Name:

COMS 4772 Homework Set 2

(1) Binary logistic regression can formulated as the following optimization problem:

$$\min_{\theta} \sum_{t=1}^{T} \log(1 + \exp(-y_t x_t^T \theta))$$

where $y_t \in \{-1, 1\}$ are class labels, x_t are feature vectors in $\mathbb{R}^n$, and $\theta \in \mathbb{R}^n$ is the vector of unknown weights. For mathematical convenience, we can define

$$\tilde{x}_t = y_t x_t,$$

multiplying the features by their corresponding labels to decrease the notational burden.

Consider a ridge regularized logistic regression problem, where we impose a 2-norm constraint on the weights vector:

$$\min_{\theta} \sum_{t=1}^{T} \log(1 + \exp(-\tilde{x}_t^T \theta)) \quad \text{s.t. } \|\theta\|_2 \leq \tau.$$

(a) Compute the dual of this problem.

Given a primal in the form of $\min_x c^\top x + k(x) + h(b - Ax)$, the dual is given by $\max b^\top z - h^*(z) - k{*}(A{*}z - c)$ where $f{*}(x)$ is the convex conjugate of f which is $f^*(x) = \sup_x y^\top x - f(x)$.

$b = 0, c = 0, x = \theta, h(v) = \sum_{t=1}^{T} \log(1 + \exp(v_i))$.
Define matrix A with rows $A_t = y_t x_t^\top = \tilde{x}_t^\top$. Hence, $h(A\theta) = \sum_{t=1}^{T} \log(1 + exp(-\tilde{A}_t^\top \theta))$.
k represents our constraint function by setting $k = \delta(x|x \in \mathcal{B}_2(\tau)$.

The convex conjugate of the logistic loss function, where $f(x) = \ln(1 + e^{-x})$, can be written as $f^*(y) = -y\ln(-y) + (1 + y)\ln(1 + y)$ computed from using σ^{-1}, the inverse sigmoid function $(x = -ln(\frac{1-y}{y})$.

The convex conjugate of the indicator function is the support function: $k^*(x) = \sup_y y \top x$.
Hence, the dual looks as follows:
$\max_{x,y} -y\ln(-y_i) + (1 + y)\ln(1 + y) - x^\top(A^\top y - c)$ s.t. $x \in \mathbb{B}(\tau)_2$

(b) What is the dimension of the dual variable? Briefly discuss the merits of the primal vs. dual formulations from the point of view of algorithmic development.

$y \in \mathbb{R}^T$. Depending on the dimension of the data and problem formulation, optimizing over the dual would benefit computational efficiency when there are less parameters(unknowns) to compute compared to the primal.

(c) If instead of $\|\theta\|_2 \leq \tau$, we had decided to impose the constraint

$$\mathbf{-1} \leq \theta \leq \mathbf{1}$$

how does the dual change?

As k represented our previous constraint, our k in the primal formulation is changing from an indicator of a 2-ball to an inf-ball, B_∞ to correspond to $\|\theta\|_\infty \leq 1$.
$\max\limits_{x,y} -y\ln(-y) + (1+y)\ln(1+y) - x^\top(A^\top y - c)$ s.t. $x \in \mathbb{B}(1)_\infty$

(2) Recall that the prox operator is defined by

$$\operatorname{prox}_g(y) = \min_x \frac{1}{2}\|x-y\|^2 + g(x).$$

(a) Show that

$$\operatorname{prox}_{g^*}(y) = y - \operatorname{prox}_g(y)$$

(b) Use part (a) to compute

$$\operatorname{prox}_{\lambda\|\cdot\|_1}(y).$$

(3) In class, we discussed iterative soft thresholding for solving the problem

$$\min_x \frac{1}{2}\|Ax - b\|^2 + \lambda\|x\|_1.$$

In this problem, you are going to apply this algorithm to sparse logistic regression models, and also try a famous acceleration technique of Beck & Teboulle to improve the algorithm. The problem is *sparse* binary logistic regression:

$$\min_\theta \sum_{t=1}^{T} \log(1 + \exp(-\tilde{x}_t^T\theta)) + \lambda\|\theta\|_1.$$

where as in the previous question, $\tilde{x}_t = y_t x_t$. Just as in sparse linear regression, we add the 1-norm penalty to drive many of the coefficients down to 0.

(a) Download the starting script file, and make sure you understand the problem setup.

(b) Implement a proximal splitting method for the above problem. You may use a constant step size. At every iteration, your algorithm should print a line listing the value and iteration.

To show that you implemented the algorithm, copy and paste a run over the first 10 and last 10 iterations into a verbatim environment, as shown below, say for 100 iterations:

```
iter 1
iter 2
...
iter 10
iter 91
iter 92
iter 100
```

(c) Solve the same problem with CVX, and show that your solution (as well as the value of your solution) agrees with the CVX solution, and its value.

