CSE 592 - Convex Optimization
HW 1

Mihir Chakradeo - 111462188

February 21, 2018

1 Gradient Descent without Line Search

1.1

Let us consider the condition of strong convexity, which implies the following theorem as seen in class:

$$f(x) >= f(y) + \nabla f(y)^T(x-y) + \frac{m}{2}\|x-y\|^2 \tag{1}$$

This is a convex function, so let us minimize y, by taking gradient and setting it to 0

$$p^* = f(X^*) >= f(X_t) + \nabla f(X_t)^T(X^* - X_t) + \frac{m}{2}\|X_t - X^*\|^2$$

$$= \min_Y f(X_t) + \nabla f(X_t)^T(Y - X_t) + \frac{m}{2}\|X_t - Y\|^2$$

$$\nabla f(X_t) + m(Y - X_t) = 0$$

$$Y^* = X_t - \frac{1}{m}\nabla f(X_t)$$

Substitute Y^* value in equation 1 we get:

$$p^* = f(X_t) + \nabla f(X_t)^T(-\frac{1}{m}\nabla f(X_t)) + \frac{m}{2}\| - \frac{1}{m}\nabla f(X_t)\|^2$$

That is,

$$f(X_t) - p^* \leq \frac{1}{2m}\|\nabla f(X_t)\|^2 \leq \epsilon \tag{2}$$

Now, let us consider the upper bound on the hessian, that is M-strongly smooth condition, that is:

$$\nabla^2 f(X) \leq MI$$

From the theorem seen in class, we have:

$$f(X+P) \leq f(X) + \nabla f(X)^T P + \frac{M}{2}\|P\|^2$$

For gradient descent, the following update rule is used:

$$X_{t+1} = X_t + \eta\Delta_t$$

That is,

$$X_{t+1} = X_t - \eta\nabla f(X_t)$$

Putting this value in strong convexity equation we get,

$$f(X_{t+1}) \leq f(X_t) + \nabla f(X_t)(-\eta\nabla f(X_t)) + \frac{M}{2}\| - \eta\nabla f(X_t)\|^2$$

We are doing this for a fixed $\eta = \frac{1}{M}$

$$f(X_{t+1}) \leq f(X_t) + \nabla f(X_t)(-\frac{1}{M}\nabla f(X_t)) + \frac{M}{2}\| - \frac{1}{M}\nabla f(X_t)\|^2$$

$$f(X_{t+1}) - p^* \leq \frac{1}{2M}\|\nabla f(X_t)\|^2$$

But, from condition of strong convexity, (equation 2) we can write the following:

$$f(X_{t+1}) - p^* \leq f(X_t) - \frac{2m}{2M}(f(X_t) - p^*) \leq \epsilon$$

$$f(X_{t+1}) - p^* \leq (f(X_t) - p^*)(1 - \frac{m}{M}) \leq \epsilon$$

This is just for one iteration, let us unroll the loop for $t + 1$ iterations,

$$f(X_{t+1}) - p^* \leq (f(X_t) - p^*)(1 - \frac{m}{M})^{t+1} \leq \epsilon$$

Taking Log of both sides we get:

$$T = \frac{1}{Log(\frac{K}{K-1})}Log(\frac{f(X_0) - P^*}{\epsilon})$$

1.1.1 Gradient Evaluations

At every iteration the gradient will be calculated for updating the direction Δ_x. That is, in all: T gradient evaluations

1.1.2 Function Evaluations

0 function evaluations as function evaluations are needed only when we need to calculate η. But for this case, η is fixed. So no function evaluations are necessary.

