Problem Set 5
Unconstrained Optimization

Shekhar Kumar

Problem 9.1

An unconstrained linear objective function will have the form $A\mathbf{x} + \mathbf{b}$. where A is a $m \times n$ matrix and x, b are vectors in $\mathbf{R}^n$. The necessary condition for a local minima to exist is that the first derivative has to be equal to zero at that point.
The first derivative, of the linear function is A. If A is zero then the linear function can be written as $\mathbf{b}$, which is a constant, and if $\mathbf{A}$ is not zero then the minima doesn't exist.

Problem 9.2

The minimization problem $||Ax - b||_2$ can be written as follows

$$\begin{aligned}
\langle Ax - b, Ax - b\rangle &= (Ax - b)^T(Ax - b) \\
(Ax - b)^T(Ax - b) &= (x^TA^T - b^T)(Ax - b) \\
&= x^TA^TAx - x^TA^Tb - b^TAx + 2b^Tb \\
&= x^TA^TAx - 2b^TAx + 2b^Tb
\end{aligned}$$

Taking the derivative of the above equation and using the fact that A^TA is always positive semidefinite:

$$\lim_{h\to 0} \frac{(x+h)^TA^TA(x+h) - x^TA^TAx - 2b^TA(x+h) - 2b^TAx}{h}$$

Solving the above limit will give us the expression given below:

$$\begin{aligned}
&2x^TA^TA - 2b^TA = 0 \\
\iff\ &x^TA^TA = b^TA \\
\iff\ &A^TAx = A^Tb
\end{aligned}$$

To check that the above linear equation gives us a condition for minima, we can take the second derivative of the original equation. It is given by $2A^TA$ which is positive definite.

Therefore, the original equation satisfies the necessary and sufficient conditions for a minima.

Problem 9.3

We were taught the following methods:-

1 Steepest Descent

2 Gradient Descent

3 Newton

4 Quasi-Newton

5 Conjugate Gradient

The gradient descent method takes smallest amount of computational power per step, but it takes a larger number of steps to converge on average. The steepest descent method is very fast but it requires calculating a further optimization problem to identify the optimal α. Newton methods converge in fewer steps, but the steps are much more computationally expensive as they require calculation of Hessian at each step.Quasi-Newton methods try to address this problem by not requiring to calculate the Hessian at each step. Conjugate gradient method is the hybrid of these two methods.It takes n steps to solve an unconstrained quadratic optimization problem, whereas the steepest descent & gradient descent may take more steps.

If the dimension is not too big, and especially if the function is differentiable we can use Newton's method. It's often a good idea to use steepest descent to get a better starting x_0 for Newton's method or when the function is not differentiable. If the dimensionality is very large, we are forced to resort to conjugate gradient.

Ultimately using these methods are an art and intuition rather than a science and there is no method to rule them all, especially for non-linear optimization problems.

Problem 9.4

Suppose x_0 is chosen in such a fashion that $Q(Qx_0 - b) = \lambda(Qx_0 - b)$, as per the specification of the problem. Using the steepest descent method and using the fact that $Df(x)^T = Qx - b$ and the value of α_k for minimizing the $\phi(\alpha_k)$we get.

Let D be the derivative of $f(x)^T$, and λ is the eigenvalue of Q with $Df(x_0)^T = Qx_0 - b$. Then,

$$
\begin{aligned}
x_1 =& x_0 - \frac{DD^T}{DQD^T}D^T \\
=& x_0 - \frac{DD^T}{D\lambda D^T}D^T \\
=& x_0 - \frac{1}{\lambda}D^T \\
=& x_0 - Q^{-1}D^T \\
=& x_0 - Q^{-1}(Qx_0 - b) \\
=& Q^{-1}b
\end{aligned}
$$

We can prove the reverse using the same equation above. If the algorithm converges in one step then $Qx_0 - b$ is eigen vector to Q.

