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

Student Name: Jaskaran Kalra
Roll Number: 180322
Date: October 30, 2020

QUESTION 1

$$\mathbf{w}_{\text{opt}} = \arg\min_{\mathbf{w}} \sum_{n=1}^{N} |y_n - \mathbf{w}^T \mathbf{x}_n| + \lambda \|\mathbf{w}\|_1$$

1) Norm functions are convex. This can be proved by showing:

$$f(tx_1 + (1-t)x_2) \;\leq\; tf(x_1) + (1-t)f(x_2), \quad t \in [0,1]$$

We can easily show that the above condition satisfies for a norm function:

$\|t\mathbf{x}_1 + (1-t)\mathbf{x}_2\|_1 \leq \|t\mathbf{x}_1\|_1 + \|(1-t)\mathbf{x}_2\|_1 = t\|\mathbf{x}_1\|_1 + (1-t)\|\mathbf{x}_2\|_1$ (Triangle Inequality)

Clearly, the condition for convexity is satisfied for the norm function, hence $\|\mathbf{w}\|_1$ is convex.

2) $y_n - \mathbf{w}^T\mathbf{x}_n$ is convex because linear functions are convex. Also, $|y_n - \mathbf{w}^T\mathbf{x}_n|$ is convex because $g(x) = |x|$ is a convex function. This implies that $\sum_{n=1}^{N} |y_n - \mathbf{w}^T\mathbf{x}_n|$ is also convex, since sum of convex functions is a convex function.

$\implies$ From 1. and 2. , we can say that both the terms in the given objective function are convex,hence the given objective function is convex.

We know that the absolute function $|t|$ is non-differentiable at t = 0. Thus, clearly, the given objective function is non-differentiable only when :
i) $y_n - \mathbf{w}^T x_n = 0$ for each n $\in$ [N]
ii) $w_d = 0$ for each d $\in$ [D]

The sub-gradient of the objective function can be found as below:

$$L(\mathbf{w}) \;=\; \sum_{n=1}^{N} |y_n - \mathbf{w}^T\mathbf{x}_n| + \lambda\|\mathbf{w}\|_1$$

$$\frac{\partial L(\mathbf{w})}{\partial \mathbf{w}} \;=\; \sum_{n=1}^{N} c_n\mathbf{x}_n \;+\; \lambda\mathbf{v} \qquad \text{such that:}$$

$$c_n = \begin{cases} -1 & \text{if } y_n - \mathbf{w}^T\mathbf{x}_n > 0 \\ 1 & \text{if } y_n - \mathbf{w}^T\mathbf{x}_n < 0 \\ k, k \in [-1,1] & \text{if } y_n - \mathbf{w}^T\mathbf{x}_n = 0 \end{cases}$$

$\mathbf{v}$ is a D-dim vector s.t. for $i = 1, 2, .., D$,

$$v_i = \begin{cases} 1 & \text{if } w_i > 0 \\ -1 & \text{if } w_i < 0 \\ k', k' \in [-1,1] & \text{if } w_i = 0 \end{cases}$$

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

Student Name: Jaskaran Kalra
Roll Number: 180322
Date: October 30, 2020

QUESTION 2

The squared loss function $L_M(\mathbf{w})$ for regression using masked inputs, is given as follows:

$$\begin{aligned}
L_M(\mathbf{w}) &= \sum_{n=1}^{N}(y_n - \mathbf{w}^T\tilde{\mathbf{x}_n})^2 \\
&= \sum_{n=1}^{N}\left(y_n - \sum_{i=1}^{D} w_i\tilde{x_{ni}}\right)^2 \\
&= \sum_{n=1}^{N}\left(y_n^2 - 2y_n(\sum_{i=1}^{D} w_i\tilde{x_{ni}}) + \sum_{i=1}^{D}(w_i\tilde{x_{ni}})^2 + \sum_{1\le i,j\le D, i\neq j} w_iw_j\tilde{x_{ni}}\tilde{x_{nj}}\right)
\end{aligned}$$

We know that, $\tilde{x_{ni}}$ is a random variable with $p(x_{ni}=1)=p$ and $p(x_{ni}=0)=1-p$.

