Notes on solver

Oliver Hinder, Yinyu Ye

April 19, 2017

Abstract

Solver

1 Log barrier sub-problems

This paper is concerned with the following problem:

$$\min f(x) - \mu \log(a(x) - r) \tag{1a}$$
$$a(x) \geq r \tag{1b}$$

The KKT conditions for (**??**) are:

$$\nabla_x \mathcal{L}(x,y) = \nabla f(x) - \nabla a(x)^T y = 0 \tag{2a}$$
$$\mathcal{C}(x,y) = Y a(x) - \mu e = 0 \tag{2b}$$
$$a(x), y \geq 0 \tag{2c}$$

Where the Lagrangian $\mathcal{L}(x,y) := f(x) - y^T a(x)$.

We combine the log barrier merit function and the complementary conditions as follows:

$$\phi(x,y) = \psi(x) + \zeta(x,y) \tag{3}$$

With:

$$\zeta(x,y) = \frac{\|\mathcal{C}(x,y)\|_\infty^3}{\mu^2}$$

We now introduces models to locally approximate these merit functions $\nabla_x \mathcal{L}(x,y)$, ψ, $\mathcal{C}$ and ϕ respectively. To describe our approximations of a function f around the point (x,y) we use the function $\tilde{\Delta}^f_{(x,y)}(u,v)$ to denote the predicted increase in the function f at the new point $(x+u, y+v)$. Observe that we use different approximations depending on the choice of function f.

We use a typical linear approximate of $\nabla_x \mathcal{L}(x,y)$ as follows:

$$\tilde{\Delta}^{\nabla_x \mathcal{L}}_{(x,y)}(d_x, d_y) = \nabla_{x,x} L(x,y) d_x + \nabla a(x) d_y \tag{4}$$

The following function $\tilde{\Delta}^{\psi}_{(x,y)}(u)$ is an approximation of the function $\psi(x)$ at the point (x,y) and predicts how much the function ψ changes as we change the current from x to $x+u$.

$$\tilde{\Delta}^{\psi}_{(x,y)}(u) = \frac{1}{2} u^T M(x,y) u + \nabla \psi(x)^T u \tag{5}$$

With:

$$M(x,y) = \nabla^2 \mathcal{L}(x,y) + \sum_i \frac{y_i}{a(x)} \nabla a(x)^T \nabla a(x) \tag{6}$$

Note that if we set $y_i = \frac{\mu}{s_i}$ then $M(x,y) = \nabla^2 \psi(x)$ and $\tilde{\Delta}^{\psi}_{(x,y)}$ becomes the second order taylor approximation of ψ at the point x. Thus we can think of $\tilde{\Delta}^{\psi}_{(x,y)}(u)$ as a primal-dual approximation of the function ψ.

We can also build a model of the $\zeta(x,y)$ as follows:

$$\tilde{\Delta}^{\zeta}_{(x,y)}(d_x, d_y) = \frac{\|Sy + Y d_s + S d_y - \mu e\|_\infty^3 - \|\mathcal{C}(x,y)\|_\infty^3}{\mu^2} \tag{7}$$

With S a diagonal matrix containing entries of $a(x)$ and $d_s = \nabla a(x) d_x$. This model $\tilde{\Delta}^{\mathcal{C}}_{(x,y)}$ corresponds to the typical primal-dual linear model of $\mathcal{C}$ i.e. $C(x+d_x, y+d_y) \approx Sy + Yd_s + Sd_y - \mu e$.

With S and Y contain the diagonal elements of $a(x)$ and y respectively.

This allows us to approximate the change in the function ϕ at the point (x,y) as follows:

$$\tilde{\Delta}^{\phi}_{(x,y)}(d_x, d_y) = \tilde{\Delta}^{\psi}_{(x,y)}(d_x) + \tilde{\Delta}^{\zeta}_{(x,y)}(d_x, d_y) \tag{8}$$

We say an iterate (x,y) satisfies approximate complementary if $(x,y) \in \mathcal{Q}_\mu$ where $\mathcal{Q}_\mu$ is defined as follows:

$$\mathcal{Q}_\mu = \left\{ (x,y) \in R^n \times R^m : a(x) > 0, y > 0, \|\mathcal{C}(x,y)\|_\infty \leq \frac{\mu}{2} \right\} \tag{9}$$

We say the point (x,y) is a μ-scaled KKT point if $(x,y) \in \mathcal{T}_\mu$ where:

$$\mathcal{T}_\mu = \{(x,y) \in \mathcal{Q}_\mu : \|\nabla \mathcal{L}(x,y)\| \leq \mu(\|y\|_1 + 1)\} \tag{10}$$

In which case the algorithm terminates.

