Similarly, we have $\Lambda = \sum_i x_i^{+} + \overline{R} \geq \frac{3}{4}n\overline{R}$.

For the bound of d, recall that $d = t/x_c^{-}$ with $x_c^{-} = x_c^{+} - x_{\circ}$ and $x_{\circ} = \mathbf{A}^{\top}(\mathbf{A}\mathbf{A}^{\top})^{-1}b = \arg\min_{\mathbf{A}x=b}\|x\|_2$. Since we assumed the linear program has outer radius R, we have that $\|x_{\circ}\|_2 \leq R$. Hence, $x_c^{-} \leq \frac{3}{2}\overline{R} + R \leq 2\overline{R}$. Therefore, $d \geq t/(2\overline{R})$. $\square$

The following Lemma shows that $(v^{+}, v^{-}, v^{\theta}) \in \mathcal{P}_{t,\overline{R}}$.

Lemma 25. *We have that $(v^{+}, v^{-}, v^{\theta}) \in \mathcal{P}_{t,\overline{R}}$. Furthermore, we have $v^{\theta} \geq \frac{1}{4}n\overline{R}$.*

Proof. Note that $(v^{+}, v^{-}, v^{\theta})$ satisfies the linear constraints in $\mathcal{P}_{t,\overline{R}}$ by construction. It suffices to prove the vector is positive. Since $x^{-} > 0$, we have $v^{-} > 0$. Since $v \geq r$, we also have $v^{+} > 0$. For v^{θ}, we use $\Lambda \geq \frac{3}{4}n\overline{R}$ (Lemma 24), $v \leq R$ and $v^{-} \leq \frac{8L\overline{R}}{t} \cdot R \leq R$ to get

$$v^{\theta} = \Lambda - \sum_{i=1}^{n} v_i^{+} \geq \frac{3}{4}n\overline{R} - \sum_{i=1}^{n}(v_i + v_i^{-}) \geq \frac{1}{4}n\overline{R}.$$

$\square$

Now, we define the path $p(t) = (1-t)(x^{+}, x^{-}, \theta) + t(v^{+}, v^{-}, v^{\theta})$. Since $p(0)$ minimizes f, we have that $\frac{d}{dt}f(p(t))|_{t=0} \geq 0$. In particular, we have

$$\begin{aligned}
0 &\leq \frac{d}{dt}f(p(t))|_{t=0} \\
&= c^{\top}(v^{+} - x^{+}) + d^{\top}(v^{-} - x^{-}) - \sum_{i=1}^{n}\frac{\mu_i^{+}}{x_i^{+}}(v^{+} - x^{+})_i - \sum_{i=1}^{n}\frac{\mu_i^{-}}{x_i^{-}}(v^{-} - x^{-})_i - \frac{\mu^{\theta}}{\theta}(v_{\theta} - \theta) \\
&= \frac{\mu^{\theta}}{\theta}(\theta - v_{\theta}) + \sum_{i=1}^{n}(c_i - \frac{\mu_i^{+}}{x_i^{+}})(v^{+} - x^{+})_i + \sum_{i=1}^{n}(d_i - \frac{\mu_i^{-}}{x_i^{-}})(v^{-} - x^{-})_i. \qquad \text{(A.3)}
\end{aligned}$$

Now, we bound each term one by one. For the first term, we note that

$$\frac{\mu^{\theta}}{\theta}(\theta - v_{\theta}) \leq \mu^{\theta} \leq 2LR. \qquad \text{(A.4)}$$

For the second term in (A.3), we have the following

Lemma 26. *We have that $\sum_{i=1}^{n}(c_i - \frac{\mu_i^{+}}{x_i^{+}})(v^{+} - x^{+})_i \leq 4nL\overline{R} - \frac{LRr}{2\min_i x_i^{+}}$.*

