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

QUESTION 1

Student Name: Musale Krushna Pavan
Roll Number: 20111268
Date: October 30, 2020

Given Absolute loss regression problem with ℓ_1 regularization

$$\boldsymbol{\omega}_{\mathbf{opt}} = \arg\min_{\boldsymbol{\omega}} \sum_{n=1}^{N} \left| y_n - \boldsymbol{\omega}^{\mathbf{T}} \mathbf{x_n} \right| + \lambda \left\| \boldsymbol{\omega} \right\|_1 \qquad \lambda > 0$$

we will use the following properties to prove its convexity:

1. Non-negative weighted sum of the convex functions are convex
2. All norms are convex functions
3. $|x|$ is convex function

- consider $\left| y_n - \boldsymbol{\omega}^T x_n \right|$ is convex using property (3)
  $\implies \sum_{n=1}^{N} \left| y_n - \boldsymbol{\omega}^T x_n \right|$ is also convex using property (1)
- also $\left\| \boldsymbol{\omega} \right\|_1$ is convex using property (2)
- $\arg\min_{\boldsymbol{\omega}} \sum_{n=1}^{N} \left| y_n - \boldsymbol{\omega}^{\mathbf{T}} \mathbf{x_n} \right| + \lambda \left\| \boldsymbol{\omega} \right\|_1 \qquad \lambda > 0$ is convex using property (1)

Hence convexity is proved.

Sub gradient vector:

$$\frac{\partial}{\partial \boldsymbol{\omega}} \sum_{n=1}^{N} \left| y_n - \boldsymbol{\omega}^{\mathbf{T}} \mathbf{x_n} \right| + \lambda \sum_{d=1}^{D} |\boldsymbol{\omega}_d|$$

$$\sum_{n=1}^{N} \frac{\partial}{\partial \boldsymbol{\omega}} \left| y_n - \boldsymbol{\omega}^{\mathbf{T}} \mathbf{x_n} \right| + \lambda \sum_{d=1}^{D} \frac{\partial}{\partial \boldsymbol{\omega}} |\boldsymbol{\omega}_d| \qquad \lambda > 0 \tag{1}$$

Consider $\frac{\partial}{\partial \boldsymbol{\omega}} \left| y_n - \boldsymbol{\omega}^{\mathbf{T}} \mathbf{x_n} \right|$

$$\frac{\partial}{\partial \boldsymbol{\omega}} \left| y_n - \boldsymbol{\omega}^{\mathbf{T}} \mathbf{x_n} \right| = \begin{cases} \mathbf{x_n} & if y_n - \boldsymbol{\omega}^{\mathbf{T}} \mathbf{x_n} > 0 \\ -\mathbf{x_n} & if y_n - \boldsymbol{\omega}^{\mathbf{T}} \mathbf{x_n} > 0 \\ C_1 \mathbf{x_n} & if y_n - \boldsymbol{\omega}^{\mathbf{T}} \mathbf{x_n} > 0; C_1 \in [-1, 1] \end{cases}$$

Consider $\frac{\partial}{\partial \boldsymbol{\omega}} |\boldsymbol{\omega}_{\mathbf{d}}|$

$$\frac{\partial}{\partial \boldsymbol{\omega}} |\boldsymbol{\omega}_{\mathbf{d}}| = \begin{cases} [0, 0, ...0, 1, 0, ...]^T & if \boldsymbol{\omega}_d > 0, 1 \text{ at position } d \\ [0, 0, ...0, -1, 0, ...]^T & if\ \boldsymbol{\omega}_d > 0, -1 \text{ at position } d \\ [0, 0, ...0, C_2, 0, ...]^T & if\ \boldsymbol{\omega}_d = 0, C_2 \text{ at position } d, C_2 \in [-1, 1] \end{cases}$$

The above 2 sub solutions can be substituted back in the equation (1) to get the sub-gradient vector.

