Homework 4

Convex Optimization 10-725

Due Friday October 25 at 11:59pm

Submit your work as a single PDF on Gradescope. Make sure to prepare your solution to each problem on a separate page. (Gradescope will ask you select the pages which contain the solution to each problem.)

Total: 84 points (+ 5 bonus points)

1 Newton's method convergence analysis (18 points)

In this problem, we will prove that Newton's method obtains a (local) quadratic rate of convergence under suitable assumptions.

1.1 Univariate setting (5 points)

As a warm up, let's start by looking at the univariate setting, where we are minimizing function $f : \mathbb{R} \to \mathbb{R}$ that is convex and three times continuously differentiable. Assume that $f''(x) \geq C_1 > 0$ and $|f'''(x)| \leq C_2$ for all x. Let $x^\star$ be the global minimizer of f, and show the Newton's method iterates, under pure step sizes, satisfy

$$|x^{(k)} - x^\star| \leq \frac{C_2}{2C_1}|x^{(k-1)} - x^\star|^2,$$

for all $k = 1, 2, 3, \ldots$. What does this imply about global convergence? Hint: use a Taylor expansion of f'.

1.2 General setting (13 points)

Now let's move to the general case, where $f : \mathbb{R}^n \to \mathbb{R}$ is convex and twice continuously differentiable. Assume that ∇f is Lipschitz with parameter L, f is strongly convex with parameter $m > 0$, and $\nabla^2 f$ is Lipschitz with parameter M:

$$\|\nabla^2 f(x) - \nabla^2 f(y)\|_{\text{op}} \leq M\|x - y\|_2,$$

for all x, y, where $\|\cdot\|_{\text{op}}$ is the operator norm. Again let $x^\star$ denote the global minimizer of f, and consider Newton's method with backtracking. We will make our lives easier by assuming that we are already in the in the "pure phase", where backtracking results in steps of size $t = 1$. (Note: you will not need to use the Lipschitz condition on ∇f in what follows, but this is required for showing that we reach the "pure phase" to begin with.)

(a, 3 pts) Using that fact that f is strongly convex, prove that for all x,

$$f(x) - f(x^\star) \leq \frac{1}{2m}\|\nabla f(x)\|_2^2.$$

Hint: use the quadratic lower bound on $f(y)$ around $f(x)$ that is implied by strong convexity, then minimize both sides over y.

(b, 6 pts) Show that at an arbitrary iteration of Newton's method, the current and next iterates x, x^+ satisfy

$$\frac{M}{2m^2}\|\nabla f(x^+)\|_2 \leq \left(\frac{M}{2m^2}\|\nabla f(x)\|_2\right)^2.$$

Hint: use a Taylor expansion of ∇f.

(c, 4 pts) Suppose that it took us k_0 iterations to reach the pure phase. Note that this means by definition that

$$\|\nabla f(x^{(k_0)})\|_2 < \eta \leq \frac{m^2}{M}.$$

By iterating the inequality in part (b) and invoking the result in part (a), show that at any iteration $k > k_0$, we have

$$f(x^{(k)}) - f(x^\star) \leq \frac{2m^3}{M^2}\Big(\frac{1}{2}\Big)^{2^{k-k_0+1}}.$$

2 GLMs + Newton's method = IRLS (15 points)

In this problem we will study the convexity properties underlying exponential families and generalized linear models (GLMs), and Newton's method applied to perform maximum likelihood. Consider an exponential family density (or mass) function over $y \in D \subseteq \mathbb{R}^n$, of the form

$$f(y;\theta) = \exp\big(y^T\theta - b(\theta)\big) f_0(y). \tag{1}$$

Note $\theta \in \mathbb{R}^n$ is called the "natural parameter".

(a, 2 pts) Prove that the domain of θ, $C = \{\theta : b(\theta) < \infty\}$, is a convex set.

(b, 5 pt) Prove that $b : C \to \mathbb{R}$ is a convex function. Hint: prove that it is twice differentiable and its Hessian is positive semidefinite everywhere.

(c, 1 pt) Suppose that we model $\theta = X\beta$, where $X \in \mathbb{R}^{n\times p}$ is a feature matrix. Note that under this parametrization, the exponential family model in (1) is called a generalized linear model (GLM). Prove that the domain of β, $B = \{\beta : X\beta \in C\}$, is a convex set.

(d, 3 pts) Show that maximizing the log likelihood in (1), when $\theta = X\beta$, is equivalent to

$$\min_\beta \; -y^T X\beta + b(X\beta). \tag{2}$$

Argue that this is a convex optimization problem. What choice of b recovers linear regression? What choice recovers logistic regression?