2 Algorithm

Let $s = a(x)$. Furthermore, let S, Y denote the diagonal matrices with entries of s and y respectively. We can linearize (2) at the iterate (x,y,s) as follows:

$$\begin{bmatrix} \nabla^2 \mathcal{L}(\hat{x},\hat{y}) & -\nabla a(\hat{x})^T & 0 \\ -\nabla a(\hat{x}) & 0 & I \\ 0 & \hat{S} & \hat{Y} \end{bmatrix} \begin{bmatrix} d_x \\ d_y \\ d_s \end{bmatrix} = - \begin{bmatrix} \nabla \mathcal{L}(x,y) \\ r \\ \mathcal{C}(x,y) \end{bmatrix} \tag{11}$$

$$\begin{bmatrix} \nabla^2 \mathcal{L}(\hat{x},\hat{y}) & -\nabla a(\hat{x})^T & 0 \\ -\hat{Y}\nabla a(\hat{x}) & 0 & \hat{Y} \\ \nabla a(\hat{x})^T \hat{S}^{-1} \hat{Y} \nabla a(\hat{x}) & \nabla a(x)^T & 0 \end{bmatrix} \begin{bmatrix} d_x \\ d_y \\ d_s \end{bmatrix} = - \begin{bmatrix} \nabla \mathcal{L}(x,y) \\ \hat{Y} r \\ \nabla a(\hat{x})^T \hat{S}^{-1} \mathcal{C}(x,y) \end{bmatrix} \tag{12}$$

$$\nabla \mathcal{L}(x,y) + \nabla a(x)^T S^{-1} \mathcal{C}(x,y) = \nabla \mathcal{L}(x,y) + \nabla a(x)^T S^{-1} (Y a(x) - \mu e) = \nabla \mathcal{L}(x, \mu S^{-1} e)$$

Observe that (**??**) may be singular or correspond to a direction that makes the log barrier objective worse. To rectify this problem we compute the direction as follows:

$$d_x = \arg \min_{\|u\|_2 \leq r} \tilde{\Delta}^{\psi}_{(x,y)}(u) \tag{13a}$$

$$d_s = \nabla a(x) d_x \tag{13b}$$

$$d_y = -S^{-1} \left(Y d_s + \mathcal{C}(x,y) \right) \tag{13c}$$

******CAREFUL WITH SIGNS i.e. should be $d_s = -\nabla a(x) d_x$, $d_y = -S^{-1}\left(Y d_s + \mathcal{C}(x,y)\right)$ ***************

It is well-known from trust region literature that there exists some $\delta \in [0, \infty)$ such that:

$$(M(x,y) + \delta I) d_x = -\nabla \psi(x) \tag{14}$$

Furthermore, by re-arranging this equation we can deduce that (d_x, d_y, d_s) satisfies a perturbed version of (**??**):

$$\begin{bmatrix} \nabla^2 \mathcal{L}(x,y) + \delta I & -\nabla a(x)^T & 0 \\ -\nabla a(x) & 0 & I \\ 0 & S & Y \end{bmatrix} \begin{bmatrix} d_x \\ d_y \\ d_s \end{bmatrix} = - \begin{bmatrix} \nabla \mathcal{L}(x,y) \\ 0 \\ \mathcal{C}(x,y) \end{bmatrix} \tag{15}$$

Algorithm 1 Primal-dual trust region step

function PRIMAL-DUAL-TRUST-REGION(x, y, r) **** $\in$ ****

$$d_x \in \arg\min_{\|u\| \le r} \tilde{\Delta}^{\psi}_{(x,y)}(u) \tag{16a}$$

$$d_s = \nabla a(x) d_x \tag{16b}$$

$$S = \mathrm{Diag}(a(x)) \tag{16c}$$

$$d_y = -S^{-1}\left(Y d_s + \mathcal{C}(x, y)\right) \tag{16d}$$

    $(x^+, y^+) \leftarrow (x + d_x, y + d_y)$
    return(x^+, y^+, d_x, d_y)
end function

Our complete algorithm is summarized as follows:

Algorithm 2 Primal-dual non-convex interior point algorithm

```
function NON-CONVEX-IPM(x^1, y^1)
    for k = 1, ..., ∞ do
        r ← R(y^k)
        repeat
            (x^+, y^+, d_x, d_y) ← PRIMAL-DUAL-TRUST-REGION(x^k, y^k, r)
            if (x^+, y^+) ∈ Q_μ then
                if (x^+, y^+) ∈ T_μ then
                    return (x^+, y^+)
                end if
            end if
            r ← r/2
        until φ(x^+) > φ(x^k) + (1/2) Δ̃^φ_(x^k, y^k)(d_x, d_y)
        x^k ← x^+
        y^k ← y^+
    end for
end function
```