Introduction to ML (CS771), Autumn 2020
Indian Institute of Technology Kanpur
Homework Assignment Number 1

QUESTION 1

Student Name: Debanjan Chatterjee
Roll Number: 20111016
Date: October 30, 2020

Given absolute loss regression problem with $l1$ regularization:

$$\mathbf{w}_{opt} = \sum_{n=1}^{N} |\mathbf{y}_n - \mathbf{w}^\top \mathbf{x}_n| + \lambda ||\mathbf{w}||_1 \tag{1}$$

where,

$$||\mathbf{w}||_1 = \sum_{d=1}^{D} |w_d| \tag{2}$$

Now, absolute value function on real numbers is convex. Hence, the objective loss function is a sum of convex functions, and the sum of convex functions is convex as well. Therefore, the **objective loss function is convex**.

Now, in order to derive the expressions of the sub-gradients of the function, lets break down the loss function in (1):

Let, the objective loss function be $J(w)$ and let, $L(w) = |\mathbf{y}_n - \mathbf{w}^\top \mathbf{x}_n|$ and $R(w) = ||\mathbf{w}||_1$. Therefore, the sub-gradients of this model will be given by:

$$\partial J(w) = \partial L(w) + \partial(\lambda R(w)) \tag{3}$$

Solving for $L(w)$ first, using affine transform rule of sub-differential calculus we get. Assume $t = \mathbf{y}_n - \mathbf{w}^\top \mathbf{x}_n$

$$\partial L(w) = -\mathbf{x}_n \partial |t| \tag{4}$$

The following cases arise:

- **Case 1**: $\partial L(w) = -\mathbf{x}_n \times 1 = -\mathbf{x}_n$ if $t > 0$
- **Case 2**: $\partial L(w) = -\mathbf{x}_n \times -1 = \mathbf{x}_n$ if $t < 0$
- **Case 3**: $\partial L(w) = -\mathbf{x}_n \times c = -c\mathbf{x}_n$ where $c \in [-1, 1]$ if $t = 0$

Now for, the other half $\lambda R(w)$,

$$\lambda ||\mathbf{w}||_1 = \lambda \sum_{d=1}^{D} |w_d| = \lambda(|w_1| + |w_2| + ... + |w_D|) \tag{5}$$

$$\partial(\lambda ||\mathbf{w}||_1) = \lambda \partial(|w_1| + |w_2| + ... + |w_D|) \tag{6}$$

Again, we will have the following cases

- **Case 1**: $\partial(\lambda R(w)) = \lambda \times 1 = \lambda$ if $w_d > 0$
- **Case 2**: $\partial(\lambda R(w)) = \lambda \times -1 = -\lambda$ if $w_d < 0$

- **Case 3**: $\partial(\lambda R(w)) = \lambda \times k = k\lambda$ where $k \in [-1, 1]$ if $w_d = 0$

Substituting all the above cases of $\partial L(w)$ and $\partial(\lambda R(w))$ and plugging it into (3), we will get the gradients/sub-gradients of this model.

Introduction to ML (CS771), Autumn 2020
Indian Institute of Technology Kanpur
Homework Assignment Number 1

Student Name: Debanjan Chatterjee
Roll Number: 20111016
Date: October 30, 2020

QUESTION 2

Squared loss function with a feature masking is expressed as

$$\sum_{n=1}^{N}(\mathbf{y}_n - \mathbf{w}^\top \bar{\mathbf{x}}_n)^2 \tag{7}$$

where,

$$\bar{\mathbf{x}}_n = \mathbf{x}_n \circ \mathbf{m}_n \tag{8}$$

Here $\mathbf{m}_n$ is a random vector, where each element is a Bernoulli random variable, according to given problem statement. Hence, expectation of each term in $\mathbf{m}_n$ (Bernoulli random variable) is p. Therefore we get,

$$\mathbb{E}(\bar{\mathbf{x}}_n) = p\mathbf{x}_n \tag{9}$$

Now expected value of the new loss function (1):