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

QUESTION 2

Student Name: Musale Krushna Pavan
Roll Number: 20111268
Date: October 30, 2020

Linear regression model with minimizing the squared loss function;

$$\sum_{n=1}^{N}(y_n - \mathbf{w^T x_n})^2$$

lets mask the features of $\mathbf{x_n}$ by $\widetilde{\mathbf{x}}_\mathbf{n} = \mathbf{x_n} \circ \mathbf{m_n}$ with $m_{nd} \sim$ Bernoulli(p) New loss function $\sum_{n=1}^{N}(y_n - \mathbf{w^T}\widetilde{\mathbf{x}}_\mathbf{n})^2$
Calculating the Estimate value of new loss function:

$$\begin{aligned}
E(\sum_{n=1}^{N}(y_n - \mathbf{w^T}\widetilde{\mathbf{x}}_\mathbf{n})^2) &= \sum_{n=1}^{N} E[(y_n - \mathbf{w^T}\widetilde{\mathbf{x}}_\mathbf{n})^2] \\
&= \sum_{n=1}^{N} E[y_n^2 - 2y_n\mathbf{w^T}\widetilde{\mathbf{x}}_\mathbf{n} + (\mathbf{w^T}\widetilde{\mathbf{x}}_\mathbf{n})^2] \\
&= \sum_{n=1} y_n^2 - 2y_n E[\mathbf{w^T}\widetilde{\mathbf{x}}_\mathbf{n}] + E[(\mathbf{w^T}\widetilde{\mathbf{x}}_\mathbf{n})^2] \\
&= \sum_{n=1} y_n^2 - 2y_n p\mathbf{w^T x_n} + E[(\mathbf{w^T}\widetilde{\mathbf{x}}_\mathbf{n})^2] \\
&= \sum_{n=1}[(y_n - p\mathbf{w^T x_n})^2] - \sum_{n=1}^{D}\{(p\mathbf{w^T x_n})^2 + \sum_{i=1}^{D} E[(w_i\widetilde{x}_{ni})^2] + \sum_{\substack{i\neq j\\ i,j\in[1,D]}} E[w_i x_{ni} w_j x_{nj}]\} \\
&= \sum_{n=1}[(y_n - p\mathbf{w^T x_n})^2] - \sum_{i=1}^{D}\{p^2(\sum_{i=1}^{D} w_i^2 x_{ni}^2 + \sum_{\substack{i\neq j\\ i,j\in[1,D]}} w_i x_{ni} w_j x x_{nj}) \\
&\qquad + p\sum_{i=1}^{D} w_i^2 x_{ni}^2 + p^2 \sum_{\substack{i\neq j\\ i,j\in[1,D]}} w_i x_{ni} w_j x x_{nj}\} \\
&= \sum_{n=1}(y_n - p\mathbf{w^T x_n})^2 + pq\sum_{n=1}^{N}\sum_{i=1}^{D} w_i^2 x_{ni}^2 \\
&= \sum_{n=1}(y_n - p\mathbf{w^T x_n})^2 + \sum_{i=1}^{D} w_i^2 C_i \quad \text{where } C_i \text{ is const wrt } w
\end{aligned}$$

The above equation is in the form of ridge regression. so minimizing the expected value of our new loss function is equivalent to minimizing the ridge regression

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

QUESTION 3

Student Name: Musale Krushna Pavan
Roll Number: 20111268
Date: October 30, 2020

My solution to problem 3 Given:

$$\{\mathbf{B},\mathbf{S}\} = \underset{\mathbf{B},\mathbf{S}}{\arg\min}\ \mathrm{Tr}[(\mathbf{Y}-\mathbf{XBS})^{\mathbf{T}}(\mathbf{Y}-\mathbf{XBS})]$$

$$\{\mathbf{B},\mathbf{S}\} = \underset{\mathbf{B},\mathbf{S}}{\arg\min}\ \mathrm{Tr}[\mathbf{Y^TY}-\mathbf{Y^TXBS}-\mathbf{S^TB^TX^TY}+\mathbf{S^TB^TX^TXBS}]$$

