[placeholder]6.867 Project 1[placeholder]

September 29, 2015

1

First, we discuss variations on gradient descent, including analytic gradient descent, finite difference gradient descent, and `Matlab`'s `fminunc`, which includes some variations such as adaptive step size, using a quadratic approximation instead of linear approximation, and using a 2-dimensional subspace to reduce computational complexity. First, we'll examine the analytic gradient descent, which relies on having an analytic gradient expression at all points in space.

We'll refer to three example functions: the n-dimensional quadratic "bowl", Q_n, the n-dimensional inverted Gaussian (centered at $\mu = 0$, with $\Sigma = \mathbf{I}_n$), N_n, and the n-dimensional sum of sin's, S_n. These are defined as follows (leaving off the normalization constant on the inverted Gaussian for simplicity):

$$Q_n = \|\mathbf{x}\|^2$$

$$N_n = -\exp\left(-\frac{1}{2}\|\mathbf{x}\|^2\right)$$

$$S_n = \sum_{i=1}^{n} \sin(x_n)$$

Next, we benchmark the gradient descent variations on these two functions (seeding with an initial guess of $(1, 1, \ldots)$):

Table 1: Iterations (step = .1, threshold = .001)

	Analytic Gradient			Finite Differences			`fminunc`		
n	Q_n	N_n	S_n	Q_n	N_n	S_n	Q_n	N_n	S_n
2	16	33	46	16	33	46			
5	18	60	50	18	60	50			
20	21	1	57	21	1	57			
100	25	1	64	25	1	64			

We notice that the finite differences and analytic methods yield the exact same number of iterations for all functions. This can be explained by examining the diagonal of the Hessian matrix for each of the functions - in all cases, the values are small, indicating low curvature, and thus that the function can be well-approximated by the linear finite difference approximation.

By plotting the difference between the finite differences approximation, we can see the accuracy achieved. Here, we again use $\delta = .01$. Figures 1 through 3 are plots of $(\nabla f(\mathbf{x}) - \tilde{\nabla} f(\mathbf{x}))/f(\mathbf{x})$ for $x = (0, 1)$.

Given the scales of the plots, it is clear that the finite differences method is quite accurate for these functions over these distance scales. As noted, this is generally true when the diagonal entries of the Hessian matrix are small.

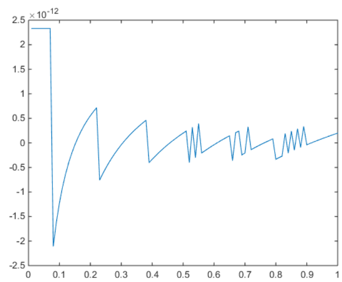

Figure 1: Q_1, scale of 10^{-12}

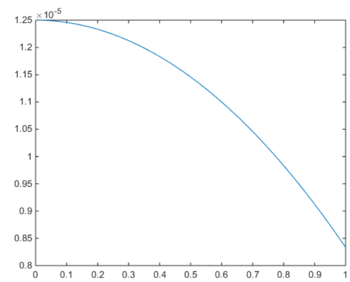

Figure 2: N_1, scale of 10^{-5}

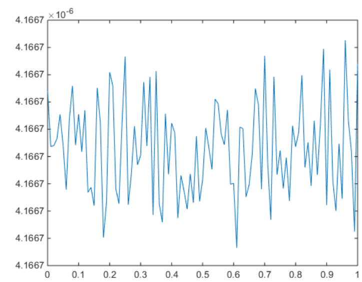

Figure 3: S_1, scale of 10^{-6}