$$\mathbb{E}(\sum_{n=1}^{N}(\mathbf{y}_n - \mathbf{w}^\top \bar{\mathbf{x}}_n)^\top (\mathbf{y}_n - \mathbf{w}^\top \bar{\mathbf{x}}_n)) \tag{10}$$

$$\sum_{n=1}^{N}\mathbb{E}(\mathbf{y}_n^\top \mathbf{y}_n) - \mathbb{E}(\bar{\mathbf{x}}_n^\top \mathbf{w}\mathbf{y}_n) - \mathbb{E}(\mathbf{w}^\top \bar{\mathbf{x}}_n \mathbf{y}_n^\top) + \mathbb{E}(\mathbf{w}^\top \bar{\mathbf{x}}_n \bar{\mathbf{x}}_n^\top \mathbf{w}) \tag{11}$$

Using (9) and the formulas $\text{Cov}[X,Y] = \mathbb{E}(XY^\top) - \mathbb{E}(X)[\mathbb{E}(Y)]^\top$ and $\text{Cov}[X,X] = \text{Cov}[X]$ in (11) we get:

$$\sum_{n=1}^{N}\mathbf{y}_n^\top \mathbf{y}_n - p\mathbf{x}_n^\top \mathbf{w}\mathbf{y}_n - \mathbf{w}^\top p\mathbf{x}_n\mathbf{y}_n^\top + \mathbf{w}^\top(\text{Cov}[\bar{\mathbf{x}}_n] + (p\mathbf{x}_n^\top)(p\mathbf{x}_n))\mathbf{w} \tag{12}$$

$$\sum_{n=1}^{N}\mathbf{y}_n^\top \mathbf{y}_n - p\mathbf{x}_n^\top \mathbf{w}\mathbf{y}_n - \mathbf{w}^\top p\mathbf{x}_n\mathbf{y}_n^\top + \mathbf{w}^\top p^2\mathbf{x}_n^\top \mathbf{x}_n\mathbf{w} + \text{Cov}[\bar{\mathbf{x}}_n]\mathbf{w}^\top \mathbf{w} \tag{13}$$

Now $\text{Cov}[\bar{\mathbf{x}}_n]$ gives a scalar value so replacing it with λ , and simplifying (7) we get

$$\sum_{n=1}^{N}(\mathbf{y}_n - p\mathbf{w}^\top \mathbf{x}_n)^\top(\mathbf{y}_n - p\mathbf{w}^\top \mathbf{x}_n) + \lambda\mathbf{w}^\top \mathbf{w} \tag{14}$$

$$\sum_{n=1}^{N}(\mathbf{y}_n - p\mathbf{w}^\top \mathbf{x}_n)^2 + \lambda\mathbf{w}^\top \mathbf{w} \tag{15}$$

Equation (15) clearly represents a new regularized least square objective function: $L_{reg}(w) = L(w) + \lambda R(w)$. Thus optimal $\mathbf{w}(\hat{\mathbf{w}})$ is given by:

$$\hat{\mathbf{w}} = \arg\min_{w}\sum_{n=1}^{N}(\mathbf{y}_n - p\mathbf{w}^\top \mathbf{x}_n)^2 + \lambda\mathbf{w}^\top \mathbf{w} \tag{16}$$

Introduction to ML (CS771), Autumn 2020
Indian Institute of Technology Kanpur
Homework Assignment Number 1

Student Name: Debanjan Chatterjee
Roll Number: 20111016
Date: October 30, 2020

QUESTION 3

Given loss function

$$\mathcal{L}(\mathbf{B},\mathbf{S}) = TRACE[(\mathbf{Y}-\mathbf{XBS})^\top(\mathbf{Y}-\mathbf{XBS})] \tag{17}$$

$$\mathcal{L}(\mathbf{B},\mathbf{S}) = TRACE[\mathbf{Y}^\top\mathbf{Y}-\mathbf{Y}^\top\mathbf{XBS}-\mathbf{S}^\top\mathbf{B}^\top\mathbf{X}^\top\mathbf{Y}+\mathbf{S}^\top\mathbf{B}^\top\mathbf{X}^\top\mathbf{YXBS}] \tag{18}$$

