Statistics and Machine Learning Homework1

Yuh-Jye Lee

National Taiwan University of Science and Technology

dmlab1.csie.ntust.edu.tw/Leepage/index_c.htm

Exercise 1:

(a) Solve

$$\min_{x \in R^2} \frac{1}{2} x^T \begin{bmatrix} 1 & 0 \\ 0 & 900 \end{bmatrix} x$$

using the *steep descent with exact line search*. You are welcome to copy the MATLAB code from my slides. Start your code with the initial point $x_0 = [1000\ 1]^T$. Stop until $\|x_{n+1} - x_n\|_2 < 10^{-8}$. Report your solution and the number of iteration.

Ans:

We consider solving a unconstrained quadratic programming problem. That is,

$$\min_{x \in R^n} \; f(x) = \frac{1}{2} x' Q x + p' x.$$

Let g_n be the gradient of $f(x)$ at x_n and

$$h(\lambda) = f(x_n + \lambda(-g_n)) = \frac{1}{2}(x_n - \lambda g_n)' Q (x_n - \lambda g_n) + p'(x_n - \lambda g_n).$$

Find λ^* such that $\dfrac{dh(\lambda)}{d\lambda} = 0$. We have $\lambda^* = \dfrac{g_n' g_n}{g_n' Q g_n}$.

```
function [x, f_value, iter] = grdlines(Q,p, x0, esp)
%
% min 0.5*x'Q*x+p'x
% Solving unconstrained minimization via
% steep descent with exact line search
%
%The stopping criterion:
% Either the ||gradient||_2^2 ,10^-12
%
%or    ||x_n+1 -x_n||_2<esp
```

```
flag =1; iter = 0; while flag > esp
         grad = Q*x0+p;
         temp1 = grad'*grad;
         if temp1 < 10^-12
             flag = esp
         else
             stepsize = temp1/(grad'*Q*grad);
             x1 = x0 - stepsize*grad;
             flag = norm(x1-x0);
             x0=x1;
         end;
         iter = iter+1;
 end;
 x = x0;
 f_value = 0.5*x'*Q*x+p'*x;
```

[Nonlinear Programming Homework 5 in NTUST]

Exercise 1:

Apply the steepest descent method with exact line search, starting at the point $x^{(0)} = [\gamma, 1]^T$, to solve

$$\min_{x \in R^2} f(x) = \frac{1}{2}(x_1^2 + \gamma x_2^2),$$

where $\gamma > 0$. Derive the *closed-form* expressions for the iterates $x^{(k)}$ and their function values.

