EE 236B HW9
Zhiyuan Cao
304397496
03/16/2017

## T9.3

### (a)

$p^\star = 1$, with optimal point not attained.

### (b)

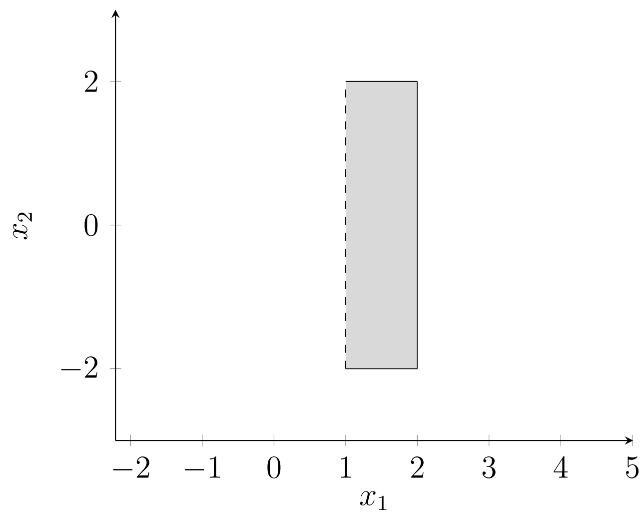

Figure 1: Level curve, constraints and feasible set

The grayed region is the sublevel set. It is not closed since on the boundary $x_1 > 1$. It is strongly convex since there exist $m \geq 0$, for instance, $m = 1$, such that the *Hessian*

$$\nabla^2 f(x) = 2I \succeq mI$$

### (c)

No, since it requires the sublevel set $S$ to be closed.

## A8.1

### (a)

For convenience we have $x = [x_1\ \sqrt{\gamma}x_2]^T$ and $y = [y_1\ y_2]^T$. For $x_1 \geq |x_2|$,

$$\sup_{||y|| \leq 1} x^T y = ||x|| = \sqrt{x_1^2 + \gamma x_2^2} \tag{1}$$

With $y$ satisfying

$$y = \frac{1}{\sqrt{x_1^2 + \gamma x_2^2}} \begin{bmatrix} x_1 \\ \sqrt{\gamma} x_2 \end{bmatrix}$$

Now since we have $x_1 \geq |x_2|$,

$$y_1 = \frac{x_1}{\sqrt{x_1^2 + \gamma x_2^2}} \geq \frac{1}{\sqrt{1+\gamma}}$$

This means that we can always find such a direction of $y$ such that it satisfies the constraint and (1) can be established. For $x_1 < |x_2|$, since we have $\gamma \geq 1$, we always have $|y_1| \leq \sqrt{\gamma}|y_2|$. Through inspection we have

$$\sup_{||y|| \leq 1, y_1 \geq 1/\sqrt{(1+\gamma)}} x^T y = \frac{x_1 + \gamma |x_2|}{\sqrt{(1+\gamma)}}$$

By taking $y_1 = 1/\sqrt{(1+\gamma)}$ and $y_2 = \pm\sqrt{\gamma}/\sqrt{(1+\gamma)}$, with $y_2$ taking the same sign as $x_2$. Since the set

$$f(x_1, x_2) = \sup\{x_1 y_1 + \sqrt{\gamma} x_2 y_2 \mid y_1^2 + y_2^2 \leq 1, y_1 \geq 1/\sqrt{(1+\gamma)}\}$$

is convex for a fixed $y$ and we just showed its equivalence to the function, the function $f(x_1, x_2)$ is also convex.