Deriving the ALT-OPT algorithm for the problem:

Step 1: Pre-compute the matrix operation $\mathbf{G} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y}$, the reason for performing this pre computation will become clear in the Step 4, when we find the upgrade expression of $\mathbf{S}$

Step 2: Initialize $B = B^{(t)}$, $t = 0$

Step 3: Solve $S^{(t+1)} = \arg\min\limits_{\mathbf{S}}\mathcal{L}(\mathbf{B}^{(t)},\mathbf{S})$, $\mathbf{B}$ is fixed at its most recent value i.e. $\mathbf{B}^{(t)}$

Therefore, to get $\mathbf{S}^{(t+1)}$ we need to solve for $\frac{\partial\mathcal{L}}{\partial S} = 0$, keeping $\mathbf{B}^{(t)}$ as constant using First-Order Optimality.

Using (18) , we get,

$$\frac{\partial\mathcal{L}}{\partial S} = 0-(\mathbf{Y}^\top\mathbf{XB}^{(t)})^\top-\mathbf{B}^{(t)\top}\mathbf{X}^\top\mathbf{Y}+\mathbf{B}^{(t)\top}\mathbf{X}^\top\mathbf{XB}^{(t)}\mathbf{S}^{(t+1)}+(\mathbf{S}^{(t+1)\top}\mathbf{B}^{(t)\top}\mathbf{X}^\top\mathbf{XB}^{(t)})^\top = 0 \tag{19}$$

Simplifying (19) and solving for $\mathbf{S}^{(t+1)}$ we get,

$$\mathbf{S}^{(t+1)} = (\mathbf{B}^{(t)\top}\mathbf{X}^\top\mathbf{XB}^{(t)})^{-1}(\mathbf{B}^{(t)\top}\mathbf{X}^\top)\mathbf{Y} \tag{20}$$

$$\mathbf{S}^{(t+1)} = ((\mathbf{XB}^{(t)})^\top\mathbf{XB}^{(t)})^{-1}(\mathbf{XB}^{(t)})^\top\mathbf{Y} \tag{21}$$

Step 4: Solve $B^{(t+1)} = \arg\min\limits_{\mathbf{B}}\mathcal{L}(\mathbf{B},\mathbf{S}^{(t+1)})$, $\mathbf{S}$ is fixed at its most recent value i.e. $\mathbf{S}^{(t+1)}$

Therefore, to get $\mathbf{B}^{(t+1)}$ we need to solve for $\frac{\partial\mathcal{L}}{\partial B} = 0$, keeping $\mathbf{S}^{(t+1)}$ as constant using First-Order Optimality.

Using (18) , we get,

$$\frac{\partial\mathcal{L}}{\partial B} = 0-(\mathbf{Y}^\top\mathbf{X})^\top\mathbf{S}^{(t+1)}-(\mathbf{S}^{(t+1)}\mathbf{Y}^\top\mathbf{X})^\top+(\mathbf{S}^{(t+1)}\mathbf{S}^{(t+1)\top}\mathbf{B}^{(t+1)\top}\mathbf{X}^\top X)^\top+(\mathbf{S}^{(t+1)\top}\mathbf{B}^{(t+1)\top}\mathbf{X}^\top\mathbf{X})^\top\mathbf{S}^{(t+1)\top} = 0 \tag{22}$$

Simplifying (22) and solving for $\mathbf{B}^{(t+1)}$ we get,

$$\mathbf{B}^{(t+1)} = (\mathbf{X}^\top\mathbf{X})^{-1}(\mathbf{X}^\top\mathbf{YS}^{(t+1)\top})(\mathbf{S}^{(t+1)}\mathbf{S}^{(t+1)\top})^{-1} \tag{23}$$

Let us assume $\mathbf{G} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y}$. By using the associative rule of matrix multiplication, we can write (23) as $\mathbf{B}^{(t+1)} = \mathbf{G}\mathbf{S}^{(t+1)\top}(\mathbf{S}^{(t+1)}\mathbf{S}^{(t+1)\top})^{-1}$. Now we can use the pre-computed value of $\mathbf{G}$ from Step 1 and plug it in (23), thus simplifying the computation each iteration.