1.2

To show that the choice of a fixed step size must depend on the function or at least the magnitude of the Hessian. Let us calculate the value of η:
From the condition smooth convexity we know that:

$$f(X+P) \leq f(X) + \nabla f(X)^T P + \frac{M}{2}\|P\|^2$$

For gradient descent, the following update rule is used:

$$X_{t+1} = X_t + \eta \Delta_t$$

That is,

$$X_{t+1} = X_t - \eta \nabla f(X_t)$$

Putting this value in strong convexity equation we get,

$$f(X_{t+1}) \leq f(X_t) + \nabla f(X_t)(-\eta \nabla f(X_t)) + \frac{M}{2}\| - \eta \nabla f(X_t)\|^2$$

Let us find the value of η by minimizing over η:

$$f(X_{t+1}) \leq \min_{\eta} f(X_t) + \nabla f(X_t)(-\eta \nabla f(X_t)) + \frac{M}{2}\| - \eta \nabla f(X_t)\|^2$$

$$f(X_{t+1}) \leq \min_{\eta} f(X_t) + \|\nabla f(X_t)\|^2(-\eta + \frac{M}{2}\eta^2)$$

To minimize, take partial derivative with respect to η and set it to 0:

$$\eta = \frac{1}{M}$$

Where M is the upper bound on the magnitude of the hessian (M is the maximum eigenvalue).
This shows that for choice of a fixed η the stepsize must depend on the magnitude of the Hessian.

Furthermore,
Consider a twice differentiable and strongly convex quadratic function:

$$2x^2 + 1$$

Let $\eta = 2$, $x_0 = 100$ be the starting point. $\frac{d}{dx}f(x) = 4x$
In Gradient descent, next X is given by $X_{t+1} = X_t - \eta \nabla f(X_t)$. Let us analyze some steps of Gradient Descent on this function:
Iteration 1:

$$X_1 = 100 - 2 \times 400$$

$$X_1 = -700$$

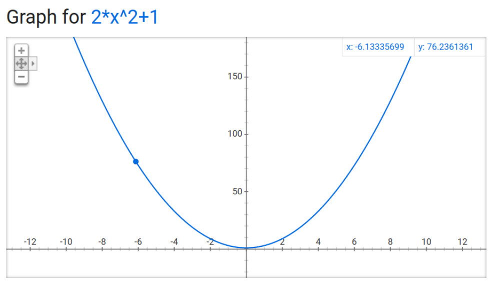

Figure 1: Quadratic Curve

Iteration 2:

$$X_2 = -700 - 2 \times (-2800)$$

$$X_2 = 4900$$

Iteration 3:

$$X_3 = 4900 - 2 \times 19600$$

$$X_3 = -34300$$

It can be seen that the X_{t+1} overshoots.

2 Newton's Method

2.1 For $x = Ay + b$, let Δx and Δy be the Newton steps for $f(x)$ and $g(y)$ respectively. Prove that $\Delta x = A\Delta y$.

Solution:
Given:

$$g(y) = f(Ay + b)$$

Take Gradient of both sides with respect to y:

$$\nabla g(y) = A^T \nabla f(Ay + b)$$

Again take Gradient of both sides with respect to y:

$$\nabla^2 g(y) = A^T \nabla^2 f(Ay + b)A$$

It is also given that $x = Ay + b$, therefore:

$$\nabla^2 g(y) = A^T \nabla^2 f(x)A$$

The Newton's step is:

$$\Delta_t = -H^{-1}G$$

Therefore,

$$\Delta_y = -[\nabla^2 g(y)]^{-1} \nabla g(y)$$

Put values of $\nabla^2 g(y)$ and $\nabla g(y)$ calculated earlier:

$$\Delta_y = -[A^T \nabla^2 f(x)A]^{-1} A^T \nabla f(x)$$

$$\Delta_y = -(A)^{-1}[\nabla^2 f(x)]^{-1}(A^T)^{-1} A^T \nabla f(x)$$

$$\Delta_y = -(A)^{-1}[\nabla^2 f(x)]^{-1} \nabla f(x)$$

Now, the Newton's step for f(x) is:

$$\Delta_x = -[\nabla^2 f(x)]^{-1} \nabla f(x)$$

Therefore,

$$\Delta_y = A^{-1} \Delta_x$$

Pre multiplying by A^{-1}:

$$\Delta_x = A\Delta_y$$

2.2 Prove that for any $\eta > 0$, the exit condition for backtracking linesearch on $f(x)$ in direction Δx will hold if and only if the exit condition holds for $g(y)$ in direction Δy.