$$\implies \quad E[\tilde{x_{ni}}] = px_{ni} \quad \text{and} \quad E[\tilde{x_{ni}^2}] = px_{ni}^2$$

Calculating the expected value of $L_D(w)$:

$$\begin{aligned}
E[L_D(\mathbf{w})] &= E\left[\sum_{n=1}^{N}\left(y_n^2 - 2y_n(\sum_{i=1}^{D} w_i\tilde{x_{ni}}) + \sum_{i=1}^{D}(w_i\tilde{x_{ni}})^2 + 2\sum_{1\le i<j\le D} w_iw_j\tilde{x_{ni}}\tilde{x_{nj}}\right)\right] \\
&= \sum_{n=1}^{N} E\left[y_n^2 - 2y_n(\sum_{i=1}^{D} w_i\tilde{x_{ni}}) + \sum_{i=1}^{D}(w_i\tilde{x_{ni}})^2 + 2\sum_{1\le i<j\le D} w_iw_j\tilde{x_{ni}}\tilde{x_{nj}}\right] \\
&= \sum_{n=1}^{N}\left(y_n^2 - 2y_np(\sum_{i=1}^{D} w_ix_{ni}) + \sum_{i=1}^{D}p(w_ix_{ni})^2 + 2\sum_{1\le i<j\le D} p^2w_iw_jx_{ni}x_{nj}\right)
\end{aligned}$$

Define $\mathbf{w}_p^T = p\mathbf{w}^T$, and the above equation can be written as :

$$\begin{aligned}
E[L_D(\mathbf{w})] &= \sum_{n=1}^{N}\left((y_n - \sum_{i=1}^{D} pw_ix_{ni})^2 + \sum_{i=1}^{D}(p-p^2)(w_ix_{ni})^2\right) \\
E[L_D(\mathbf{w})] &= \sum_{n=1}^{N}\left((y_n - \mathbf{w}_p\mathbf{x}_n)^2 + \sum_{i=1}^{D}(1/p-1)(w_{pi}x_{ni})^2\right)
\end{aligned}$$

We can see that the above loss function is a regularized loss function, with the regularization term $\sum_{n=1}^{N}\sum_{i=1}^{D}\lambda(w_{pi}^Tx_{ni})^2$ where $\lambda = 1/p - 1$.

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

Student Name: Jaskaran Kalra
Roll Number: 180322
Date: October 30, 2020

QUESTION 3

We need to learn both $\mathbf{B}$ and $\mathbf{S}$ by solving the following problem:

$$\{\hat{\mathbf{B}}, \hat{\mathbf{S}}\} = \arg\min_{\mathbf{B},\mathbf{S}} \text{TRACE}[(\mathbf{Y} - \mathbf{XBS})^T(\mathbf{Y} - \mathbf{XBS})]$$

To solve the problem we'll take the following **alternative-optimization** approach:

1. Initialize $\mathbf{S} = \mathbf{S}^{[0]} \quad , t = 0$
2. Solving for $\mathbf{B}^{(t+1)}$:

$$\mathbf{B}^{(t+1)} = \arg\min_{\mathbf{B}} \text{TRACE}[(\mathbf{Y} - \mathbf{XBS}^{(t)})^T(\mathbf{Y} - \mathbf{XBS}^{(t)})]$$

For simplicity, writing $\mathbf{S}^{(t)}$ as $\mathbf{S}$.

To get the solution for $\mathbf{B}^{(t+1)}$, set $\frac{\partial}{\partial \mathbf{B}}\text{TRACE}[(\mathbf{Y} - \mathbf{XBS})^T(\mathbf{Y} - \mathbf{XBS}] = 0$

$$\frac{\partial}{\partial \mathbf{B}}\text{TRACE}[(\mathbf{Y} - \mathbf{XBS})^T(\mathbf{Y} - \mathbf{XBS})] = -2\mathbf{X}^T\mathbf{YS}^T + 2\mathbf{X}^T\mathbf{XBSS}^T = 0$$

$$\implies 2\mathbf{X}^T\mathbf{XBSS}^T = 2\mathbf{X}^T\mathbf{YS}^T$$

$$\implies \mathbf{B}^{(t+1)} = (\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{YS}^T)(\mathbf{SS}^T)^{-1} \tag{1}$$

3. Solving for $\mathbf{S}^{(t+1)}$:

$$\mathbf{S}^{(t+1)} = \arg\min_{\mathbf{B}} \text{TRACE}[(\mathbf{Y} - \mathbf{XB}^{(t+1)}\mathbf{S}^{(t)})^T(\mathbf{Y} - \mathbf{XB}^{(t+1)}\mathbf{S}^{(t)})]$$

For simplicity, writing $\mathbf{B}^{(t+1)}$ as $\mathbf{B}$.