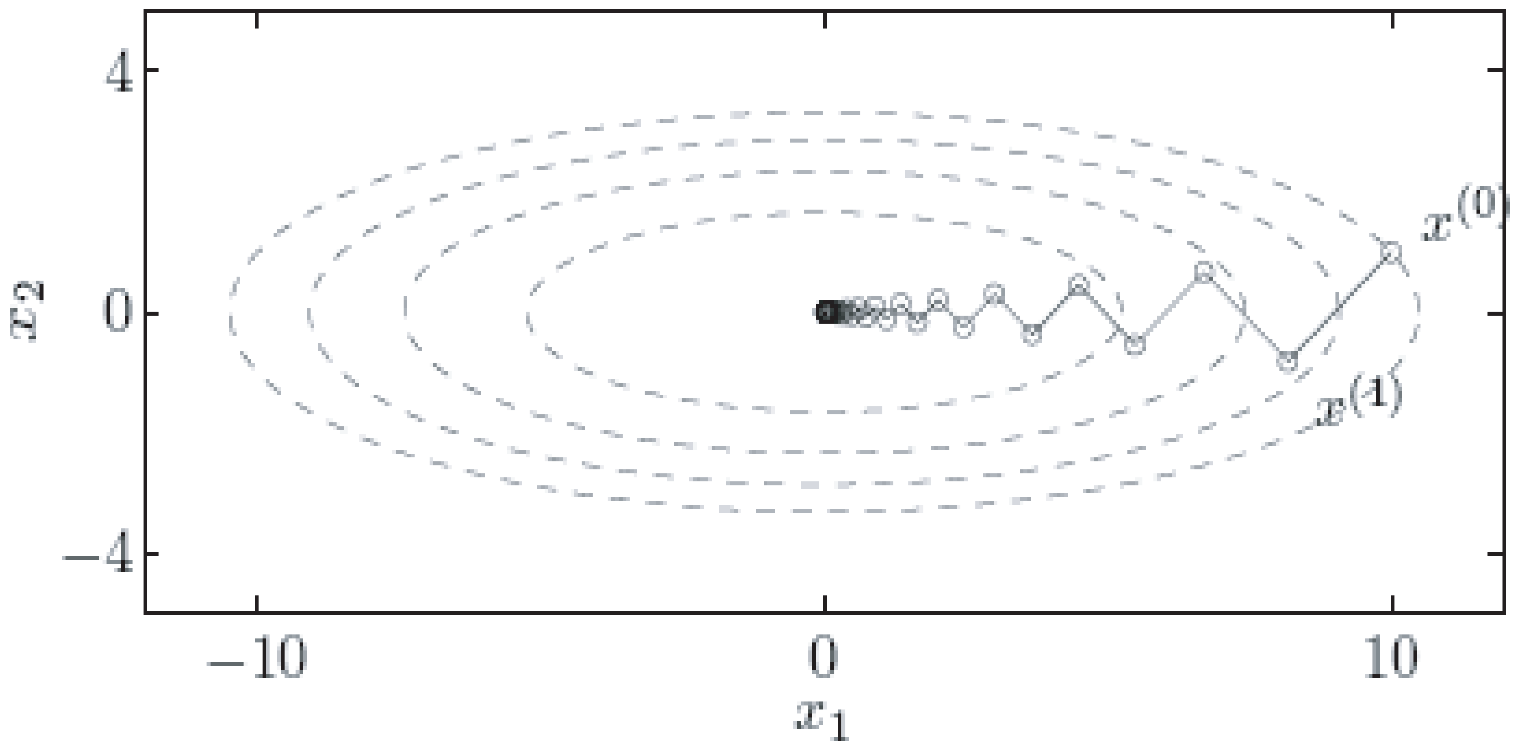

Figure 9.2 Some contour lines of the function $f(x) = (1/2)(x_1^2 + 10x_2^2)$. The condition number of the sublevel sets, which are ellipsoids, is exactly 10. The figure shows the iterates of the gradient method with exact line search, started at $x^{(0)} = (10, 1)$.

Ans:

1. $f(x) = \frac{1}{2}(x_1^2 + \gamma x_2^2)$ and $x^{(0)} = [\gamma, 1]^T \Rightarrow f(x^{(0)}) = \frac{1}{2}(\gamma^2 + \gamma)$
2. $\nabla f(x) = [x_1, \gamma x_2]^T \Rightarrow \nabla f(x^{(0)}) = [\gamma, \gamma]^T$
3. $f(x^{(0)} - \lambda \nabla f(x^{(0)})) = f([\gamma, 1]^T - \lambda[\gamma, \gamma]^T) = f([(1-\lambda)\gamma, 1 - \lambda\gamma]^T)$
$= \frac{1}{2}((1-\lambda)^2\gamma^2 + \gamma(1-\lambda\gamma)^2))$
4. $f^{'}(x^{(0)} - \lambda \nabla f(x^{(0)})) = \gamma^2\lambda - \gamma^2 + \gamma^3\lambda - \gamma^2 = 0 \Rightarrow \lambda = \frac{2}{1+\gamma}$
5. $x^{(1)} = x^{(0)} - \lambda \nabla f(x^{(0)}) = [(1-\lambda)\gamma, 1-\lambda\gamma]^T = [\gamma(\frac{\gamma-1}{\gamma+1}), -(\frac{\gamma-1}{\gamma+1})]^T$
and $f(x^{(1)}) = \frac{1}{2}(\gamma^2(\frac{\gamma-1}{\gamma+1})^2 + \gamma(\frac{\gamma-1}{\gamma+1})^2) = (\frac{\gamma-1}{\gamma+1})^{2\cdot 1} \cdot f(x^{(0)})$
6. Similarity, we get $x^{(2)} = x^{(1)} - \lambda \nabla f(x^{(1)}) = [\gamma(\frac{\gamma-1}{\gamma+1})^2, (-\frac{\gamma-1}{\gamma+1})^2]^T$ and
$f(x^{(2)}) = (\frac{\gamma-1}{\gamma+1})^{2\cdot 2} \cdot f(x^{(0)})$
7. Thus, we can derive that $x_1^{(k)} = \gamma(\frac{\gamma-1}{\gamma+1})^k$ and $x_2^{(k)} = (-\frac{\gamma-1}{\gamma+1})^k$ and
$f(x^{(k)}) = (\frac{\gamma-1}{\gamma+1})^{2\cdot k} \cdot f(x^{(0)})$