Proof. Note that

$$\begin{aligned}
\sum_{i=1}^{n}(c_i - \frac{\mu_i^{+}}{x_i^{+}})(v^{+} - x^{+})_i &= \sum_{i=1}^{n}(c_i v_i^{+} - \frac{\mu_i^{+}}{x_i^{+}}v_i^{+} - c_i x_i^{+} + \mu_i^{+}) \\
&\leq \sum_{i=1}^{n} c_i v_i^{+} + \sum_{i=1}^{n}\mu_i^{+} - \sum_{i=1}^{n}\frac{\mu_i^{+}}{x_i^{+}}v_i^{+} \\
&\leq \|c\|_{\infty}\|v^{+}\|_1 + 2nLR - \frac{1}{2}\sum_{i=1}^{n}\frac{LRr}{x_i^{+}}
\end{aligned}$$

where we used $\mu_i^{+} \in [\frac{LR}{2}, 2LR]$ and $v_i^{+} \geq v_i \geq r$ at the end. The result follows from $\|c\|_{\infty} \leq L$, $\|v^{+}\|_1 \leq \Lambda \leq 3n\overline{R}$ (Lemma 24). $\square$

Lemma 27. *We have that $\sum_{i=1}^{n}(d_i - \frac{\mu_i^{-}}{x_i^{-}})(v^{-} - x^{-})_i \leq 2LR - \frac{t}{4\overline{R}}\max_i x_i^{-}$.*

Proof. Using $v^- = \min(x^-, \frac{8L\overline{R}}{t} \cdot R)$, we have $v_i^- \le x_i^-$. We can ignore the terms with $v_i^- = x_i^-$.

For $v_i^- < x_i^-$, we have $x_i^- \ge \frac{8L\overline{R}}{t} R$. Using $d_i \ge \frac{t}{2\overline{R}}$ (Lemma 24), we have

$$d_i - \frac{\mu_i^-}{x_i^-} \ge d_i - \frac{\mu_i^-}{\frac{8L\overline{R}}{t} R} \ge d_i - \frac{2LR}{\frac{8L\overline{R}}{t} R} = d_i - \frac{t}{4\overline{R}} \ge \frac{t}{4\overline{R}}.$$

Hence, we have

$$\sum_{i=1}^{n} (d_i - \frac{\mu_i^-}{x_i^-})(v^- - x^-)_i \le \frac{t}{4\overline{R}} \sum_{i=1}^{n} (v^- - x^-)_i \le \frac{t}{4\overline{R}} (\frac{8L\overline{R}}{t} \cdot R - \max_i x_i^-).$$

$\square$

Combining (A.3), (A.4), Lemma 26 and Lemma 27, we have

$$\begin{aligned} 0 &\le \frac{\mu^\theta}{\theta}(\theta - v_\theta) + \sum_{i=1}^{n} (c_i - \frac{\mu_i^+}{x_i^+})(v^+ - x^+)_i + \sum_{i=1}^{n} (d_i - \frac{\mu_i^-}{x_i^-})(v^- - x^-)_i \\ &\le 2LR + 4nL\overline{R} - \frac{LRr}{2\min_i x_i^+} + 2LR - \frac{t}{4\overline{R}} \max_i x_i^- \\ &= 5nL\overline{R} - \frac{LRr}{2\min_i x_i^+} - \frac{t}{4\overline{R}} \max_i x_i^-. \end{aligned}$$

Hence, we have that

$$\frac{LRr}{2\min_i x_i^+} + \frac{t}{4\overline{R}} \max_i x_i^- \le 5nL\overline{R}.$$

In particular, this shows the following:

Lemma 28. *We have that* $\min_i x_i^+ \ge \frac{Rr}{10n\overline{R}}$ *and* $\max_i x_i^- \le \frac{20nL\overline{R}}{t} \cdot \overline{R}$.

Now, we are ready to prove the second conclusion of Theorem 11.

Lemma 29. *For any primal* $x \stackrel{\text{def}}{=} (x^+, x^-, \theta) \in \mathcal{P}_{t,\overline{R}}$ *and dual* $s \stackrel{\text{def}}{=} (s^+, s^-, s^\theta) \in \mathcal{D}_{t,\overline{R}}$ *such that* $\frac{5}{6}LR \le x_i s_i \le \frac{7}{6}LR$*, we have that*

$$(x^+ - x^-, s^+ - s^\theta) \in \mathcal{P} \times \mathcal{D}$$

and that $x_i^- \le \epsilon x_i^+$ *and* $s^\theta \le \epsilon s_i^+$ *for all* i.