To get the solution for $\mathbf{S}^{(t+1)}$, set $\frac{\partial}{\partial \mathbf{S}}\text{TRACE}[(\mathbf{Y} - \mathbf{XBS})^T(\mathbf{Y} - \mathbf{XBS}] = 0$

$$\frac{\partial}{\partial \mathbf{S}}\text{TRACE}[(\mathbf{Y} - \mathbf{XBS})^T(\mathbf{Y} - \mathbf{XBS})] = -2\mathbf{B}^T\mathbf{X}^T\mathbf{Y} + 2\mathbf{B}^T\mathbf{X}^T\mathbf{XBS} = 0$$

$$\implies 2\mathbf{B}^T\mathbf{X}^T\mathbf{XBS} = 2\mathbf{B}^T\mathbf{X}^T\mathbf{Y}$$

$$\implies \mathbf{S}^{(t+1)} = (\mathbf{B}^T\mathbf{X}^T\mathbf{XB})^{-1}(\mathbf{B}^T\mathbf{X}^T\mathbf{Y}) \tag{2}$$

4. t := t+1. Go to step 2 if not converged.

We can see from the closed form solutions of $\mathbf{S}$ and $\mathbf{B}$ that in $\mathbf{S}$, we'll have to calculate inverse of a matrix of size K x K, whereas in B, we'll have to calculate inverses of 2 matrices of size (K x K) and (D x D). Thus, in terms of computational complexity, it is clearly more "difficult" to compute B than S.

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

Student Name: Jaskaran Kalra
Roll Number: 180322
Date: October 30, 2020

QUESTION 4

$$L(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}(y_n - \mathbf{w}^T\mathbf{x}_n)^2 + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w}$$

$$\frac{\partial L(\mathbf{w})}{\partial w_i} = -\sum_{n=1}^{N}(y_n - \mathbf{w}^T\mathbf{x}_n)x_{ni} + \lambda w_i$$

$$\frac{\partial^2 L(\mathbf{w})}{\partial w_i w_j} = \sum_{n=1}^{N} x_{ni}x_{nj} \qquad , i \neq j$$

$$\sum_{n=1}^{N} x_{ni}^2 + \lambda \qquad , i = j$$

From the above equations we can write,

Gradient, $\mathbf{g} = -\mathbf{X}^T(\mathbf{y} - \mathbf{X}\mathbf{w}) + \lambda\mathbf{w}$

Hessian, $\mathbf{H} = \mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_d$

The update equation for Newton's method is,

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

Initializing $\mathbf{w}^{[0]}$ and substituting $\mathbf{H}$ and $\mathbf{g}$ in the above equation,

$$\mathbf{w}^{[1]} = \mathbf{w}^{[0]} - (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_d)^{-1}(-\mathbf{X}^T(\mathbf{y} - \mathbf{X}\mathbf{w}^{[0]}) + \lambda\mathbf{w}^{[0]})$$
$$\mathbf{w}^{[1]} = \mathbf{w}^{[0]} + (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_d)^{-1}(\mathbf{X}^T\mathbf{y} - (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_d)\mathbf{w}^{[0]})$$
$$\mathbf{w}^{[1]} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_d)^{-1}\mathbf{X}^T\mathbf{y} = \mathbf{w}_{opt}$$

We know that, $(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_d)^{-1}\mathbf{X}^T\mathbf{y}$ is the optimal solution for the given optimization problem. Thus, it takes only 1 iteration for the newton's method to converge.

