CSCI 5525: Machine Learning

Homework 4

Jingxiang Li

December 2, 2015

Problem 1.a

The objective function is defined as

$$f(w) = h(w) + \lambda g(w) = -\sum_{i=1}^{N} \left\{ y_i w^T x_i - \log(1 + \exp(w^T x_i)) \right\} + \lambda ||w||_2^2$$

Note that $h(w)$ is separable, hence we could separate it into m mini-batches of n points each, where $N = mn$. In ADMM each mini-batch maintains its own parameter w_h, and let S_h be the set of points in mini-batch h, the objective function becomes

$$\begin{aligned} f(w) &= \sum_{h=1}^{m} h(w_h) + \lambda g(w) \\ &= -\sum_{h=1}^{m} \sum_{i \in S_h} \left\{ y_i w_h^T x_i - \log(1 + \exp(w_h^T x_i)) \right\} + \lambda ||w||_2^2 \end{aligned}$$

Then we can derive the Augmented Lagrangian of this objective function as follows

$$L = -\sum_{h=1}^{m} \sum_{i \in S_h} \left\{ y_i w_h^T x_i - \log(1 + \exp(w_h^T x_i)) \right\} + \lambda ||w||_2^2 + \sum_{h=1}^{m} \left\{ \langle u_h, w_h - w \rangle + \rho/2 ||w_h - w||_2^2 \right\}$$

where u_h is the Lagrangian multiplier for constraint $w_h = w$

Next, we can derive the ADMM updates for this optimization problem

1. $w_h^{k+1} = \arg\min_{w_h} \sum_{i=S_h} \left\{ y_i w_h^T x_i - \log(1 + \exp(w_h^T x_i)) \right\} + \left\langle u_h^k, w_h - w^k \right\rangle + \rho/2 ||w_h - w^k||_2^2$
2. $w^{k+1} = \arg\min_{w} \lambda ||w||_2^2 + \sum_{h=1}^{m} \left\{ \left\langle u_h^k, w_h^{k+1} - w \right\rangle + \rho/2 ||w_h^{k+1} - w||_2^2 \right\}$
3. $u_h^{k+1} = u_h^k + \rho(w_h^{k+1} - w^{k+1})$

Problem 1.b

In the ADMM algorithm, step 1 and 3 can be executed in parallel. Because once w is updated, step 1 and step 3 can be done in each mini-batch separately, without any information from other mini-batches.

Step 2 require access to more than one mini-batch, because it needs updated w_h from all mini-batches to update w.

Problem 1.c

Yes, it is a double-loop algorithm. The outer loop is the ADMM updates listed in problem 1.a, and the inner loop is used for solving step 1 and step 2. Step 2 may have explicit solution which does not require iterative solution. However for step 1, which is known as a logistic regression problem, it's almost impossible to directly solve it by simple algebra, suggesting that iterative methods like l-BFGS are necessary for solving step 1. Hence the ADMM must be a double-loop algorithm.

Problem 2.a

First, I will show $VC(F) \geq d + 1$

To prove $VC(F) \geq d + 1$, it's sufficient to find $d + 1$ points that F is able to shatter. Let's make X as follows:

$$X = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}_{(d+1)\times d}$$

Hence for any response vector $y \in \{-1, +1\}^{d+1}$, we can directly solve the equation $Xw + w_0 = y$, where $w = [y_2 - y_1, y_3 - y_1, \ldots, y_{d+1} - y_1]^T$ and $w_0 = y_1$. Note that $w^T x + w_0 \in F$, suggesting that there exists a set of points with cardinality $d + 1$ can be shattered by F, i.e. $VC(F) \geq d + 1$

Then I will show $VC(F) < d + 2$

Note that we can define $w^* = [w_0, w]^T$, and let $x^* = [1, x]^T$, then $F = \{f \ : \ f(x) = \text{sign}(\langle w^*, x^* \rangle)\}$. Then for any set of points with cardinality $d + 2$, the design matrix X^* will have $d + 2$ rows and $d + 1$ columns, which suggests that rows are linear correlated, i.e. there exists j such that $x_j^* = \sum_{i \neq j} a_i x_i^*$. Then we know that $x_j^* w^* = \sum_{i \neq j} a_i x_i^{*T} w^*$, which means the sign of $x_j^* w^*$ is determined by $\sum_{i \neq j} a_i x_i^{*T} w^*$. Let $y_j = -\text{sign}(\sum_{i \neq j} a_i x_i^{*T} w^*)$, then F cannot shatter this set of points. Note that this argument holds for any set of points with cardinality $d + 2$, suggesting that $VC(F) < d + 2$

Since $d + 1 \leq VC(F) < d + 2$, $VC(F) = d + 1$. Q.E.D.