**(b)**

With $x^{(0)} = (\gamma, 1)$, we have

$$\nabla f = \frac{1}{\sqrt{x_1^2 + \gamma x_2^2}} \begin{bmatrix} x_1 \\ \gamma x_2 \end{bmatrix}$$

For convenience we write

$$t\Delta x = -t\nabla f = \frac{-t}{\sqrt{x_1^2 + \gamma x_2^2}} \begin{bmatrix} x_1 \\ \gamma x_2 \end{bmatrix} = -s \begin{bmatrix} x_1 \\ \gamma x_2 \end{bmatrix}$$

Therefore we have,

$$x^{(k+1)} = x^{(k)} - s^{(k)} \begin{bmatrix} x_1^{(k)} \\ \gamma x_2^{(k)} \end{bmatrix} \qquad s^{(k)} = \frac{t^{(k)}}{\sqrt{(x_1^{(k)})^2 + \gamma (x_2^{(k)})^2}}$$

With $x^{(0)} = (\gamma, 1)$, it fits the iterates at $k = 0$. Using induction assume at iteration $k$,

$$x_1^{(k)} = \gamma(\frac{\gamma - 1}{\gamma + 1})^k, \qquad x_2^{(k)} = (-\frac{\gamma - 1}{\gamma + 1})^k$$

$$x^{(k+1)} = \begin{bmatrix} \gamma - s^{(k)}\gamma \\ (-1)^k - (-1)^k \gamma s^{(k)} \end{bmatrix} (\frac{\gamma - 1}{\gamma + 1})^k$$

Plugging back to $f(x_1, x_2)$ and take derivative over $s^{(k)}$, set it to zero, and we have

$$s^{(k)} = \frac{2}{\gamma + 1}$$

and

$$x_1^{(k+1)} = \gamma(\frac{\gamma - 1}{\gamma + 1})^{k+1}, \qquad x_2^{(k+1)} = (-\frac{\gamma - 1}{\gamma + 1})^{k+1}$$

## A8.9

### (a)

First we rewrite the problem as

$$\text{minimize} \qquad -\sum_{\bar{y}_i = 1} \log \Phi(a_i^T x - b_i) - \sum_{\bar{y}_i = -1} \log \Phi(b_i - a_i^T x)$$

Where $\Phi(x)$ is the cumulative Gaussian distribution function, which is log-concave. Therefore this problem is convex.

### (b)

To simplify the problem, we redefine $A$ and $b$ as

$$A = \mathbf{diag}(y)A, \qquad b = \mathbf{diag}(y)b.$$

Thus the problem can be written as

$$\text{minimize} \qquad h(Ax - b),$$

where

$$h(\omega) = -\sum_{i=1}^{m} \log \Phi(\omega_i).$$

The gradient and Hessian of $f(x) = H(Ax - b)$ are

$$\nabla f(x) = A^T \nabla h(Ax - b), \qquad \nabla f(x)^2 = A^T \nabla^2 h(Ax - b) A$$

The first derivative of $h$ are

$$\frac{\partial h(\omega)}{\partial \omega_i} = \frac{-1/\sqrt{2\pi}}{\exp(\omega_i^2/2)\Phi(\omega_i)}$$

The Hessian of $h$ are

$$\frac{\partial^2 h(\omega)}{\partial \omega_i^2} = \frac{\omega_i/\sqrt{2\pi}}{\exp(\omega_i^2/2)\Phi(\omega_i)} + (\frac{1/\sqrt{2\pi}}{\exp(\omega_i^2/2)\Phi(\omega_i)})^2$$

This converges within 5 steps and $x = (0.27, 9.15, 7.98, 6.70, 6.02, 5.0, 4.30, 2.68, 2.02, 0.68)$.

```
clc;clear;
one_bit_meas_data;

A = diag(y)*A;
b = y.*b;
x = A\b;
for k = 1 : 50
   w = A*x - b;
   Phi = 0.5*erfc(-w/sqrt(2));
   Phix = 0.5*sqrt(2*pi)*erfcx(-w/sqrt(2));
   obj = -sum(log(Phi));
   grad = -A'*(1./Phix);
   hess = A'*diag((w+1./(Phix))./Phix)*A;
   v = -hess\grad;
   f = grad'*v;
   if (-f/2 < 1e-8)
       break;
   end
   t = 1;
   while(-sum(log(0.5*erfc(-(A*(x+t*v)-b)/sqrt(2)))) > ...
         obj + 0.01*t*f)
         t = t/2;
   end
   x = x + t*v;
end
```