

# LINEAR PROGRAMMING IN POLYNOMIAL TIME

BRIAN AXELROD

ABSTRACT. Polynomial time methods for LPs and SDPs

## CONTENTS

1. Ellipsoid	1
2. Interior Point Methods	2
2.1. Barrier Methods	2
2.2. KKT conditions	2
2.3. Newton's method	2
2.4. A Barrier Method for Linear programming	3
2.5. Barrier Method for LPs using Newton Iterations	3
2.6. Newton Iteration	4
References	5

## 1. ELLIPSOID

The ellipsoid method was the first polynomial time method to solve linear programs. It is of theoretical interest because it is polynomial time, can handle an infinite number of constraints, and can be extended to handle more than just LPs. A quick outline of the method is below.

- (1) Start with  $E_0 = E(0, R^2)$
- (2) Iterate from 0 to  $2m(m+1) \log R/r + 1$ 
  - (a) Let  $E_k$  be the minimum volume ellipsoid containing  $E_{k-1} \cap H$  where  $H$  is a separating hyperplane given by the separating oracle
  - (b) If the center  $p$ , of  $E_k$  is feasible, return  $p$
- (3) Return that the LP is infeasible

1.0.1. *Minimum Volume Ellipsoid containing  $E_{k-1} \cap H$ .* Assume that the separating hyperplane is of the form  $a_i^T y \leq a_i^T y_{k-1}$  for  $y_{k-1}$  the center of  $E_{k-1}$ .

**Theorem 1.1.** *The minimum volume ellipsoid containing  $E(y, B) \cap (a_i^T y \leq a_i^T y_k)$  is*

$$E \left( y - \frac{1}{(m+1)(a_i^T B a_i)^{1/2}} B a_i, B - \frac{m^2}{m^2 - 1} \frac{B a_i a_i^T B}{a_i^T B a_i} \right)$$

Sadly the analysis of ellipsoid is a bit messy and the algorithm is slow in practice.

## 2. INTERIOR POINT METHODS

Interior point methods are a family of optimization methods that take steps within the interior of the feasible set. They yield the best known complexity for solving LPs, are efficient in practice and are of theoretical interest because they can have runtime bounds based on internal structure of the problem you are trying to solve.

Using an interior point method means having to drop exactness right away. We will be able to recover this exactness for LPs by projection, but we will not be able to recover it for SDPs.

**2.1. Barrier Methods.** Barrier methods work by relaxing an optimization of the form  $\min c^T x$  s.t.  $x \in S$  to  $c^T x + B(\mu, x)$  s.t.  $x \in S$ . The barrier function  $B$  serves to prevent the optimum of the unconstrained optimization from being outside the feasible set. We try to choose a barrier function  $B$  with the following properties:

- (1) Differentiable on the domain
- (2) Infinite outside the domain
- (3) Approaches zero inside the domain as  $\mu$  approaches 0

**2.2. KKT conditions.** The KKT conditions are a set of necessary optimality conditions for nonlinear programs. Given the following nonlinear program for  $f(x), g_i(x)$  convex and  $h_i(x)$  affine:

$$\begin{aligned} \min & f(x) \\ \text{s.t.} & g_i(x) \leq 0 \\ & h_i(x) = 0 \end{aligned}$$

$\bar{x}$  is an optimum implies  $\nabla f(\bar{x}) + \sum u_i g_i(\bar{x}) + \sum v_i h_i(\bar{x}) = 0$  for some  $u_i \geq 0$  and  $u_i g_i(\bar{x}) = 0$ .

**2.3. Newton's method.** Newton's Method is a second order optimization technique (uses hessian information) that works by estimating the zero of the gradient. It uses a second order Taylor approximation of the function  $f(x) \approx f(\bar{x}) + \nabla f(\bar{x})^T (x - \bar{x}) + \frac{1}{2} (x - \bar{x})^T H(\bar{x}) (x - \bar{x})$ . Using a Taylor approximation of the gradient we get:

$$\begin{aligned} \nabla f(x) &\approx \nabla f(\bar{x}) + H(\bar{x})(x - \bar{x}) \\ 0 &= \nabla f(\bar{x}) + H(\bar{x})(x - \bar{x}) \\ -\nabla f(\bar{x}) &= H(\bar{x})(x - \bar{x}) \\ -H(\bar{x})^{-1} \nabla f(\bar{x}) &= x - \bar{x} \\ \bar{x} - H(\bar{x})^{-1} \nabla f(\bar{x}) &= x \end{aligned}$$

While each iteration of this algorithm is expensive (it requires inverting the hessian), it converges with very few iterations (quadratic convergence if you start within the appropriate basin).