8. We can prove this result by Induction,

When $i = 1$, it is ok.

Assume $i = k$ is ok., then $x_1^{(k)} = \gamma(\frac{\gamma-1}{\gamma+1})^k$ and $x_2^{(k)} = (-\frac{\gamma-1}{\gamma+1})^k$ and $f(x^{(k)}) = (\frac{\gamma-1}{\gamma+1})^{2\cdot k} \cdot f(x^{(0)})$

When $i = k + 1$,
$f(x^{(k+1)} = f(x^{(k)} - \lambda\nabla f(x^{(k)})) = f([\gamma(\frac{\gamma-1}{\gamma+1})^k, (-\frac{\gamma-1}{\gamma+1})^k]^T - \lambda[\gamma(\frac{\gamma-1}{\gamma+1})^k, \gamma(-\frac{\gamma-1}{\gamma+1})^k]^T) = f([\gamma(\frac{\gamma-1}{\gamma+1})^k(1-\lambda), (-\frac{\gamma-1}{\gamma+1})^k(1-\lambda\gamma)]^T) = \frac{1}{2}(\gamma^2(\frac{\gamma-1}{\gamma+1})^{2k}(1-\lambda)^2 + \gamma(-\frac{\gamma-1}{\gamma+1})^{2k}(1-\lambda\gamma)^2)$

$f'(x^{(k)} - \lambda\nabla f(x^{(k)})) = \gamma^2(\frac{\gamma-1}{\gamma+1})^{2k} \cdot \lambda - \gamma^2(\frac{\gamma-1}{\gamma+1})^{2k} + \gamma^3(\frac{\gamma-1}{\gamma+1})^{2k} - \gamma^2(\frac{\gamma-1}{\gamma+1})^{2k} = 0 \Rightarrow \lambda = \frac{2}{1+\gamma}$

Thus, $x^{(k+1)} = [\gamma(\frac{\gamma-1}{\gamma+1})^k(1-\frac{2}{1+\gamma}), (-\frac{\gamma-1}{\gamma+1})^k(1-\gamma\cdot(\frac{2}{1+\gamma}))]^T = [\gamma(\frac{\gamma-1}{\gamma+1})^{k+1}, (-\frac{\gamma-1}{\gamma+1})^{k+1}]^T$

And $f(x^{(k+1)}) = \frac{1}{2}(\gamma^2 \cdot (\frac{\gamma-1}{\gamma+1})^{2(k+1)} + \gamma \cdot (-\frac{\gamma-1}{\gamma+1})^{2\cdot(k+1)} = (\frac{\gamma-1}{\gamma+1})^{2\cdot(k+1)} \cdot \frac{1}{2}(\gamma^2 + \gamma) = (\frac{\gamma-1}{\gamma+1})^{2\cdot(k+1)} \cdot f(x^{(0)})$

Hence, when $i = k + 1$ **it's ok.**

By Induction,

$x_1^{(k)} = \gamma(\frac{\gamma-1}{\gamma+1})^k$ **and** $x_2^{(k)} = (-\frac{\gamma-1}{\gamma+1})^k$ **and** $f(x^{(k)}) = (\frac{\gamma-1}{\gamma+1})^{2\cdot k} \cdot f(x^{(0)})$

(b) Implement the Newton's method for minimizing a quadratic function $f(x) = \frac{1}{2}x^T Q x + p^T x$ in MATLAB code. Apply your code to solve the minimization problem in (a).

