Interior Point Method II

Department of Statistics

November 9, 2023

University of Seoul

Primal-Dual Interior Point Method

Interior Point Methods

Consider a convex optimization problem

$$\begin{aligned} \text{minimize} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \leq 0 \quad i = 1, \ldots, m \\ & Ax = b, \end{aligned} \tag{1}$$

where $f_0, \ldots, f_m : \mathbb{R}^n \to \mathbb{R}$ are convex and twice continuously differentiable, $A \in \mathbb{R}^{p \times n}$ with $\text{rank}(A) = p < n$.

For simplicity, assume that f_0 is quadratic and f_i for $i = 1, \cdots, m$ are linear.

The centrality condition (perturbed KKT conditions)

The KKT conditions for a relaxed problem can be understood as modified KKT conditions:

- $x = x^*(t)$ is a point on the central path if and only if there exists $\lambda = \lambda^*(t)$ and $\nu = \nu^*(t)$ such that

$$
\begin{aligned}
Ax = b, \quad f_i(x) &\leq 0, \qquad i = 1, \ldots, m \\
\lambda &> 0 \\
\nabla f_0(x) + \sum_{i=1}^{m} \lambda_i \nabla f_i(x) + A^T \nu &= 0 \\
\lambda_i f_i(x) &= -1/t, \qquad i = 1, \ldots, m
\end{aligned} \tag{2}
$$

As $t \to \infty$, the solution $x^*(t)$ converges to the optimal solution of the original problem.

When $x^*(t)$ is feasible, $\lambda_i = \lambda_i^*(t) = -1/(tf_i(x^*(t))) > 0$ for $i = 1, \cdots, m$. The primal and dual solutions are obtained by solving the equations,

$$
\begin{aligned}
\nabla f_0(x) + \sum_{i=1}^{m} \lambda_i \nabla f_i(x) + A^T \nu &= 0 \\
\lambda_i f_i(x) + 1/t &= 0, \qquad i = 1, \ldots, m \\
Ax - b &= 0.
\end{aligned}
$$

Finding an update direction

$$\text{Let } F : (x, \lambda, \nu) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \mapsto \begin{pmatrix} F_1(x, \lambda, \nu) \\ F_2(x, \lambda, \nu) \\ F_3(x, \lambda, \nu) \end{pmatrix} \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p,$$

where

- $F_1(x, \lambda, \nu) = \nabla f_0(x) + \sum_{i=1}^m \lambda_i \nabla f_i(x) + A^T \nu$
- $F_2(x, \lambda, \nu) = \lambda \circ f(x) + (1/t)\mathbf{1}$, where $\circ$ denotes the elementwise product.
- $F_3(x, \lambda, \nu) = Ax - b$.

The (nonlinear) equations are written by

$$F(x, \lambda, \nu) = 0 \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p.$$

Finding an update direction

Newton's method provides an iterative algorithm to solve the equations as

1. Set $k = 0$ and an initial $(x^{(k)}, \lambda^{(k)}, \nu^{(k)})$.
2. Compute a linear approximation of $F(x, \lambda, \nu)$ at $(x^{(k)}, \lambda^{(k)}, \nu^{(k)})$

$$\tilde{F}(x, \lambda, \nu) = F(x^{(k)}, \lambda^{(k)}, \nu^{(k)}) + \nabla F(x^{(k)}, \lambda^{(k)}, \nu^{(k)})((x, \lambda, \nu) - (x^{(k)}, \lambda^{(k)}, \nu^{(k)}))$$

3. Obtain the solution $(x^{(k+1)}, \lambda^{(k+1)}, \nu^{(k+1)})$ of the equation $\tilde{F}(x, \lambda, \nu) = 0$
4. $k \to k + 1$ and go to the second step unless the solution converges.