**2.4. A Barrier Method for Linear programming.** Consider the following linear program:

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

With the dual

$$\begin{aligned} \min \quad & b^T \pi \\ \text{s.t.} \quad & A^T \pi + s = c \\ & s \geq 0 \end{aligned}$$

Observe that the function  $-\mu \sum \ln x_i$ , with the function infinite for  $x_i < 0$  serves as a logarithmic barrier for the constraint  $x \geq 0$ . This leads us to the following optimization for a fixed  $\mu$ :

$$\begin{aligned} \min \quad & c^T x - \mu \sum \ln x_i \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

With the dual

$$\begin{aligned} \max \quad & b^T \pi - \mu \sum \ln(s_i) \\ \text{s.t.} \quad & A^T \pi + s = c \\ & s \geq 0 \end{aligned}$$

The gradients of the objectives is simply  $c - \mu(\text{diag}(x))^{-1}e = \begin{pmatrix} c_1 + \frac{\mu}{x_1} \\ \dots \\ c_n + \frac{\mu}{x_n} \end{pmatrix}$ . The

hessian is simply  $\mu(\text{diag}(x)^{-2}) = \begin{pmatrix} \frac{\mu}{x_1^2} & & \\ & \dots & \\ & & \frac{\mu}{x_n^2} \end{pmatrix}$

The basic idea is that we decrease  $\mu$ , solve the optimization problem and repeat. Eventually  $\mu$  will converge to 0 and the optimum of our approximate optimization will approach the optimum of the original optimization problem.

**2.5. Barrier Method for LPs using Newton Iterations.** We present the following textbook algorithm [1]:

- (1) let  $(x^0, \pi^0, s^0)$  be a  $\beta$ -approximate solution to  $\mu^0$  for some  $0 \leq \beta < 1$ .
- (2) set  $\mu_k = \left(1 - \frac{\sqrt{\beta} - \beta}{\sqrt{\beta} + \sqrt{n}}\right) \mu_{k-1}$
- (3) Compute the newton step by solving the following system of equations:

$$\begin{aligned} c - \nabla + H\Delta x &= A^T \pi_k \\ A\Delta x &= 0 \end{aligned}$$

- (4) set

$$\begin{aligned} x_k &= x_{k-1} + \delta x \\ s_k &= c - A^T \pi_{k-1} \end{aligned}$$

- (5) repeat steps 2-4 until desired precision is reached

**2.6. Newton Iteration.** One of the clever parts of this algorithm is the approximate newton iteration. We recall our original cost function, and write down it's second order approximation:

$$\begin{aligned} c^T x - \mu \sum \ln x_i &\approx f(\bar{x}) + \nabla f(\bar{x})^T (x - \bar{x}) + \frac{1}{2} (x - \bar{x})^T H(\bar{x}) (x - \bar{x}) \\ &= c^T \bar{x} - \mu \sum \ln \bar{x}_i + \left( c + \begin{pmatrix} \frac{\mu}{x_1} \\ \dots \\ \frac{\mu}{x_n} \end{pmatrix} \right)^T (x - \bar{x}) + \dots \end{aligned}$$

$$\text{s.t. } A\Delta x = 0$$

The KKT conditions are both necessary and sufficient for the second order approximation since the only constraint is linear:

$$\begin{aligned} c - \nabla f + H_f &= A^T \pi_k \\ A\Delta x &= 0 \end{aligned}$$

We can solve for  $\Delta x, \pi_k$  explicitly with  $X = \text{diag}(x)$ :

$$\Delta x = \bar{X} [I - \bar{X} A^T (A \bar{X}^2 A^T)^{-1} A \bar{X}] \left( e - \frac{1}{\mu_k} \bar{X} c \right)$$

Now we examine the KKT conditions for the original problem:

$$\begin{aligned} Ax &= b, x > 0 \\ A^T \pi + s &= c \\ \frac{1}{\mu} X S e - e &= 0 \end{aligned}$$

That however is hard to solve since it is not quite linear in the variables. Thus we use the following  $\beta$ -approximate form of the KKT conditions:

$$\begin{aligned} Ax &= b, x > 0 \\ A^T \pi + s &= c \\ \left\| \frac{1}{\mu} X s - e \right\| &\leq \beta \end{aligned}$$

**Theorem 2.1.** *If  $(\pi, s)$  is a  $\beta$ -approximate solution for a particular  $\mu$ , then the duality gap satisfies:*

$$n\mu(1 - \beta) \leq c^T x - b^T \pi \leq n\mu(1 + \beta)$$

**Theorem 2.2.** *If  $(x, \pi, s)$  is a  $\beta$ -approximate solution for a particular  $\mu$  then the result of the newton step is a  $\beta^2$  approximate solution.*

## REFERENCES

- [1] Robert M. Freund, Jorge R. Vera Fundamentals of Nonlinear Optimization: a Constructive Approach (unreleased draft)