(d) Skim the FISTA paper: `http://mechroom.technion.ac.il/~becka/papers/71654.pdf`
On page 11, find the FISTA algorithm (the fixed step size version). Implement it for the logistic regression problem, again pasting the first 10 and last 10 iterations:

```
iter 1
iter 2
...
iter 10
iter 91
iter 92
iter 100
```

FISTA with fixed-step size:
Input: $L = L(f)$, where L is a Lipschitz-constant of ∇f.
Initialzize $y_1 = x_0$, $t_1 = 1$. For steps $k \geq 1$: $x_k = P_L(y_k)$
$t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$

$y_{k+1} = x_k + \frac{t_k - 1}{t_{k+1}}(x_k - x_{k-1})$

$P_L(y) :=$ argmin $Q_L(x, y)$ where $Q_L(x, y) = f(y) + (x - y)^\top \nabla f(y) + \frac{L}{2}\|x - y\|^2 + g(x)$
Note: $Q_L(x, y) = f(y) + (x - y)^\top \nabla f(y) + L \cdot \text{prox}_{\ g}(y)$.

(e) Make a plot, comparing per-iteration progress of the two algorithms on the same problem. The x-axis of your plot should be iteration number, and the y axis the value of the objective function. Did the acceleration... accelerate anything?

(4) Consider the problem of minimizing a smooth function subject to inequality constraints:

$$\min_x f(x) \quad \text{s.t.} \quad Cx \leq c.$$

For our purposes, it is convenient to introduce nonnegative slack variables $s \geq 0$, rewriting the problem

$$\min_{x,s} f(x) \quad \text{s.t.} \quad Cx + s = c, \quad s \geq 0.$$

The Lagrangian for this problem is given by

$$\mathcal{L}(x, s, \lambda) = f(x) + \lambda^T(Cx + s - c) + \delta(s|\mathbb{R}^n_+)$$

(a) Obtain the first-order necessary condition for a local minimum of $\mathcal{L}$ in x.

$\frac{\partial \mathcal{L}}{\partial x} = \nabla_x f(x) + \lambda^\top C = 0 \Rightarrow \nabla f(x) = -\lambda^\top C$

(b) Obtain the necessary condition for a local maximum of $\mathcal{L}$ in λ.

$\frac{\partial \mathcal{L}}{\partial s} = \lambda^\top \cdot \mathbb{1}\{s \geq 0\} = 0$

(c) Argue that at any saddle point of $\mathcal{L}$, $\bar{\lambda}_i \bar{s}_i = 0$.

$\bar{\lambda}_i \bar{s}_i = 0$ satisfies KKT's complementary slackness condition. Saddle points satisfy KKT's first order necessary conditions and we see that this complementary slackness condition must hold when taking the partial of the Lagrangian with respect to s. We arrive at a saddle point and not optimality because $f(x)$ is not necessarily convex.

(d) Now consider a log-barrier modified primal problem:

$$\min_{x,s} f(x) - \mu \sum \log(s_i) \quad \text{s.t.} \quad Cx + s = c.$$

(e) Form the Lagrangian for this problem, and compute equations corresponding to first-order necessary conditions in all three variables x, s, λ. Compare these equations to the equations in parts (a-c).

$\frac{\partial \mathcal{L}}{\partial x} = \nabla_x f(x) + \lambda^\top C = 0$
$\frac{\partial \mathcal{L}}{\partial s_i} = -\frac{\mu}{s_i} + \lambda_i = 0 \Rightarrow \lambda = \frac{\mu e}{s} = \mu e^\top s^{-1}$
$\frac{\partial \mathcal{L}}{\partial \lambda} = Cx + s - c = 0$

(5) **Bonus**.

(a) Design a Newton method to directly solve the optimality conditions in part (e) of (4). You will be able to represent the higher order system as a 3×3 block matrix, with blocks for x, s, λ. Once you have the general form, please specify it to the case

$$\min_x \frac{1}{2}\|Ax - b\|^2 \quad \text{s.t.} -\mathbf{1} \le x \le \mathbf{1}.$$

(b) Implement your Newton method to solve the log-barrier regression problem for a fixed value of μ, and verify that your solution matches that of CVX. Be careful with the step length - don't let your updated s components go negative. To initialize, set all components of s and λ to 10.

To show you implemented the method, paste the iterations in a verbatim environment, and also show that you get the same value as CVX. At each iteration, output the iteration number, the value of the log-barrier objective, the value of μ, and/or the norm of the KKT system in part (e) of (4) that you are trying to drive to 0.

(c) Modify your algorithm to divide μ by 10 every other iteration. Again, paste your iterations into a verbatim environment in this document, and check that you got the same solution as CVX on the box-constrained regression problem. If so, you just implemented your first primal-dual interior point method.

(d) Write a proximal gradient method for the box-constrained regression problem, and make a plot of function value vs. iteration comparing this method to your interior point method.