Using ALT-OPT method initializing $\mathbf{B}=\mathbf{B}^{(0)}$ and $t=0$

$$\mathbf{S}^{(t+1)} = \underset{\mathbf{S}}{\arg\min}\ \ \mathcal{L}(\mathbf{B}^{(t)},\mathbf{S})$$

Derivating with respect to $\mathbf{S}$ and equating to 0:

$$\frac{\partial}{\partial\mathbf{S}}\ \mathrm{Tr}[\mathbf{Y^TY}-\mathbf{Y^TXB^{(t)}S}-\mathbf{S^TB^{(t)}}^{\mathbf{T}}\mathbf{X^TY}+\mathbf{S^TB^{(t)}}^{\mathbf{T}}\mathbf{X^TXB^{(t)}S}]=0$$

$$\implies \mathbf{0}-\mathbf{B^{(t)}}^{\mathbf{T}}\mathbf{X^TY}-\mathbf{B^{(t)}}^{\mathbf{T}}\mathbf{X^TY}+(\mathbf{B^{(t)}}^{\mathbf{T}}\mathbf{X^TXB^{(t)}}+\mathbf{B^{(t)}}^{\mathbf{T}}\mathbf{X^TXB^{(t)}})\mathbf{S}=0$$

$$\implies (\mathbf{B^{(t)}}^{\mathbf{T}}\mathbf{X^TXB^{(t)}})\mathbf{S}=\mathbf{B^{(t)}}^{\mathbf{T}}\mathbf{X^TY}$$

$$\implies \mathbf{S}=(\mathbf{B^{(t)}}^{\mathbf{T}}\mathbf{X^TXB^{(t)}})^{-\mathbf{1}}\mathbf{B^{(t)}}^{\mathbf{T}}\mathbf{X^TY}$$

Therefore $\mathbf{S^{(t+1)}}=(\mathbf{B^{(t)}}^{\mathbf{T}}\mathbf{X^TXB^{(t)}})^{-\mathbf{1}}\mathbf{B^{(t)}}^{\mathbf{T}}\mathbf{X^TY}$
Now next step in alt opt algorithm we update

$$\mathbf{B}^{(t+1)} = \underset{\mathbf{B}}{\arg\min}\ \ \mathcal{L}(\mathbf{B},\mathbf{S^{(t+1)}})$$

Derivating with respect to $\mathbf{B}$ and equating to 0:

$$\frac{\partial}{\partial\mathbf{B}}\ \mathrm{Tr}[\mathbf{Y^TY}-\mathbf{Y^TXBS^{(t+1)}}-\mathbf{S^{(1)}}^{\mathbf{T}}\mathbf{B^TX^TY}+\mathbf{S^{(t+1)}}^{\mathbf{T}}\mathbf{B^TX^TXBS^{(t+1)}}]=0$$

$$\implies \mathbf{0}-\mathbf{X^TYS^{(t+1)}}^{\mathbf{T}}-\mathbf{X^TYS^{(t+1)}}^{\mathbf{T}}+\mathbf{X^TXBS^{(t+1)}S^{(t+1)}}^{\mathbf{T}}+\mathbf{X^TXBS^{(t+1)}S^{(t+1)}}^{\mathbf{T}}=0$$

$$\implies \mathbf{X^TXBS^{(t+1)}S^{(1)}}^{\mathbf{T}}=\mathbf{X^TYS^{(t+1)}}^{\mathbf{T}}$$

$$\mathbf{B}=(\mathbf{X^TX})^{-\mathbf{1}}\mathbf{X^TYS^{(t+1)}}^{\mathbf{T}}(\mathbf{S^{(t+1)}S^{(t+1)}}^{\mathbf{T}})^{-\mathbf{1}}$$