Problem 9.5

The application of steepest descent method is contingent on function $f(x)$ being continuous and differentiable. Therefore, we take $f(x)$ to be differentiable in its domain. We are given the following other results:

$$
\begin{aligned}
x_{k+1} - x_k &= -\alpha_k Df(x_k)^T \\
x_{k+2} - x_{k+1} &= -\alpha_{k+1} Df(x_{k+1})^T
\end{aligned}
$$

Where α_k, α_{k+1} are minimizers of $f(x_k - \alpha_k Df(x_k)^T)$ and $f(x_{k+1} - \alpha_{k+1} Df(x_{k+1})^T)$ respectively. The condition for orthogonality is

$$
\begin{aligned}
& \langle x_{k+1} - x_k, x_{k+2} - x_{k+1} \rangle = 0 \\
\iff & (x_{k+1} - x_k)^T (x_{k+2} - x_{k+1}) = 0 \\
\iff & \alpha_k \alpha_{k+1} Df(x_k)^T Df(x_{k+1}) = 0
\end{aligned}
$$

If α_k is the minimizer then the following condition needs to be satisfied:-

$$
\frac{df(x_{k+1})}{d\alpha_k} = Df(x_k)^T Df(x_{k+1}) = 0
$$

Combining the two equations above gives us our result.

Problem 9.6

Solution in the Jupyter notebook submitted with this document.

Problem 9.7

Solution in the Jupyter notebook submitted with this document.

Problem 9.8

Solution in the Jupyter notebook submitted with this document.

Problem 9.9

Solution in the Jupyter notebook submitted with this document.

Problem 9.10

Newton method iterations are give by the following rule:

$$x_{k+1} = x_k - (D^2(f(x_k)))^{-1} Df(x_k)^T$$

As seen in problem 9.2 and 9.3 above. The first derivative of a quadratic form is given by:

$$\mathbf{Qx} - \mathbf{b}$$

The second derivative is given by $\mathbf{Q}$.

Assuming that the initial value for starting the iterations is x_0, and substituting the values of first and second derivative we get:

$$\begin{aligned} x_1 &= x_0 - D^2 f(x_0)^{-1} Df(x_0) \\ &= x_0 - Q^{-1}(Qx_0 - b) \\ &= x_0 - x_0 + Q^{-1}b = Q^{-1}b \end{aligned}$$

The iteration will end here as x_1 is note dependent on x_0. Therefore, the Newton method for quadratic forms will converge to the optimal in one step, starting from any initial value.

Problem 9.12

We are given that λ represents the eigen values of matrix $\mathbf{A}$ and $\mathbf{B} = \mathbf{A} + \mu\mathbf{I}$. Let x be an eigenvector of $\mathbf{A}$. Then

$$\begin{aligned} Ax &= \lambda x \\ (B - \mu I)x &= \lambda x \\ Bx &= (\lambda + \mu)x \end{aligned}$$

This shows that eigen vectors of A and b are same and the eigen values of B are given by $\lambda + \mu$.

Problem 9.15

Using the property that $A^{-1}A = I$. We show the result below: Let

$X = (C^{-1} + DA^{-1}B)^{-1}$. Further simplifying the equations we get our result as follows:

$$\begin{aligned} & (A^{-1} - A^{-1}BXDA^{-1})(A + BCD) \\ &= AA^{-1} - A^{-1}BXDA^{-1}A + A^{-1}BCD - A^{-1}BXDA^{-1}BCD \\ &= I + A^{-1}BCD - A^{-1}BXD - A^{-1}BXDA^{-1}BCD \\ &= I + A^{-1}BCD - A^{-1}BX\big[I + DA^{-1}BC\big]D \\ &= I + A^{-1}BCD - A^{-1}BX\big[C^{-1}C + DA^{-1}BC\big]D \\ &= I + A^{-1}BCD - A^{-1}BX\big[C^{-1} + DA^{-1}B\big]CD \\ &= I + A^{-1}BCD - A^{-1}B\big[C^{-1} + DA^{-1}B\big]^{-1}\big[C^{-1} + DA^{-1}B\big]CD \\ &= I + A^{-1}BCD - A^{-1}BCD \\ &= I \end{aligned}$$

Problem 9.16

We have to use the same formula given in Problem 9.15, above to prove the result below. As suggested in the class, C in the **SMW** formula is 1 in equation given at the end of section 9.4.1.