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

Student Name: Jaskaran Kalra
Roll Number: 180322
Date: October 30, 2020

QUESTION 5

Let $\mathbf{y}_n$ (n = 1, 2,.., N) be a 6-dim vector s.t. $y_{ni} = 1$, if the die shows the number i on the n^{th} toss, and $y_{ni} = 0$ otherwise. (i = 1,2,.,6)

The total likelihood is given by:

$$\prod_{n=1}^{N} p(\mathbf{y}_n|\boldsymbol{\pi}) \;=\; \prod_{n=1}^{N}\prod_{i=1}^{6} \pi_i^{y_{ni}} \;=\; \prod_{i=1}^{6} \pi_i^{N_i}$$

We will assume a Dirichlet prior for the probability vector $\boldsymbol{\pi}$

$$p(\boldsymbol{\pi}) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{i=1}^{6} \pi_i^{\alpha_i - 1} \qquad \text{,where} B(\boldsymbol{\alpha}) = \frac{\prod_{i=1}^{6}\Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{6}\alpha_i)}$$

The MAP solution is given by:

$$\hat{\boldsymbol{\pi}}_{\text{MAP}} = \arg\max_{\boldsymbol{\pi}} \;\log(\prod_{n=1}^{N} p(\mathbf{y}_n|\boldsymbol{\pi})) \;+\; \log(p(\boldsymbol{\pi}))$$

By plugging in the expressions of the likelihood and prior from above and neglecting the constant terms, we get:

$$\hat{\boldsymbol{\pi}}_{\text{MAP}} = \arg\max_{\boldsymbol{\pi}} \sum_{i=1}^{6} log(\pi_i^{N_i+\alpha_i-1})$$

We have this constraint on the variables (π_1, π_2, .., π_6):

$$g(\pi_1, \pi_2, .., \pi_6) = \textstyle\sum_i \pi_i = 1$$

We introduce a Lagrange multiplier λ. Let the Lagrangian function be:

$$L(\pi_1, \pi_2, .., \pi_6, \lambda) \;=\; \sum_{i=1}^{6} log(\pi_i^{N_i+\alpha_i-1}) \;+\; \lambda(\pi_1 + \pi_2 + .. + \pi_6 - 1)$$

Taking gradient on both sides:

$$dL(\pi_1, \pi_2, .., \pi_6, \lambda) \;=\; \sum_{i=1}^{6} (\frac{N_i+\alpha_i-1}{\pi_i} + \lambda) d\pi_i \;+\; d\lambda(\pi_1 + \pi_2 + .. + \pi_6 - 1)$$

Solving for $dL = 0$ gives:

$$\pi_i = \frac{N_i+\alpha_i-1}{\sum_{j=1}^{6} N_j+\alpha_j-1} \text{ , which is the MAP solution.}$$

In cases where one or more than one number has not shown up on the dice yet, the MAP solution will be better than the MLE solution. In these cases the MLE solution will predict zero probability for the occurence of those numbers, while the MAP solution won't.

The full posterior distribution can be calculated from the given formula:

$$p(\boldsymbol{\pi}|\mathbf{y}) = \frac{p(\boldsymbol{\pi})p(\mathbf{y}|\boldsymbol{\pi})}{p(\mathbf{y})}$$

Computing $p(\mathbf{y})$ is hard in general. But since we are working with a conjugate pair of the prior and the likelihood, we don't need to compute this as we will see.

The denominator is constant w.r.t. $\boldsymbol{\pi}$, and the numerator, as we can easily see from the values derived in eqn. (1) and (2), is proportional to $\prod_{i=1}^{6} \boldsymbol{\pi}_i^{N_i+\alpha_i-1}$, which is nothing but the Dirichlet distribution, given by:

$$p(\mathbf{y}|\boldsymbol{\pi}) = \frac{1}{B(\boldsymbol{\alpha}')} \prod_{i=1}^{6} \pi_i^{N_i+\alpha_i-1} \qquad \text{,where} B(\boldsymbol{\alpha}') = \frac{\prod_{i=1}^{6} \Gamma(N_i+\alpha_i)}{\Gamma(\sum_{i=1}^{6} N_i+\alpha_i)}$$

Given this posterior, we can get the MLE and MAP estimate without solving the MLE and MAP optimization problems. To solve the MLE problem, we can assume a uniform prior by setting the α_is to be equal to 1 for all i = 1, 2, .., 6. After this find the mode of the resulting distribution. This would give us the MLE estimate. To get the MAP estimate, we can simply find the mode of the posterior distribution. That would give us the MAP solution.