Therefore $\mathbf{B^{(t+1)}}=(\mathbf{X^TX})^{-\mathbf{1}}\mathbf{X^TYS^{(t+1)}}^{\mathbf{T}}(\mathbf{S^{(t+1)}S^{(t+1)}}^{\mathbf{T}})^{-\mathbf{1}}$

We can observe that while computing $\mathbf{B^{(1)}}$ we required 2 inverse terms to compute so the sub problem of computing $\mathbf{B}$ is harder than the sub-problem of $\mathbf{S}$ which requires only one inverse term to compute.

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

Student Name: Musale Krushna Pavan
Roll Number: 20111268
Date: October 30, 2020

QUESTION 4

My solution to problem 4 Ridge Regression using Newton's Method

$$\boldsymbol{\omega}_{opt} = \arg\min_{\boldsymbol{\omega}} \frac{1}{2}(\mathbf{y} - \mathbf{X}\boldsymbol{\omega})^{\mathbf{T}}(\mathbf{y} - \mathbf{X}\boldsymbol{\omega}) + \frac{\lambda}{2}\boldsymbol{\omega}^{\mathbf{T}}\boldsymbol{\omega}$$

Considering the loss function:

$$\mathbf{L}(\boldsymbol{\omega}) = \frac{1}{2}(\mathbf{y}^{\mathbf{T}}\mathbf{y} - \mathbf{y}^{\mathbf{T}}\mathbf{X}\boldsymbol{\omega} - \boldsymbol{\omega}^{\mathbf{T}}\mathbf{X}^{\mathbf{T}}\mathbf{y} + \boldsymbol{\omega}^{\mathbf{T}}\mathbf{X}^{\mathbf{T}}\mathbf{X}\boldsymbol{\omega}) + \frac{\lambda}{2}\boldsymbol{\omega}^{\mathbf{T}}\boldsymbol{\omega}$$

considering the gradient of loss function:

$$\triangledown\mathbf{L}(\boldsymbol{\omega}) = \frac{1}{2}(\mathbf{0} - \mathbf{X}^{\mathbf{T}}\mathbf{y} - \mathbf{X}^{\mathbf{T}}\mathbf{y} + \mathbf{2}\mathbf{X}^{\mathbf{T}}\mathbf{X}\boldsymbol{\omega}) + \frac{\lambda}{2}\mathbf{2}\boldsymbol{\omega}$$

$$\triangledown\mathbf{L}(\boldsymbol{\omega}) = \mathbf{X}^{\mathbf{T}}\mathbf{X}\boldsymbol{\omega} - \mathbf{X}^{\mathbf{T}}\mathbf{y} + \lambda\boldsymbol{\omega}$$

$$\triangledown\mathbf{L}(\boldsymbol{\omega}) = (\mathbf{X}^{\mathbf{T}}\mathbf{X} + \lambda\mathbf{I_D})\boldsymbol{\omega} - \mathbf{X}^{\mathbf{T}}\mathbf{y}$$

considering the hessian of loss function:

$$\triangledown^2\mathbf{L}(\boldsymbol{\omega}) = \mathbf{X}^{\mathbf{T}}\mathbf{X} + \lambda\mathbf{I_D}$$

Now using the newtons method

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

subsitituting the values in newtons formula:

$$\boldsymbol{\omega}^{(t+1)} = \boldsymbol{\omega}^{(t)} - (\mathbf{X}^{\mathbf{T}}\mathbf{X} + \lambda\mathbf{I_D})^{-\mathbf{1}}((\mathbf{X}^{\mathbf{T}}\mathbf{X} + \lambda\mathbf{I_D})\boldsymbol{\omega}^{(\mathbf{t})} - \mathbf{X}^{\mathbf{T}}\mathbf{y})$$

$$\boldsymbol{\omega}^{(t+1)} = (\mathbf{X}^{\mathbf{T}}\mathbf{X} + \lambda\mathbf{I_D})^{-\mathbf{1}}\mathbf{X}^{\mathbf{T}}\mathbf{y}$$

As the $\boldsymbol{\omega}^{t+1}$ is independent of $\boldsymbol{\omega}$ term. The loss functions gradient and hassian becomes 0 in the next iteration.
So we need only **two iteration** to converge.