We are given:

$$A_{k+1} = A_k + \frac{y_k - A_k s_k}{\|s_k\|^2}.1.s_k^T$$

Applying the SMW formula we get

$$A_{k+1}^{-1}$$

$$\begin{aligned}
&= A_k^{-1} - A_k^{-1}\left[\frac{y_k - A_k s_k}{\|s_k\|^2}\right]\left[1 + s_k^T A^{-1}\frac{y_k - A_k s_k}{\|s_k\|^2}\right]^{-1} s_k^T A_k^{-1} \\
&= A_k^{-1} - \left[\frac{A_k^{-1}y_k - s_k}{\|s_k\|^2}\right]\left[1 + \frac{s_k^T A^{-1} y_k - s_k^T s_k}{s_k^T s_k}\right]^{-1} s_k^T A_k^{-1} \\
&= A_k^{-1} + \left[\frac{s_k - A_k^{-1}y_k}{\|s_k\|^2}\right]\left[1 + \frac{s_k^T A^{-1} y_k - s_k^T s_k}{s_k^T s_k}\right]^{-1} s_k^T A_k^{-1} \\
&= A_k^{-1} + \left[\frac{s_k - A_k^{-1}y_k}{s_k^T s_k}\right]\left[\frac{s_k^T s_k}{s_k^T A_k^{-1} y_k}\right]^{-1} s_k^T A_k^{-1} \\
&= A_k^{-1} + \frac{(s_k - A_k^{-1}y_k)s_k^T A_k^{-1}}{s_k^T A_k^{-1} y_k}
\end{aligned}$$

We get our desired result.

Problem 9.17

$$A_k + \frac{y_k y_k^T}{y_k^T s_k} - \frac{A_k s_k s_k^T A_k}{s_k^T A_k s_k} = A_k + \begin{bmatrix} A_k s_k & y_k \end{bmatrix} \begin{bmatrix} -\frac{s_k^T A_k}{s_k^T A_k s_k} \\ \frac{y_k^T}{y_k^T s_k} \end{bmatrix}$$

Applying the SMW formula now, and assuming C to be 1 ,we will get our desired result.

Problem 9.18

Expanding $\phi_k(\alpha)$ as per the quadratic form, gives us the following equation:-

$$\phi_k(\alpha) = f(x_k + \alpha_k d_k)$$

$$\iff \frac{1}{2}x_k^T Q x_k + \alpha_k (d^k)^T Q x_k + \frac{\alpha_k^2}{2}(d^k)^T Q d^k - x_k^T b - \alpha_k (d^k)^T b$$

For finding the optimal α, taking the derivative wrt α.

$$\frac{\partial \phi_k(\alpha_k)}{\partial \alpha_k} = \alpha_k (d^k)^T Q d^k - (d^k)^T b + (d^k)^T Q x_k = 0$$

Rearranging the terms above gives us the result.

Problem 9.20

Define the basis $W_i := \{\tilde{r}^0, \tilde{r}^0, ..., \tilde{r}^{i-1}\}$ where $\tilde{r}^k := b - Qx^k$. By applying Gram-Schmit process, we can construct

$$r^k := \tilde{r}^k - \sum_{j=0}^{k-1} \frac{(r^j, \tilde{r}^k)_Q}{\|r^j\|_Q^2} r^j$$

Note that $W_i = span\{r^0, ..., r^{i-1}\}$ because every r^i can be written as a linear combination of $\tilde{r}^0, ..., \tilde{r}^i$. Also, note that with this setting we can deduce that the Conjugate Gradient method in each iteration is solving the following minimization problem;

$$\min_{x \in x^0 + W_i} f(x)$$

where the minimizer of this problem is x^i. Thus, the following problem results in $t = 0$ such that

$$\min_t f(x^i + t\tilde{r}^j)$$

with $j < i$. Thus, the first order condition of this problem is

$$Df(x^i)\tilde{r}^j = 0$$

implying the following;

$$\begin{aligned} 0 =& Df(x^i)\tilde{r}^j \\ =& (Qx^i - b)^T \tilde{r}^j \\ =& -\tilde{r}^i \tilde{r}^j \end{aligned}$$

$\forall j < i$.