Finding an update direction

$$
\begin{aligned}
\nabla F(x,\lambda,\nu) &= \begin{pmatrix} \frac{\partial F_1(x)}{\partial x^\top} & \frac{\partial F_1(x)}{\partial \lambda^\top} & \frac{\partial F_1(x)}{\partial \nu^\top} \\ \frac{\partial F_2(x)}{\partial x^\top} & \frac{\partial F_2(x)}{\partial \lambda^\top} & \frac{\partial F_2(x)}{\partial \nu^\top} \\ \frac{\partial F_3(x)}{\partial x^\top} & \frac{\partial F_3(x)}{\partial \lambda^\top} & \frac{\partial F_3(x)}{\partial \nu^\top} \end{pmatrix} \\
&= \begin{pmatrix} \nabla^2 f_0(x) + \sum_{i=1}^m \lambda_i \nabla^2 f_i(x) & \nabla f_1(x) & \cdots & \cdots & \nabla f_m(x) & A^\top \\ \lambda_1 \nabla f_1(x)^\top & \lambda_1 & \cdots & \cdots & 0 & 0 \\ \lambda_2 \nabla f_2(x)^\top & 0 & \lambda_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \lambda_m \nabla f_m(x)^\top & 0 & \cdots & \cdots & \lambda_m & 0 \\ A & 0 & \cdots & \cdots & 0 & 0 \end{pmatrix}
\end{aligned}
$$

Finding an update direction

Let $\mathcal{D}\mathbf{f} = (\nabla f_1, \cdots, \nabla f_m(x)) \in \mathbb{R}^{n\times m}$, then $\nabla F(x, \lambda, \nu)$ is more simply written by

$$\nabla F(x, \lambda, \nu) = \begin{pmatrix} \nabla^2 f_0(x) + \sum_{i=1}^m \lambda_i \nabla^2 f_i(x) & \mathcal{D}\mathbf{f} & A^\top \\ \text{diag}(\lambda)(\mathcal{D}\mathbf{f})^\top & \text{diag}(\lambda) & 0 \\ A & 0 & 0 \end{pmatrix}$$

Let $(\Delta x, \Delta\lambda, \Delta\nu) = (x, \lambda, \nu) - (x^{(k)}, \lambda^{(k)}, \nu^{(k)})$ then the solution of $\tilde{F}(x, \lambda, \nu) = 0$

$$(\Delta x, \Delta\lambda, \Delta\nu) = -\nabla F(x^{(k)}, \lambda^{(k)}, \nu^{(k)})^{-1} F(x^{(k)}, \lambda^{(k)}, \nu^{(k)})$$

Feasibility conditions and residuals

- $\nabla f_0(x) + \sum_{i=1}^{m} \lambda_i \nabla f_i(x) + A^T \nu = 0$ implies that x achieves that

$$g(\lambda, \nu) = \inf_{x \in \mathcal{D}} f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \nu^\top (Ax - b).$$

  Because λ and ν are correctly given for x minimizing the Lagrangian,

$$\epsilon_d = \nabla f_0(x) + \sum_{i=1}^{m} \lambda_i \nabla f_i(x) + A^T \nu$$

  is called the dual residual.

- Similarly, $\epsilon_p = Ax - b$ is called the primal residual.
- $\epsilon_c = \lambda \circ \mathbf{f}(x) + 1/t$ is called the central residual.

Feasibility conditions and residuals

Return to our problem (2), then for a given $(x^{(k)}, \lambda^{(k)}, \nu^{(k)})$

$$\begin{aligned} F(x^{(k)}, \lambda^{(k)}, \nu^{(k)}) &= \begin{pmatrix} F_1(x^{(k)}, \lambda^{(k)}, \nu^{(k)}) \\ F_2(x^{(k)}, \lambda^{(k)}, \nu^{(k)}) \\ F_3(x^{(k)}, \lambda^{(k)}, \nu^{(k)}) \end{pmatrix} = \begin{pmatrix} \nabla f_0(x^{(k)}) + \sum_{i=1}^m \lambda_i^{(k)} \nabla f_i(x^{(k)}) + A^\top \nu^{(k)} \\ \lambda^{(k)} \circ \mathbf{f}(x^{(k)}) + (1/t)\mathbf{1} \\ Ax^{(k)} - b \end{pmatrix} \\ &= \begin{pmatrix} \epsilon_d \\ \epsilon_c \\ \epsilon_p \end{pmatrix} \end{aligned}$$

Thus we call $F(x^{(k)}, \lambda^{(k)}, \nu^{(k)})$ a residual vector.