Proof. First we check $x \stackrel{\text{def}}{=} x^+ - x^- \in \mathcal{P}$. By the choice of t and $\overline{R}$, Lemma 28 shows that

$$\frac{\max_i x_i^-}{\min_i x_i^+} \le \frac{\frac{20nL\overline{R}}{t} \cdot \overline{R}}{\frac{Rr}{10n\overline{R}}} = \frac{200n^2 L \frac{\overline{R}^3}{Rr}}{t} \le \epsilon.$$

Hence, we have $x^+ - x^- > 0$ and that $\mathbf{A}(x^+ - x^-) = b$.

Next, we check $s \stackrel{\text{def}}{=} s^+ - s^\theta \in \mathcal{D}$. Since $x \in \mathcal{P}$, we have $x \le R$ and $x_i^+ \le \frac{3}{2} x_i \le \frac{3}{2} R$. Since $x_i^+ s_i^+ \ge \frac{5}{6} LR$, we have $s_i^+ \ge \frac{1}{2} L$. On the other hand, we have $\theta = \Lambda - \sum_{i=1}^{n} x_i^+ \ge \Lambda - 2nR \ge \frac{1}{2} n\overline{R}$ (Lemma 24). Hence, $s^\theta \le \frac{\frac{7}{6}LR}{\frac{1}{2}n\overline{R}} \le \frac{5LR}{2n\overline{R}}$. Combining both and the choice of $\overline{R}$, we have

$$\frac{s^\theta}{\min_i s_i^+} \le \frac{\frac{5LR}{2n\overline{R}}}{L/2} = \frac{5R}{n\overline{R}} \le \epsilon$$

Hence, we have $s = s^+ - s^\theta > 0$ and that $\mathbf{A}^\top y + s = \mathbf{A}^\top y + s^+ - s^\theta = \mathbf{A}^\top y + s^+ + \lambda = c$ (See (A.1)). $\square$

To ensure the reduction does not increase the complexity of solving linear system, we note that the modified linear program, the linear system is

$$\overline{\mathbf{A}} = \left[\begin{array}{ccc} \mathbf{A} & -\mathbf{A} & 0 \\ 1 & 0 & 1 \end{array} \right]$$

For any diagonal matrices $\mathbf{W}_1, \mathbf{W}_2$ and any scalar α, we have

$$\mathbf{H} \stackrel{\text{def}}{=} \overline{\mathbf{A}} \left[\begin{array}{ccc} \mathbf{W}_1 & \mathbf{0} & 0 \\ \mathbf{0} & \mathbf{W}_2 & 0 \\ 0 & 0 & \alpha \end{array} \right] \overline{\mathbf{A}}^\top = \left[\begin{array}{cc} \mathbf{A}^\top(\mathbf{W}_1 + \mathbf{W}_2)\mathbf{A} & \mathbf{A}\mathbf{W}_1 1_n \\ (\mathbf{A}\mathbf{W}_1 1_n)^\top & 1_n^\top \mathbf{W}_1 1_n + \alpha \end{array} \right].$$

Note that the second row and column block has size 1. By block inverse formula, $\mathbf{H}^{-1}v$ is an explicit formula involving $(\mathbf{A}^\top(\mathbf{W}_1 + \mathbf{W}_2)\mathbf{A})^{-1}v_{1:n}$ and $(\mathbf{A}^\top(\mathbf{W}_1 + \mathbf{W}_2)\mathbf{A})^{-1}\mathbf{A}\mathbf{W}_1 1_n$. Hence, we can compute $\mathbf{H}^{-1}v$ by solving two linear systems of the form $\mathbf{A}^\top\mathbf{W}\mathbf{A}$ and some extra linear work.