Problem 2.b

Note that the Rademacher complexity is defined as

$$R_n(F) = \mathrm{E}\left[\sup_{f \in F} \frac{1}{n} \sum_{i=1}^{n} \rho_i f(x_i)\right]$$

where $\rho_i, i = 1, 2, \ldots, n$ are Rademacher variables takes values +1 and -1 with probability 0.5, respectively.

Then

$$\begin{aligned} & \mathrm{E}\left[\sup_{f \in F} \frac{1}{n} \sum_{i=1}^{n} \rho_i (f(x_i') - f(x_i))\right] \\ = \ & \mathrm{E}\left[\sup_{f \in F} \frac{1}{n} \sum_{i=1}^{n} \rho_i (f(x_i')) + \frac{1}{n} \sum_{i=1}^{n} (-\rho_i)(f(x_i))\right] \\ \leq \ & \mathrm{E}\left[\sup_{f \in F} \frac{1}{n} \sum_{i=1}^{n} \rho_i (f(x_i')) + \sup_{f \in F} \frac{1}{n} \sum_{i=1}^{n} (-\rho_i)(f(x_i))\right] \end{aligned}$$

Note that the distributions of $-\rho_i$ and ρ_i are exactly the same, then

$$\begin{aligned}
& \mathrm{E}\left[\sup_{f\in F}\frac{1}{n}\sum_{i=1}^{n}\rho_i(f(x_i')-f(x_i))\right] \\
\leq\ & \mathrm{E}\left[\sup_{f\in F}\frac{1}{n}\sum_{i=1}^{n}\rho_i(f(x_i'))+\sup_{f\in F}\frac{1}{n}\sum_{i=1}^{n}\rho_i(f(x_i))\right] \\
=\ & \mathrm{E}\left[\sup_{f\in F}\frac{1}{n}\sum_{i=1}^{n}\rho_i(f(x_i'))\right]+\mathrm{E}\left[\sup_{f\in F}\frac{1}{n}\sum_{i=1}^{n}\rho_i(f(x_i))\right] \\
=\ & 2R_n(F)
\end{aligned}$$

i.e.

$$\mathrm{E}\left[\sup_{f\in F}\frac{1}{n}\sum_{i=1}^{n}\rho_i(f(x_i')-f(x_i))\right]\leq 2R_n(F)$$

Q.E.D.

Problem 3.a

For any expert $i = 1, 2, \ldots n$, we have

$$\begin{aligned} w_{T+1}(i) &\leq 1 \\ \frac{w_T(i)\beta^{\ell_T(i)}}{Z_{T+1}} &\leq 1 \\ \frac{w_{T-1}(i)\beta^{\ell_{T-1}(i)+\ell_T(i)}}{Z_T Z_{T+1}} &\leq 1 \\ \frac{w_1(i)\beta^{\ell_1(i)+\ell_2(i)+\cdots+\ell_T(i)}}{Z_2 Z_3 \ldots Z_{T+1}} &\leq 1 \end{aligned}$$

Note that $w_1(i) = 1/n$, $i = 1, 2, \ldots n$, we have

$$\begin{aligned} \beta^{\ell_1(i)+\ell_2(i)+\cdots+\ell_T(i)} &\leq n \cdot Z_2 Z_3 \ldots Z_{T+1} \\ \beta^{\ell_1(i)+\ell_2(i)+\cdots+\ell_T(i)} &\leq n \cdot \prod_{t=1}^{T} \sum_{i=1}^{n} w_t(i)\beta^{\ell_t(i)} \end{aligned}$$

Since $\beta^{\ell_t(i)} \leq 1 - (1-\beta)\ell_t(i)$, we have

$$\beta^{\ell_1(i)+\ell_2(i)+\cdots+\ell_T(i)} \leq n \cdot \prod_{t=1}^{T} \sum_{i=1}^{n} w_t(i)[1 - (1-\beta)\ell_t(i)]$$

Note that $sum_{i=1}^{n} w_t(i) = 1$, then

$$\beta^{\ell_1(i)+\ell_2(i)+\cdots+\ell_T(i)} \leq n \cdot \prod_{t=1}^{T} \left\{ 1 - \sum_{i=1}^{n} w_t(i)[(1-\beta)\ell_t(i)] \right\}$$

