OR 7240 – Fall 2017

Homework 4

Due: Friday, 8 December 2017

Instructions:

- This assignment is to be done individually - you must do your own work. The questions may look long but the answers are generally short and easy.

Part I. First and Second Order Optimality Conditions (20 Points)

1. **(10 Points)** For each value of the scalar β, find the set of all stationary points of the following function of the two variables x and y

$$f(x, y) = x^2 + y^2 + \beta xy + x + 2y$$

   Which of these stationary points are global minima?

2. **(10 Points)** Use optimality conditions to show that for all $x > 0$ we have

$$\frac{1}{x} + x \geq 2$$

Part II. KKT Conditions & Duality Theory (20 Points)

3. **(20 Points)** Problem 6.11 of the textbook, parts (a) and (b). For part (b) ignore the instruction and derive the dual of the given problem without considering the definition of set X.

Part III. Unconstrained Optimization (25 Points)

4. **(10 Points)** Apply Newton's method to the minimization of $f(x) = \|x\|^\beta$, where $\beta > 1$.

   (a) For what starting points and values of β does the method converge to the optimal solution?

   (b) What happens when $\beta \leq 1$?

5. **(15 Points)** Optimality conditions and coordinate–wise descent for ℓ_1-regularized minimization. We consider the problem of minimizing

$$\theta(\boldsymbol{x}) = f(\boldsymbol{x}) + \lambda \|\boldsymbol{x}\|_1,$$

   where $f : \mathbb{R}^n \to \mathbb{R}$ is convex and differentiable, and $\lambda \geq 0$. The number λ is the regularization parameter, and is used to control the trade-off between small f and small $\|\boldsymbol{x}\|_1$. When ℓ_1-regularization is used as a heuristic for finding a sparse $\boldsymbol{x}$ for which $f(\boldsymbol{x})$ is small, λ controls (roughly) the trade-off between $f(\boldsymbol{x})$ and the cardinality (number of nonzero elements) of $\boldsymbol{x}$.

(a) Show that $\boldsymbol{x} = 0$ is optimal for this problem (i.e., minimizes θ) if and only if $\left\|\nabla f(0)\right\|_\infty \leq \lambda$. In particular, for $\lambda \geq \lambda_{\max} = \left\|\nabla f(0)\right\|_\infty$, ℓ_1-regularization yields the sparsest possible $\boldsymbol{x}$, the zero vector.

Remark. The value $\lambda_{\max}$ gives a good reference point for choosing a value of the penalty parameter λ in ℓ_1-regularized minimization. A common choice is to start with $\lambda = \lambda_{\max}/2$, and then adjust λ to achieve the desired sparsity/fit trade-off.

(b) *Coordinate-wise descent.* In the coordinate-wise descent method for minimizing a convex function θ, we first minimize over x_1, keeping all other variables fixed; then we minimize over x_2, keeping all other variables fixed, and so on. After minimizing over x_n, we go back to x_1 and repeat the whole process, repeatedly cycling over all n variables.

Show that coordinate-wise descent fails for the function

$$\theta(\boldsymbol{x}) = |x_1 - x_2| + 0.1(x_1 + x_2).$$

Comment: The main goal of this problem is to educate you about these concepts otherwise, the problem itself is super easy.

Part IV. Constrained Optimization (10 Points)

6. **(10 Points)** Describe the difference between simplex algorithm and primal-dual interior point algorithm in solving a linear program.

Part V. Computational Problem (25 Points)

7. **(25 Points)** Computational problem:

(a) Randomly generate a convex quadratic function:

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T A\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{x},$$

with $\boldsymbol{b}$ randomly drawn from $[-1, 1]^n$, $n = 100$, and $A = U^T DU$, where U is a random orthonormal matrix, i.e. $U^T U = I$, and D is a random nonnegative diagonal matrix.

(b) Implement the classical gradient descent method using the following step size rules: exact line minimization, constant value $\alpha = 0.3$, and Armijo.

Comment: Implement your work and plot your figures in Matlab. Your submitted work should include a brief description of your implementation, a few simulation figures illustrating the performance of the step size rules. For example, you can plot the function values vs iteration numbers.