(e, 4 pts) Let

$$\mu_\beta = \nabla b(X\beta) \quad \text{and} \quad V_\beta = \nabla^2 b(X\beta).$$

Show that pure Newton's method (with step size $t = 1$) applied to (2) is equivalent to the updates

$$\beta^+ = (X^T V_\beta X)^{-1} X^T V_\beta z_\beta,$$

where $z_\beta = X\beta + V_\beta^{-1}(y - \mu_\beta)$. Explain why this is called *iteratively reweighted least squares* (IRLS): what optimization problem (for fixed β) would have β^+ as its solution?

3 Binary sequence denoising: dual Newton method (23 points)

Recall from Q4 of Homework 3 the binary sequence denoising problem, where we are given $z_i \in \{0,1\}$, $i = 1, 2, \ldots, n$, and we can think of an underlying logistic model giving us probabilities

$$p_i = \frac{e^{\theta_i}}{1 + e^{\theta_i}}, \quad i = 1, \ldots, n.$$

We compute estimates $\hat{\theta}_i$, $i = 1, \ldots, n$ by solving the logistic fused lasso problem:

$$\min_{\theta} \sum_{i=1}^{n} \big(- z_i \theta_i + \log(1 + e^{\theta_i}) \big) + \lambda \sum_{i=1}^{n-1} |\theta_i - \theta_{i+1}|. \tag{3}$$

By setting $y_i = 2z_i - 1$, $i = 1, \ldots, n$ and $D \in \mathbb{R}^{(n-1) \times n}$ to be a bidiagonal matrix with lower and upper diagonals -1 and 1, the problem (3) is equivalent to

$$\min_{\theta} \sum_{i=1}^{n} \log(1 + e^{-y_i \theta_i}) + \lambda \|D\theta\|_1. \tag{4}$$

Recall also that the dual of (4) is

$$\begin{aligned} \min_{u} \quad & \sum_{i=1}^{n} \Big(y_i (D^T u)_i \log(y_i (D^T u)_i) + (1 - y_i (D^T u)_i) \log(1 - y_i (D^T u)_i) \Big) \\ \text{subject to} \quad & 0 \leq y_i (D^T u)_i \leq 1, \ i = 1, \ldots, n, \ \|u\|_\infty \leq \lambda. \end{aligned} \tag{5}$$

and the relationship between the solutions in (4), (5) is

$$\theta_i = y_i \log \left(\frac{1 - y_i (D^T u)_i}{y_i (D^T u)_i} \right), \ i = 1, \ldots, n. \tag{6}$$

Now come the questions.

(a, 15 pts) Write down the gradient and Hessian of the "softened" dual problem

$$\begin{aligned} \min_{u} \ & \sum_{i=1}^{n} \Big(y_i (D^T u)_i \log(y_i (D^T u)_i) + (1 - y_i (D^T u)_i) \log(1 - y_i (D^T u)_i) \Big) \\ & - \frac{1}{\tau} \cdot \sum_{i=1}^{n} \Big(\log(y_i (D^T u)_i) + \log(1 - y_i (D^T u)_i) \Big) - \frac{1}{\tau} \cdot \sum_{i=1}^{n} \Big(\log(\lambda - u_i) + \log(u_i + \lambda)) \Big). \end{aligned} \tag{7}$$

Implement Newton's method to solve (7), using backtracking to determine the step size at each iteration, and stopping when the difference in criterion value across successive iterations is less than a user-specified tolerance level ϵ. Remember to attach all of your code in an appendix to your homework. Hint: you will need to solve an LP in order to initialize Newton's method at a feasible dual point; recall Q4(c) of Homework 3.

After implementing it, run your algorithm on the data `binseq.txt` from Homework 3. Use $\tau = 10^3$. Also use $\lambda = 20$ as the tuning parameter, $\beta = 0.8$ as the contraction factor for backtracking, and $\epsilon = 10^{-6}$ as the stopping tolerance. Report how many iterations it took to converge, and compare this to the 50,000 iterations it took gradient descent to reach rough approximate solution in Q4(d) of Homework 3. It should be much, much less! After using the primal-dual relationship (6) to get a primal solution from your dual solution, compute the estimated probabilities $\hat{p}_i$, $i = 1, \ldots, n$, and plot them. On the same figure, plot the estimated probabilities from the solution obtained by running

proximal gradient descent directly on the primal, from Q4(a) of Homework 3. Do these two look similar?

(b, 8 pts) Now (modifying your primal proximal gradient and dual Newton implementations if needed, so that all of the intermediate iterates in the algorithm are saved), rerun primal proximal gradient descent and dual Newton's method, each with $\epsilon = 0$, and for 100 iterations. Compute the primal criterion $f(\theta^{(k)})$ at each iterate of $\theta^{(k)}$ of these algorithms. Take the minimum observed criterion value across these two algorithms, subtract 10^{-6}, and call this $f^\star$. Then plot the criterion gap as a function of iteration number, for both primal proximal gradient and dual Newton, on the same figure. Make sure the y-axis is on a log scale. What do you notice about how fast each algorithm converges? What do you notice about the point at which the dual Newton method converges to? It should noticeably be suboptimal; explain why.