Here we take logarithm on both sides of the inequality

$$\log(\beta) \sum_{t=1}^{T} \ell_t(i) \leq \log(n) + \sum_{t=1}^{T} \log\left(1 - \sum_{i=1}^{n} (1-\beta) w_t(i)\ell_t(i) \right)$$

Since $\log(1+x) < x$, we have

$$\begin{aligned} \log(\beta) \sum_{t=1}^{T} \ell_t(i) &\leq \log(n) + \sum_{t=1}^{T} \left(- \sum_{i=1}^{n} (1-\beta) w_t(i)\ell_t(i) \right) \\ (1-\beta) \sum_{t=1}^{T} \sum_{i=1}^{n} w_t(i)\ell_t(i) &\leq \log(n) - \log(\beta) \sum_{t=1}^{T} \ell_t(i) \\ \sum_{t=1}^{T} \sum_{i=1}^{n} w_t(i)\ell_t(i) &\leq \frac{\log(n) + \log(1/\beta) \sum_{t=1}^{T} \ell_t(i)}{1-\beta} \\ \sum_{t=1}^{T} \boldsymbol{w}_t^T \ell_t &\leq \frac{\log(n) + \log(1/\beta) \sum_{t=1}^{T} \ell_t(i)}{1-\beta} \\ L_{\text{adapt}} &\leq \frac{\log(1/\beta)}{1-\beta} \sum_{t=1}^{T} \ell_t(i) + \frac{1}{1-\beta} \log(n) \end{aligned}$$

Notice that the above inequality holds for any $i = 1, 2, ..., n$, hence

$$L_{\text{adapt}} \leq \frac{\log(1/\beta)}{1-\beta} \min_i \sum_{t=1}^{T} \ell_t(i) + \frac{1}{1-\beta}\log(n)$$
$$L_{\text{adapt}} \leq \frac{\log(1/\beta)}{1-\beta} \min_i L_i + \frac{1}{1-\beta}\log(n)$$

Q.E.D.

Problem 3.b

If $\min_i L_i = L_{\min}$ is fixed, then it's intuitive to find β that minimizing the upper-bound of L_{adapt}. However, directly minimize the upper-bound derived from problem 3.a is difficult, we first try to further relax the upper-bound. Note that $\log(x) < x - 1, \forall x \in (0, \infty)$, then $\log(1/\beta) < (1-\beta)/\beta$, and the upper-bound becomes

$$L_{\text{adapt}} \leq \frac{1}{\beta} L_{\min} + \frac{1}{1-\beta}\log(n)$$

Let

$$f(\beta) = \frac{1}{\beta} L_{\min} + \frac{1}{1-\beta}\log(n)$$

it's easy to derive it's derivative

$$\begin{aligned}\nabla f(\beta) &= -\frac{1}{\beta^2} L_{\min} + \frac{1}{(\beta-1)^2}\log(n) \\ &= \frac{\beta^2 \log(n) - (\beta-1)^2 L_{\min}}{(\beta-1)^2\beta^2}\end{aligned}$$

Let $\nabla f(\beta) = 0$, we have

$$g(\beta) = \beta^2 \log(n) - (\beta-1)^2 L_{\min} = 0$$

which is a quadratic function of β. solution of the above equality can be obtained by simple algebra

$$\begin{cases} x_1 = \frac{L_{\min} - \sqrt{L_{\min}\log(n)}}{L_{\min} - \log(n)} \\ x_2 = \frac{L_{\min} + \sqrt{L_{\min}\log(n)}}{L_{\min} - \log(n)} \end{cases}$$

Notice that local minimum of f is always achieved at x_1. If $\log(n) > L_{\min}$, then $g(\beta)$ is convex and hence local minimum is achieved at the larger root of it, which is x_1; on the other hand, if $\log(n) < L_{\min}$, then $g(\beta)$ is concave and hence local minimum is achieved at the smaller root of it, which is still x_1. Therefore we can set optimal β^* to be

$$\beta^* = \begin{cases} 1 & \text{if } \frac{L_{\min} - \sqrt{L_{\min}\log(n)}}{L_{\min} - \log(n)} > 1 \\ 0 & \text{if } \frac{L_{\min} - \sqrt{L_{\min}\log(n)}}{L_{\min} - \log(n)} < 0 \\ \frac{L_{\min} - \sqrt{L_{\min}\log(n)}}{L_{\min} - \log(n)} & \text{otherwise} \end{cases}$$