Ans:

```
function [x, f_value, iter] = newtonqp(Q,p, x0, esp)
%
% min 0.5*x'Q*x+p'x
% Solving unconstrained QP via
% Newton's method
%
%The stopping criterion:
% Either the ||gradient||_2^2 ,10^-12
%
%or    ||x_n+1 -x_n||_2<esp
```

```
flag =1; iter = 0; while flag > esp
        grad = Q*x0+p;
        temp1 = grad'*grad;
        if temp1 < 10^-12
            flag = esp
        else
            %d=inv(Q)*grad;
            d=x0+inv(Q)*p;
            x1 = x0 - d;
            flag = norm(x1-x0);
            x0=x1;
        end;
        iter = iter+1;
 end;
 x = x0;
 f_value = 0.5*x'*Q*x+p'*x;
```

Exercise 2: **Find an approximate solution using MATLAB to the following system by minimizing $\|Ax - b\|_p$ for $p = 1, 2, \infty$. Write down both the approximate solution, and the value of the $\|Ax - b\|_p$. Draw the solution points in R^2 and the four equations being solved.**

$$\begin{aligned} x_1 &+ 2x_2 &= 2 \\ 2x_1 &- x_2 &= -2 \\ x_1 &+ x_2 &= 3 \\ 4x_1 &- x_2 &= -4 \end{aligned}$$

Ans:

(a) $\|\boldsymbol{Ax}-\boldsymbol{b}\|_1$**:**

```
function [x, residual, one_error]=oneapprox(A,b)
%
%Input A: mXn matrix
%          b: m-vector
%
%Solve the problem by LP
%Output: the approximate solution of Ax=b
%              one_error =||Ax-b||_1
%
[m,n]=size(A); obj_p=[zeros(n,1); ones(m,1)];
H=[A -eye(m);-A -eye(m)]; h=[b;-b];
[sol, one_error]=linprog(obj_p,H,h);
x=sol(1:n); residual=sol((n+1):(m+n));
```

We have $\boldsymbol{x^* = [-0.6667, 1.333]'}$ **and** $\|\boldsymbol{Ax^* - b}\|_1 = \boldsymbol{3}$**.**

(b) $\|Ax - b\|_2$:

This problem is equivalent to

$$\min_{x \in R^2} \frac{1}{2}\|Ax - b\|_2^2 \Leftrightarrow \min_{x \in R^2} \frac{1}{2} x' A' A x - b' A x.$$

Hence, can use the code given in Exercise 1 (b). Please note that the objective function value returned by the code is *not* $\|Ax - b\|_2$. We have $x^* = [-0.4552, 1.6621]'$ and $\|Ax^* - b\|_2 = 2.1367$. Of course, you can solve the *normal equation*, $x^* = (A'A)^{-1}A'b$ directly.

(c) $\|\boldsymbol{Ax} - \boldsymbol{b}\|_\infty$**:**

```
function [x, inf_error,residual ]=infapprox(A,b)
%
%Input A: mXn matrix
%         b: m-vector
%
%Solve the problem by LP
%Output: the approximate solution of Ax=b
%              inf_error =||Ax-b||_inf
%
 [m,n]=size(A); obj_p=[zeros(n,1); 1];
 H=[A -ones(m,1);-A -ones(m,1)]; h=[b;-b];
 [sol, one_error]=linprog(obj_p,H,h); x=sol(1:n);
 inf_error=sol((n+1)); residual=A*x-b;
```

We have $\boldsymbol{x^* = [-0.2, 1.8]'}$ **and** $\|\boldsymbol{Ax^*} - \boldsymbol{b}\|_\infty = \mathbf{1.4.}$