Repeat the same analysis (rerunning the algorithms, computing the criterions, computing $f^\star$, producing a figure) when $\lambda = 0.02$. Address the same questions; comment on the differences you see to the case $\lambda = 20$. **Bonus (2 pts):** explain why the differences between large and small λ cases occur.

4 Binary sequence denoising: dual barrier method (28 points)

(a, 10 pts) Implement the barrier method to solve the logistic fused lasso dual (5). With each outer iteration of the barrier method, you will solve the barrier problem (7) via Newton's method, as you implemented in Q3(a). Your implementation should take an initial value τ_0 for the barrier parameter, a multiplicative update factor $\mu > 1$ for the barrier parameter, and an "outer" tolerance ϵ_{outer} for stopping when the duality gap is smaller than this value. Remember to attach all of your code in an appendix to your homework.

Run your implementation on the binseq data (`binseq.txt` from Homework 3), with $\lambda = 20$, the "outer" parameters set at $\tau_0 = 5$, $\mu = 10$, and $\epsilon_{\text{outer}} = 10^{-8}$, and the "inner" parameters for Newton's method set as in Q3(a) ($\beta = 0.8$ for the contraction factor for backtracking, and $\epsilon_{\text{inner}} = 10^{-6}$ for the inner stopping tolerance). Report how many total iterations it took the barrier method to converge (sum of the Newton steps over all outer iterations). After using the primal-dual relationship (6) to get a primal solution from your dual solution, compute the estimated probabilities $\hat{p}_i$, $i = 1, \ldots, n$, and plot them. On the same figure, plot the estimated probabilities from the solution obtained by primal proximal gradient descent (from Q4(a) of Homework 3), and dual Newton's method (from from Q3(a) of this homework). Does the dual barrier method solution look closer to the primal proximal gradient solution than the dual Newton's method solution does?

Hint: storing D as a sparse and/or structured (banded) matrix should make a big difference in computational speed, both because multiplying by D or D^T should be much faster, and (more importantly) because once things are properly set up, solving a linear system in the Hessian should be much, much faster.

(b, 8 pts) Rerun your primal proximal gradient descent implementation on the binseq data, over 80 values of λ in between 0.001 and 200, equally spaced on the log scale. Starting from the largest λ value to the smallest, run proximal gradient using both warm starts: initializing from the previously computed solution, and using cold starts: initializing from $\theta = 0$. For the rest of the parameter values, use the same defaults as in Q4(a) of Homework 3 ($t = 1$ as the initial step size before each backtracking loop, $\beta = 0.8$ as the contraction factor in backtracking, and $\epsilon = 10^{-6}$ as the stopping tolerance), and use a maximum number of 1000 iterations.

For each strategy: warm/cold starts, record the total number of iterations (sum the number of backtracking iterations, i.e., prox evaluations, performed by the algorithm) at each value of λ. Remember that the prox evaluation is more or less the fundamental unit of computation for proximal gradient descent. Also record the final criterion value (upon convergence or termination at 1000

iterations), for later analysis (not in this question). Plot the number of total iterations taken by the algorithm as a function of λ, overlaying the curves for both warm and cold starts, and with the x-axis on a log scale. Do you see a difference? And more broadly, what do the curves portray about the difficulty of solving the problem (4) as a function of λ?

(c, 8 pts) Repeat the analysis as in part (b), but using your dual barrier method implementation. Now you will go from the smallest λ value to the largest. For warm starts, as before, you will just use the previously computed solution as your initial point (note that in the other direction, the warm starts would not be feasible). For the cold starts, you can simply initialize using a single fixed feasible point u_0 obtained by solving the feasibility LP at the smallest λ value, once at the start, which will then remain feasible for all larger λ. For the rest of the parameter values, use the same defaults as in Q4(a) of this homework, and use a maximum number of 100 iterations.

For each strategy: warm/cold starts, record the total number of iterations (sum the number of Newton steps) at each value of λ. Remember that the Newton step is more or less the fundamental unit of computation for the barrier method. Also record the final criterion value (upon convergence or termination at 100 iterations), again for later analysis (not in this question). Plot the number of total iterations taken by the algorithm as a function of λ, overlaying the curves for both warm and cold starts, and with the x-axis on a log scale. Comment on the difference between the strategies, and any differences to the results for the proximal gradient case in part (b).

(d, 2 pts) Plot the difference in final criterion values between primal proximal gradient and dual barrier method, as a function of λ, from your computations parts (b) and (c) (use the criterions from the warm starts). Which algorithm is more accurate, and are there any noticeable patterns?

Bonus (3 pts): What is the difference between computational complexity of one call to the prox and one Newton step? Are they of the same order (as a function of n)? Explain, and then perform numerical timings to support your arguments.