Sprind 2018: CS-UA 473
Machine Learning
Assignment 1 (due Feb. 28 , 2018)
Kexin Huang
kh2383

1. $D_{log}(y, X; M) = \frac{1}{log2} log(1 + exp(-s(y, x, M)))$, here note that, $y = \{-1, 1\}$and $s = yw^t X$.
   As y only takes two values, we can divide the above equation to two parts:
   $D_{log}(y, X; M) = \frac{1}{log2} - log(1 + exp(-s(y, x, M)))^{-1}$
   $= -\frac{1}{log2} * (log\sigma(-yw^t X))$
   $= -\frac{1}{log2} * [(\mathbb{1}_{y=1} log(\sigma(-1 * w^t * X)) + \mathbb{1}_{y=-1} log(\sigma(-(-1) * w^t * X))]$
   $= -\frac{1}{log2} * [(\mathbb{1}_{y=1} log(\sigma(-1 * w^t * X)) + \mathbb{1}_{y=-1} log(\sigma(+w^t * X))]$
   note that in logistic regression, $y' = \{0, 1\}$where 0 corresponds to -1, 1 corresponds to 1, we can see the identity function $\mathbb{1}_{y=1}$ is exactly what is y' as $\mathbb{1}_{y=1} = 1$ when $y = 1, y' = 1$, and $\mathbb{1}_{y=1} = 0$ when $y = -1, y' = 0$ and same case for $\mathbb{1}_{y=-1} = (1 - y')$ where as $\mathbb{1}_{y=-1} = 1$ when $y = -1, (1 - y') = 1$, and $\mathbb{1}_{y=-1} = 0$ when $y = 1, (1 - y') = 0$. therefore, above equation turns into following:
   $= -\frac{1}{log2} * [y' * log(\sigma(-1 * w^t * X)) + (1 - y') * log(\sigma(+w^t * X))]$
   Now let's look into $\sigma(+w^t * X)$, this term equals to $(1 - \sigma(-w^t * X))$ because $(1 - \sigma(-w^t * X)) = \frac{exp(-w^t * X)}{1 + exp(-w^t * X)}$times up and bottom with $exp(w^t * X)$, we get $\frac{exp(-w^t * X)}{1 + exp(-w^t * X)} = \frac{1}{exp(w^t * X) + 1} = \sigma(+w^t * X)$, therefore, we can derive

   $D_{log}(y, X; M) = -\frac{1}{log2} * [y' * log(\sigma(-1 * w^t * X)) + (1 - y') * log(1 - \sigma(-w^t * X)] = D_{logisticregression}$

2. Yes, we cannot directly use a gradient-based algorithm as D is not differentiable at the critical point where gradient from left and right is not equal, here I propose several solutions with their own advantages:
   1. As it is differentiable except critical point, for practical purpose, if it is a continuous function on R, then the probability of hitting the critical point is relatively small in most case, so we can just define a gradient scheme D'(X)=1when score ¡1, and D'(X)=0 when score¿1 and D'(X)=c where c=[0,1], when score=1, this achieves the practical purpose of minimize the D. However, it is not mathematically rigorous.
   2. One way to achieve gradient in a rigorous way is to use subgradient as there exist subgradient for this D and it is an approximation for the gradient at that point. There is better way:
   3. We can change D a little bit by doing smoothing so that it is differentiable everywhere. After some research on Google, Rennie and Srebro's smoothing do the trick as it makes gradient to be 0 at score=1.
   4. We can also try to use Squared Hinge Loss $[h(x)]^2$ as $(h[(x)]^2)' = 2h(x)h'(x)$ where h(x)=0 when score is 1 and x=0 so it is differentiable at score=1

3. Model Selection

a. We should choose the model performs the best on validation set. i.e $M_t^{(i)} = argmin_{i=1,2 and t=1...T} R_{val,t}^{(i)}$

b. We should report the test set error for the best model we choose, i.e. $R_{test,t}^{i} = argmin_{i=1,2 and t=1...T} R_{val,t}^{(i)}$