Step 5: $t = t + 1$. Go to Step 3 if not converged yet.

While solving both the sub-problems (solving for $\mathbf{B}$ and $\mathbf{S}$) we observe, that in the upgrade expressions (21) and (23), for $\mathbf{B}$, computation of $(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y}$ has already been completed in Step 1, and need not be repeated for every iteration, thus for $\mathbf{B}$, 3 matrix multiplications and 1 matrix inversion is needed each iteration. Whereas, for $\mathbf{S}$, 4 matrix multiplications and 1 matrix inversions are needed each iteration. Thus, its relatively easier to solve for $\mathbf{B}$ than $\mathbf{S}$, howver the difference is very slight. One additional comment, would be that since both the update expressions contain matrix inversions, as matrix inversion can be a very expensive (slow) computation for large matrices, an alternate approach might be to use some iterative optimization technique like gradient descent to solve the sub-problems (will be faster).

Introduction to ML (CS771), Autumn 2020
Indian Institute of Technology Kanpur
Homework Assignment Number 1

Student Name: Debanjan Chatterjee
Roll Number: 20111016
Date: October 30, 2020

QUESTION 4

Given loss function:

$$\hat{\mathbf{w}} = \arg\min_{w}(\frac{1}{2}\sum_{n=1}^{N}(\mathbf{y}_n - \mathbf{w}^\top\mathbf{x}_n) + \frac{\lambda}{2}\mathbf{w}^\top\mathbf{w} \tag{24}$$

Now, for the Newton's method we would need to find the gradient $\mathbf{g}$ and Hessian $\mathbf{H}$ of $\hat{\mathbf{w}}$

$$\mathbf{g} = \frac{\partial\hat{\mathbf{w}}}{\partial w} = 2\frac{\sum_{n=1}^{N}(-\mathbf{x}_n)(\mathbf{y}_n - \mathbf{x}_n^\top\mathbf{w})}{2} + 2\frac{\lambda\mathbf{w}}{2} \tag{25}$$

$$\mathbf{g} = \sum_{n=1}^{N}(-\mathbf{x}_n)(\mathbf{y}_n - \mathbf{x}_n^\top\mathbf{w}) + \lambda\mathbf{w} \tag{26}$$

$$\mathbf{H} = \frac{\partial\mathbf{g}}{\partial w} = \sum_{n=1}^{N}\mathbf{x}_n\mathbf{x}_n^\top + \lambda\mathbf{I}_D \tag{27}$$

Now in Newton's method weight up gradation can be expressed as:

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + (\mathbf{H}^{(t)})^{-1}\mathbf{g}^t \tag{28}$$

Using (26) and (27):

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - (\sum_{n=1}^{N}\mathbf{x}_n\mathbf{x}_n^\top + \lambda\mathbf{I}_D)^{-1}(\sum_{n=1}^{N}(-\mathbf{x}_n)(\mathbf{y}_n - \mathbf{x}_n^\top\mathbf{w}^{(t)}) + \lambda\mathbf{w}^t) \tag{29}$$

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + (\sum_{n=1}^{N}\mathbf{x}_n\mathbf{x}_n^\top + \lambda\mathbf{I}_D)^{-1}\sum_{n=1}^{N}(\mathbf{x}_n\mathbf{y}_n - \mathbf{x}_n\mathbf{x}_n^\top\mathbf{w}^{(t)} - \lambda\mathbf{w}^{(t)}) \tag{30}$$

