Math homework 6 - OSM Bootcamp 2018

Cooper Nederhood

2018.07.30

Exercise 9.1
Let $f : \mathbb{R}^n \to \mathbb{R}$ be linear with representation A, so $f(x) = Ax$
Assume f is *not* constant. Thus, $\forall x \in \mathbb{R}^n$ and $\forall \epsilon > 0$ there exists $h < \epsilon$ such that $A(x + h) < Ax$ and thus no x can ever be a minimum.
Going the other way, assume f has a minimum. Then there exists x and there exists $\epsilon > 0$ such that for all $y \in B_\epsilon$ we have $Ax < Ay$. But by linearity of A this implies $Ax = b$, so f is constant.

Exercise 9.2
Let $g(x) = X^T A^T Ax - 2b^T Ax$. Then FOC imply that $g'(x) = 2A^T Ax - 2b^T A = 0 \Rightarrow A^T Ax = A^T b = 0$
$g''(x) = 2A^T A$ which is SPD implying we are at a minimum.

Exercise 9.3

1. **Gradient descent**
   Gradient descent methods are algorithms that move in the negative direction of the gradient, which is the direction in which the function is decreasing the most. The next x_{k+1} is found by moving some scalar multiple in the direction of the gradient. There are many ways to chose the scalar value, including explicitly minimizing along the vector formed by the gradient direction, or by relying on some heuristic to achieve descent. These descent methods are useful when the objection function is large and sparse, in which case the second derivatives and inverted matrices required by Newton methods are expensive to compute. Depending on the level curves, the descent methods may converge rapidly or the algorithm may move in an in-efficient zig-zag pattern, which is typical when the algorithm is in some narrow canyon.

2. **Newton and Quasi-Newton methods**
   Newton method approaches, in one way or another, minimize the 2nd order Taylor approximation of the function in some neighborhood. Thus, it's both a local approximation method **AND** a descent method - WOW! Newton converges very rapidly (like one step) if f is quadratic with symmetric Q and positive definite. However, there are many weaknesses of Newton's method which cause less than ideal. For example, if the initial guess is very far from the optimal point it may never converge. In this case, you can use gradient descent (which always improves the guess) to get close then try Newton again. Large dimensionality can make the 2nd derivatives and inversions expensive. And if the 2nd deriv is not PD, then we need to construct trust regions. Basically, the pure newton method has great convergence but with strict conditions, so there are a variety of little tweaks we can make if we don't have these ideal conditions, which we probably won't given the cold, capricious nature of our world.

3. **Conjugate gradient**
   Conjugate gradient is very much the happy medium of descent methods' slow but reliable convergence, and Newton methods' fast but assumption-reliant and expensive convergence. Conjugate gradient (GM) methods are different from Newton methods in that they never compute or store $n \times n$ Hessians or approximations. Rather, they move toward the minimizer by moving along

Q-conjugate directions. This step is relatively cheap. These methods are very useful when solving large quadratic problems where Q is symmetric and PD and sparse (so calculating the Hessian would be expensive).

Exercise 9.4
Let's assume $D(f(x_0) = Qx_0 - b$ is an eigenvector with eigenvalue λ. Then, because f is quadratic we know $\alpha = \frac{Df(x_0)Df(x_0)^T}{Df(x_0)QDf(x_0)^T} = \frac{1}{\lambda}$
Thus, $x_1 = x_0 - \frac{1}{\lambda}(Qx_0 - b)$ where the vector in parentheses is an eigenvector and thus we have:
$Qx_1 = Qx_0 - (Qx_0 - b) = b \Rightarrow x_1 = Q^{-1}b$ and we have our result.
Goin the other way,

Exercise 9.5
Because α_k minimizes $\phi(\alpha) := f(x_k - \alpha Df(x_k))$ we know that $\phi'(\alpha_k) = 0$
And by the chain rule this then implies $D\phi_k(\alpha_k) = Df(x_k)Df(x_k - \alpha Df(x_k)) = \langle Df(x_k), Df(k_{k+1})\rangle = 0$
And then, $\langle x_{k+1} - x_k, x_{k+2} - x_{k+1}\rangle = \alpha_k\alpha_{k+1}\langle Df(x_k), Df(x_{k+1})\rangle$ which combined with the above implies their inner product is zero, and thus they're orthogonal.

Exercise 9.6 - 9.9 See corresponding python code

Exercise 9.10
We know that $Df(x) = X^TQ - b$ and that $D^2f(x) = Q > 0$, implying any critical point is a minimizer. By Newton's method, $x_1 = x_0 - Q^{-1}(Qx_0 - b) = Q^{-1}b$ and if we plug this into our equation for $Df(x)$ we see that $Df(x_0) = 0$ so it's a critical point and as discussed above must therefore be a minimizer.

Exercise 9.12
Let λ_i be and e-value of A so there exists v_i such that $Av_i = \lambda_i v_i$
Thus $Bv_i = (A + \mu I)v_i = Av_i + \mu Iv_i = \lambda_i v_i + \mu v_i = (\lambda_i + \mu)v_i$
So, v_i is an eigenvector of B with corresponding eigenvalue $\lambda_i + \mu$.

Exercise 9.15
Notice that

$$BC(C^{-1} + DA^{-1}B) = B + BCDA^{-1}B = (A + BCD)A^{-1}B$$

Then,

$$(A + BCD)^{-1}BC = A^{-1}B(C^{-1} + DA^{-1}B)^{-1}$$

Thus,

$$\begin{aligned} A^{-1} = (A + BCD)^{-1}(A + BCD)A^{-1} &= (A + BCD)^{-1}(1 + BCDA^{-1}) \\ &= (A + BCD)^{-1} + [(A + BCD)^{-1}BC]DA^{-1} \\ &= (A + BCD)^{-1} + A^{-1}B(C^{-1}DA^{-1}B)^{-1}DA^{-1} \end{aligned}$$