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

Student Name: Musale Krushna Pavan
Roll Number: 20111268
Date: October 30, 2020

QUESTION 5

Given a six faced dice rolled N The number of times each face appeared is $N_1, N_2, ..., N_6$
The probability of each face π_k $k \in (1,2,3,4,5,6)$ $\pi_k \in (0,1)$
The likelihood probability mass function for probability vector $\boldsymbol{\pi} = [\pi_1, \pi_2, ..., \pi_6]$ is Multinolli distribution.

$$P(\mathbf{y}|\pi) = \prod_{n=1}^{N} \prod_{i=1}^{6} \pi_i^{\mathbb{I}[y_n == i]} = \prod_{i=1}^{6} \pi_i^{N_i}$$

where $\mathbb{I}[y_n == i]$ is function return 1 if $y_n == i$ else 0 and $\sum_{i=1}^{6} \pi_i = 1$ Now the prior for the probability vector $\boldsymbol{\pi}$ is Dirchlet distribution:

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

Now for MAP solution:

$$\arg\min_{\boldsymbol{\pi}} \prod_{i=1}^{6} \pi_i^{N_i} \pi_i^{\alpha_i - 1} = \arg\min_{\boldsymbol{\pi}} \prod_{i=1}^{6} \pi_i^{N_i + \alpha_i - 1} \qquad \text{where } \sum_{i=1}^{6} \pi_i = 1$$

As it is constrained optimization we use lagranges method:

$$L(\boldsymbol{\pi}, K) = \prod_{i=1}^{6} (N_i + \alpha_i - 1) \log \pi_i + K(\sum_{i=1}^{6} \pi_i - 1)$$

Taking the derivative with respect to each π_i and K, and setting them to zero

$$\pi_i = \frac{N_i + \alpha_i - 1}{K}$$

$$K = N + \sum_{i=1}^{6} \alpha_i - 6 \qquad \text{using } \sum_{i=1}^{6} \pi_i = 1$$

MAP is given as :

$$\pi_i = \frac{N_i + \alpha_i - 1}{N + \sum_{i=1}^{6} \alpha_i - 6}$$

MAP solution will be better than MLE when there are less number of trails i.e when N is small.

Now calculating the full Bayesian posterior

$$
\begin{aligned}
P(\boldsymbol{\pi}|\mathbf{y}) &= \frac{P(\boldsymbol{\pi}) * P(\mathbf{y}|\boldsymbol{\pi})}{P(\boldsymbol{\pi}|\mathbf{y})} \\
&\propto \qquad \frac{1}{\mathbf{B}(\boldsymbol{\alpha})} \prod_{i=1}^{6} \pi_i^{\alpha_i - 1} \prod_{i=1}^{6} \pi_i^{N_i} \qquad \text{where denominator is constant wrt} \boldsymbol{\pi} \\
&\propto \qquad \prod_{i=1}^{6} \pi_i^{N_i + \alpha_i - 1} \\
&= \qquad \text{Dirichlet}(\boldsymbol{\pi}, N_1 + \alpha_1, ..., N_6 + \alpha_6)
\end{aligned}
$$

As the maximum value of posterior is at mode of Dirichlet distribution we can directly get MAP. i.e

$$\pi_i = \frac{N_i + \alpha_i - 1}{N + \sum_{i=1}^{6} \alpha_i - 6}$$

and we can also get MLE when our prior is uniform i.e when $\alpha_i = 1$

$$\pi_i = \frac{N_i}{N}$$