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + (\sum_{n=1}^{N}\mathbf{x}_n\mathbf{x}_n^\top + \lambda\mathbf{I}_D)^{-1}\sum_{n=1}^{N}(\mathbf{x}_n\mathbf{y}_n) - (\sum_{n=1}^{N}\mathbf{x}_n\mathbf{x}_n^\top + \lambda\mathbf{I}_D)^{-1}\sum_{n=1}^{N}(\mathbf{x}_n\mathbf{x}_n^\top - \lambda\mathbf{I}_D)\mathbf{w}^{(t)} \tag{31}$$

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_D)^{-1}\mathbf{X}^\top\mathbf{y} - \mathbf{w}^{(t)} \tag{32}$$

$$\mathbf{w}^{(t+1)} = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_D)^{-1}\mathbf{X}^\top\mathbf{y} \tag{33}$$

Since $\mathbf{w}^{(t+1)}$ expression is independent of $\mathbf{w}^{(t)}$,) Newton's method for the problem will only take one iteration to converge (we have a closed form solution).

Introduction to ML (CS771), Autumn 2020
Indian Institute of Technology Kanpur
Homework Assignment Number 1

Student Name: Debanjan Chatterjee
Roll Number: 20111016
Date: October 30, 2020

QUESTION 5

For the given dice roll problem , **Multinomial** will be used for the likelihood. According to th given problem statement, the likelihood is:

$$P(N|\pi) = \frac{(N)!}{\prod_{i=1}^{6} N_i!} \prod_{i=1}^{6} \pi_i^{N_i} \tag{34}$$

Dirichlet will be used for prior and it is as follow:

$$P(\pi|\alpha) = \frac{1}{Beta(\alpha)} \prod_{i=1}^{6} \pi_i^{\alpha_i - 1} \tag{35}$$

where,

$$Beta(\alpha) = \frac{\prod_{i=1}^{6} \Gamma(\alpha_i)}{\Gamma(\prod_{i=1}^{6} \alpha_i)} \tag{36}$$

Hence, to compute the MAP solution, the Log Likelihood will be:

$$LP(\pi) = \arg\max_{\pi}((\log P(N|\pi)) + \log(P(\pi)|\alpha)) \text{ such that } \sum_{i=1}^{6} \pi_i = 1 \tag{37}$$

$$LP(\pi) = \arg\max_{\pi}(\sum_{i=1}^{6} N_i \log \pi_i + \sum_{i=1}^{6}(\alpha_i - 1)\log \pi_i) \text{ such that } \sum_{i=1}^{6} \pi_i = 1 \tag{38}$$

(The terms independent of π are excluded)
Converting (38) into an unconstrained problem (39) where λ is Lagrangian multiplier.

$$LP(\pi) = \arg\max_{\pi}(\sum_{i=1}^{6} N_i \log \pi_i + \sum_{i=1}^{6}(\alpha_i - 1)\log \pi_i + \lambda(1 - \sum_{i=1}^{6} \pi_i)) \tag{39}$$

Therefore, by equating $\frac{\partial LP(\pi)}{\partial \pi_m} = 0$, we can get π_{MAP} :

$$\frac{\partial LP(\pi)}{\partial \pi_m} = \frac{\partial}{\partial \pi_m}(\sum_{i=1}^{6} N_i \log \pi_i + \sum_{i=1}^{6}(\alpha_i - 1)\log \pi_i + \lambda(1 - \sum_{i=1}^{6} \pi_i)) = 0 \tag{40}$$