Solution:

The exit condition for Backtracking is given as:

$$f(x + \eta\Delta_t) \leq f(x) + \alpha\eta\nabla f(x)^T\Delta x$$

Putting $x = Ay + b$, we get,

$$f(A(y + \eta\Delta_t) + b) \leq f(Ay + b) + \alpha\eta\nabla f(Ay + b)^T\Delta x$$

Putting $\Delta_x = -[\nabla^2 f(x)]^{-1}\nabla f(x)$

$$f(A(y + \eta\Delta_t) + b) \leq f(Ay + b) - \alpha\eta\nabla f(Ay + b)^T[\nabla^2 f(x)]^{-1}\nabla f(x)$$

$$f(A(y + \eta\Delta_t) + b) \leq f(Ay + b) - \alpha\eta\nabla f(Ay + b)^T[\nabla^2 f(Ay + b)]^{-1}\nabla f(Ay + b)$$

$$f(A(y + \eta\Delta_t) + b) \leq f(Ay + b) - \alpha\eta\nabla f(Ay + b)^T[\nabla^2 f(Ay + b)]^{-1}\nabla f(Ay + b)$$

$$f(A(y+\eta\Delta_t)+b) \leq f(Ay+b)-\alpha\eta[\nabla f(Ay+b)^T A][A^T\nabla^2 f(Ay+b)A]^{-1}A^T\nabla f(Ay+b)$$

From Question 2.1, we have the following results:

$$\nabla g(y) = A^T\nabla f(Ay + b)$$
$$\nabla^2 g(y) = A^T\nabla^2 f(Ay + b)A$$

Therefore:

$$f(A(y + \eta\Delta_t) + b) \leq g(y) - \alpha\eta[\nabla g(y)^T][\nabla^2 g(y)]^{-1}\nabla g(y)$$

$$f(A(y + \eta\Delta_t) + b) \leq g(y) + \alpha\eta\nabla g(y)^T\Delta_y$$

But, the RHS is exactly the stopping condition for:

$$g(y + \eta\Delta_t) \leq g(y) + \alpha\eta\nabla g(y)^T\Delta_y$$

Hence, we can say that the exit condition for backtracking line search on $f(x)$ in direction Δ_x holds if and only if the exit condition holds for $g(y)$ in direction Δ_y

2.3 Consider running Newton's method on $g(.)$ starting at some $y^{(0)}$ and on $f(.)$ starting at $x^{(0)} = Ay^{(0)} + b$. Use the above to prove that the sequences of iterates obeys $x^{(k)} = Ay^{(k)} + b$ and $f(x^{(k)}) = g(y^{(k)})$.

Solution:

Let us use Mathematical Induction to prove this

It is given that

$$x_0 = Ay_0 + b$$

The update condition for Newton's algorithm is:

$$x_{t+1} = x_t + \eta\Delta_x$$

Where $\Delta_x = -H^{-1}G$

STEP 1: Let us prove for x_1:

$$x_1 = x_0 + \eta\Delta_x$$

$$x_1 = Ay_0 + b + \eta\Delta_x$$

Also, we proved in Question 2.1 that $\Delta_x = A\Delta_y$

$$x_1 = Ay_0 + b + \eta A\Delta_y$$

$$x_1 = A(y_0 + \eta\Delta_y) + b$$

But, $y_0 + \eta\Delta_y = y_1$

$$x_1 = Ay_1 + b$$

STEP 2: Assume for x_{k-1}:

$$x_{k-1} = Ay_{k-1} + b$$

STEP 3: Proof for x_k

$$x_k = x_{k-1} + \eta\Delta_x$$

$$x_k = Ay_{k-1} + b + \eta A\Delta_y$$

$$x_k = A(y_{k-1} + \eta\Delta_y) + b$$

But, we know that $y_k = y_{k-1} + \eta\Delta_y$. Therefore:

$$x_k = Ay_k + b$$

Now, proof that the sequences of iterations obeys $f(x^{(k)}) = g(y^{(k)})$
We know that, from the Taylor Series expansion:

$$f(x_0 + \eta\Delta_x) = f(x_0) + \eta\nabla^T f(x)\Delta_x$$

$$\Delta_x = -[\nabla^2 f(x)]^{-1}\nabla f(x)$$

$$f(x_0 + \eta\Delta_x) = f(x_0) - \eta\nabla^T f(x_0)[\nabla^2 f(x_0)]^{-1}\nabla f(x_0)$$

$$f(x_0 + \eta\Delta_x) = f(Ay_0 + b) - \eta\nabla^T f(Ay_0 + b)[\nabla^2 f(Ay_0 + b)]^{-1}\nabla f(Ay_0 + b)$$

We have already seen earlier that $f(Ay+b)$ can be written $g(y)$ and using earlier results, we get:

$$f(x_0 + \eta\Delta_x) = g(y_0) - \eta\nabla^T g(y_0)[\nabla^2 g(y_0)]^{-1}\nabla g(y_0)$$

But, LHS is $f(x_0 + \eta\Delta_x) = f(x_1)$
Also,

$$g(y_1) = g(y_0 + \eta\Delta_y) = g(y_0) - \eta\nabla^T g(y_0)[\nabla^2 g(y_0)]^{-1}\nabla g(y_0) \;\; --- (4)$$

Therefore, $f(x_1) = g(y_1)$
As we proved in the earlier part of the question that $x_k = Ay_k + b$, using this and equation (4) we can say that: $f(x^{(k)}) = g(y^{(k)})$.

2.4 Prove that Newton's decrement for $f(.)$ at x is equal to Newton's decrement for $g(.)$ at y, and so the stopping conditions are also identical.

Solution: The Newton's decrement is defined as:

$$\lambda = (\nabla f(x)^T \nabla^2 f(x)^{-1} \nabla f(x))^{\frac{1}{2}}$$

So, the Newton's decrement for $f(.)$ at x is:

$$\lambda(x) = (\nabla f(x)^T \nabla^2 f(x)^{-1} \nabla f(x))^{\frac{1}{2}}$$

The Newton's decrement for $g(.)$ at y is:

$$\lambda(y) = (\nabla g(y)^T \nabla^2 g(y)^{-1} \nabla g(y))^{\frac{1}{2}}$$

Let us square to remove the squareroot:

$$\lambda^2(x) = \nabla f(x)^T \nabla^2 f(x)^{-1} \nabla f(x)$$

and

$$\lambda^2(y) = \nabla g(y)^T \nabla^2 g(y)^{-1} \nabla g(y) \;\; --- (3)$$

It is given that $x = Ay + b$. Therefore:

$$\lambda^2(x) = \nabla f(Ay+b)^T \nabla^2 f(Ay+b)^{-1} \nabla f(Ay+b)$$

$$\lambda^2(x) = [\nabla f(Ay+b)^T A][A^T \nabla^2 f(Ay+b) A]^{-1} A^T \nabla f(Ay+b)$$

From Question 2.1, we have the following results:

$$\nabla g(y) = A^T \nabla f(Ay+b)$$
$$\nabla^2 g(y) = A^T \nabla^2 f(Ay+b) A$$

Therefore:

$$\lambda^2(x) = \nabla g(y)^T \nabla^2 g(y)^{-1} \nabla g(y)$$

But, the RHS is exactly same as of $\lambda^2(y)$ (equation (3)).
Hence, we can say that $\lambda^2(x) = \lambda^2(y)$.
That is, we can say that $\lambda(x) = \lambda(y)$.
The stopping condition for Newton's algorithm is $\frac{\lambda^2}{2} \leq \epsilon$.
As $\lambda^2(x) = \lambda^2(y)$, we can say that the stopping conditions for the two are identical.

3 Programming Exercise

Submitted algorithms.py on blackboard