Dual Gap and Surrogate Dual Gap

- Dual Gap (precision bound):

$$\sum_{i=1}^{m} \lambda_i^* f_i(x^*(t)) = \sum_{i=1}^{m} \left(-\frac{1}{t f_i(x^*(t))} \right) f_i(x^*(t)) = -\frac{m}{t}$$

- Surrogate Dual Gap: for given $x(t)$ and $\lambda(t)$, the surrogate dual gap is defined by

$$\sum_{i=1}^{m} \lambda_i(t) f_i(x(t)).$$

Since $t^{-1} = -\sum_{i=1}^{m} \lambda_i^*(t) f_i(x^*(t))/m$ for the dual gap, the t^{-1} in the following algorithm is determined by $t^{-1} = -\sum_{i=1}^{m} \lambda_i(t) f_i(x(t))/m$.

Primal and Dual Interior Point Method

1. Set an initial $(x^{(k)}, \lambda^{(k)}, \nu^{(k)})$
2. Compute a surrogate dual gap and set t by the dual gap.
3. Find a direction $(\Delta x, \Delta\lambda, \Delta\nu)$
4. Compute a step size θ for $(x^+, \lambda^+, \nu^+) = (x^{(k)}, \lambda^{(k)}, \nu^{(k)}) + \theta(\Delta x, \Delta\lambda, \Delta\nu)$ satisfying
   - $\lambda \geq 0$
   - $f_i(x) \leq 0$
   - the norm of residuals decrease
5. update t and go to step 3.

Example 1 (Quadratic Programming)

$$
\begin{aligned}
\min \quad & \frac{1}{2} z^\top P z + q^\top z \\
\text{subject to} \quad & \tilde{A} z + s - \tilde{b} = 0 \\
& \tilde{G} z - \tilde{h} = 0 \\
& -s \preceq 0,
\end{aligned}
$$

where $z \in \mathbb{R}^r$, $s \in \mathbb{R}^m$, $P \in \mathcal{S}^r_{++}$, $\tilde{A} \in \mathbb{R}^{m \times r}$, $\tilde{G} \in \mathbb{R}^{k \times r}$. Let $n = r + m$.

- $f_0(z,s) = \frac{1}{2}z^\top Pz + q^\top z - t^{-1}\sum_{i=1}^m \log s_i$
- $\nabla_z f_0(z,s) = Pz + q$, $\nabla_s f_0(z,s) = -t^{-1}s^{-1}$ where $s^{-1} = (1/s_1, \cdots, 1/s_m)^\top$
- $f_i(z,s) = -s_i$ for $i = 1, \cdots, m$, and $\nabla_s f_i = -e_i$ ($e_i \in \mathbb{R}^m$ is a unit vector of which the ith element is 1).
- Hessian:

$$\nabla^2 f_0(z,s) = \begin{pmatrix} P & 0 \\ 0 & t^{-1}\text{diag}(s^{-2}) \end{pmatrix} \in \mathbb{R}^{n\times n} \text{ and } \nabla^2 f_i(z,s) = 0 \in \mathbb{R}^{n\times n} \text{ for } i = 1, \cdots, m.$$

- equality constraint: $Ax = b$, where

$$A = \begin{pmatrix} \tilde{A} & I \\ \tilde{G} & 0 \end{pmatrix} \in \mathbb{R}^{(m+k)\times n},\ b = \begin{pmatrix} \tilde{b} \\ \tilde{h} \end{pmatrix} \in \mathbb{R}^{m+k},\ \text{and } x = \begin{pmatrix} z \\ s \end{pmatrix} \in \mathbb{R}^n.$$

Standard form

$$\nabla f_0(x) = \begin{pmatrix} Pz + q \\ -t^{-1}s^{-1} \end{pmatrix}, \; \sum_{i=1}^{m} \lambda_i \nabla f_i(x) = \begin{pmatrix} 0 \\ -\lambda \end{pmatrix}, \text{ and } \lambda \circ \mathbf{f}(x) = -\begin{pmatrix} \lambda_1 s_1 \\ \vdots \\ \lambda_m s_m \end{pmatrix}$$

$$\mathcal{D}\mathbf{f} = \begin{pmatrix} 0 \\ -I \end{pmatrix} \in \mathbb{R}^{n\times m}, \; \nabla^2 f_0(x) = \begin{pmatrix} P & 0 \\ 0 & t^{-1}\text{diag}(s^{-2}) \end{pmatrix} \text{ and } \nabla^2 f_i(x) = 0 \in \mathbb{R}^{n\times n}.$$

HW

Compute the Newton direction for linear programming.