Solving (40) we get:

$$\frac{(N_m + \alpha_m - 1)}{\pi_m} - \lambda = 0 \tag{41}$$

Therefore,

$$\pi_{MAP} = \pi_m = \frac{(N_m + \alpha_m - 1)}{\lambda} \tag{42}$$

Now, we need to find the value of λ (Lagrange multiplier).
Using $\sum_{i=1}^{6} \pi_i = 1$ and (42), we get:

$$\frac{(N_1+\alpha_1-1)}{\lambda}+\frac{(N_1+\alpha_2-1)}{\lambda}+\frac{(N_3+\alpha_3-1)}{\lambda}+\frac{(N_4+\alpha_4-1)}{\lambda}+\frac{(N_5+\alpha_5-1)}{\lambda}+\frac{(N_6+\alpha_6-1)}{\lambda}=1 \tag{43}$$

$$\lambda = (N_1+N_2+N_3+N_4+N_5+N_6)+(\alpha_1+\alpha_2+\alpha_3+\alpha_4+\alpha_5+\alpha_6)-6 \tag{44}$$

Now we know $N_1+N_2+N_3+N_4+N_5+N_6 = N$ Therefore,

$$\lambda = N+(\alpha_1+\alpha_2+\alpha_3+\alpha_4+\alpha_5+\alpha_6)-6 \tag{45}$$

Now substituting (45) in (42) , we get:

When N (the number of times the dice is rolled) is very small, MAP solution will be preferred over MLE solution as MLE solution may tend to over-fit the data.

The Fully Bayesian Inference can be obtained by:

$$P(\pi|N) = \frac{P(\pi)P(N|\pi)}{P(N)} \tag{46}$$

$$P(\pi|N) = \frac{\frac{\Gamma(\prod_{i=1}^{6}\alpha_i)}{\prod_{i=1}^{6}\Gamma(\alpha_i)}\prod_{i=1}^{6}\pi_i^{\alpha_i-1}\frac{(N)!}{\prod_{i=1}^{6}N_i!}\prod_{i=1}^{6}\pi_i^{N_i}}{\int\frac{\Gamma(\prod_{i=1}^{6}\alpha_i)}{\prod_{i=1}^{6}\Gamma(\alpha_i)}\prod_{i=1}^{6}\pi_i^{\alpha_i-1}\frac{(N)!}{\prod_{i=1}^{6}N_i!}\prod_{i=1}^{6}\pi_i^{N_i}} \tag{47}$$

$$P(\pi|N) = \frac{\frac{\Gamma(\prod_{i=1}^{6}\alpha_i)}{\prod_{i=1}^{6}\Gamma(\alpha_i)}\frac{(N)!}{\prod_{i=1}^{6}N_i!}\prod_{i=1}^{6}\pi_i^{\alpha_i+N_i-1}}{\int\frac{\Gamma(\prod_{i=1}^{6}\alpha_i)}{\prod_{i=1}^{6}\Gamma(\alpha_i)}\frac{(N)!}{\prod_{i=1}^{6}N_i!}\prod_{i=1}^{6}\pi_i^{\alpha_i+N_i-1}} \tag{48}$$

Now, the likelihood (Multinoulli) and the prior (Dirichlet) are conjugate to each other, hence we can find the fully Bayesian inference analytically, Hence,

$$P(\pi|N) \propto \text{Dirichlet}(\pi|\alpha+N) \tag{49}$$

Yes, we can find MAP and MLE from Fully Bayesian Inference. The posterior obtained in Fully Bayesian Inference, is a Dirichlet Distribution with parameter: $\alpha+N$.The Mode of Dirichlet $(\pi|\alpha+N)$ will be the MAP estimate.

$$\text{Mode of Dirichlet}(\pi|\alpha) = \hat{\pi_k} = \frac{\alpha_k-1}{\sum_{i=1}^{6}(\alpha_k-1)} \tag{50}$$

Therefore,

$$\text{Mode of Dirichlet}(\pi|\alpha+N) = \hat{\pi_k} = \frac{N_k+\alpha_k-1}{N+\sum_{i=1}^{6}(\alpha_k-1)} \tag{51}$$

For MLE, by ignoring the prior in Fully Bayesian Inference we get:

$$P(\pi|N) \propto \prod_{i=1}^{6}\pi_i^{N_i} \tag{52}$$

Therefore, the MLE estimate will be as follows:

$$\hat{\pi_k} = \frac{N_k}{N} \tag{53}$$