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

Student Name: Abhay
Roll Number: 180014
Date: October 30, 2020

QUESTION 1

According to question,

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

. Let us decompose objective function($\mathcal{L}(\mathbf{w})$) into $f(\mathbf{w}) = \sum_{n=1}^{N} |y_n - \mathbf{w}^T\mathbf{x}_n|$ and $g(\mathbf{w}) = \lambda||\mathbf{w}||_1 = \lambda\sum_{d=1}^{D} |w_d|$. As discussed in class, we know

1. sum of two convex function is convex
2. ax+b and $|x|$ are both convex.
3. f(x) is convex $\implies$ f(ax+b) is convex.

Now from property 2 and 3, we know that each term inside summation in $f(\mathbf{w})$ and $g(\mathbf{w})$ is convex($\lambda > 0$). Thus from property 1, we can say $f(\mathbf{w})$ and $g(\mathbf{w})$ are both convex. From the same property, we can say that our objective function which is essentially $f(\mathbf{w}) + g(\mathbf{w})$ is a convex function.

Consider $f_n(\mathbf{w}) = |y_n - \mathbf{w}^T\mathbf{x}_n|$. Here subscript n denotes the nth training example. Now as done in class using affine transformation rule, sub-gradient of $f_n(\mathbf{w})$ is

$$\partial f_n(\mathbf{w}) = \begin{cases} \mathbf{x}_n, & \text{if } y_n - \mathbf{w}^T\mathbf{x}_n < 0 \\ -\mathbf{x}_n, & \text{if } y_n - \mathbf{w}^T\mathbf{x}_n > 0 \\ -\mathbf{x}_n.c, & \text{if } y_n - \mathbf{w}^T\mathbf{x}_n = 0, \text{here } c \in [-1,1] \end{cases} \tag{1}$$

From the sum rule of sub-gradients we can say $\partial f(\mathbf{w}) = \sum_{n=1}^{N}(\partial f_n(\mathbf{w}))$.

Now sub-gradient of $g(\mathbf{w})$ will be a vector of dimension Dx1.The dth element of sub-gradient(say $\mathbf{p}$) is given by

$$\mathbf{p}_d = \begin{cases} 1, & \text{if } w_d > 0 \\ -1, & \text{if } w_d < 0 \\ c & \text{if } w_d = 0, \text{here } c \in [-1,1] \end{cases} \tag{2}$$

Using sum rule of sub-gradients on $f(\mathbf{w})$ and $g(\mathbf{w})$, the sub-gradient for objective function($\mathcal{L}$) can be written as

$$\partial\mathcal{L}(\mathbf{w}) = \partial f(\mathbf{w}) + \partial g(\mathbf{w}) = \sum_{n=1}^{N}(\partial f_n(\mathbf{w})) + \lambda\mathbf{p}$$

where $\partial f_n(\mathbf{w})$ and $\mathbf{p}$ are defined in eqn 1 and 2 resp.

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

Student Name: Abhay
Roll Number: 180014
Date: October 30, 2020

QUESTION 2

Original squared loss function is given by

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

According to question, after using dropout, our new loss function is

$$\mathcal{L}_d = \sum_{n=1}^{N} (y_n - \mathbf{w}^T \tilde{\mathbf{x}}_n)^2$$

where $\tilde{\mathbf{x}}_n = \mathbf{x}_n \circ \mathbf{m}_n$, $\circ$ denotes element wise product and $\mathbf{m}_n$ denotes $Dx1$ binary mask vector with $m_{nd} \sim$ Bernoulli(p). Now by definition of vector multiplication, we can write $\mathbf{w}^T \mathbf{x}_n$ can be written as $\sum_{d=1}^{D} w_d * \tilde{x}_{nd}$. Moreover, $\tilde{x}_{nd} = x_{nd} * m_{nd}$. Therefore above equation becomes According to question, after using dropout, our new loss function is

$$\mathcal{L}_d = \sum_{n=1}^{N} \left(y_n - \sum_{d=1}^{D} w_d * m_{nd} * x_{nd} \right)^2$$

For minimization, we will need the gradient of the function which will be $Dx1$ vector.Now, we the ith term of the gradient vector will be given by

$$\frac{\partial \mathcal{L}_d}{\partial w_i} = \sum_{n=1}^{N} \left(2 * (y_n - \sum_{d=1}^{D} w_d * m_{nd} * x_{nd}) * (-1 * m_{ni} * x_{ni}) \right)$$

$$= 2 * \sum_{n=1}^{N} \left(-y_n * m_{ni} * x_{ni} + \sum_{d=1}^{D} (w_d * m_{nd} * x_{nd} * m_{ni} * x_{ni}) \right)$$

$$= 2 * \sum_{n=1}^{N} \left(-y_n * m_{ni} * x_{ni} + w_i * m_{ni}^2 * x_{ni}^2 + \sum_{d=1, d \neq i}^{D} (w_d * m_{nd} * x_{nd} * m_{ni} * x_{ni}) \right)$$

Now since we minimise the expected value of function, using Linearity of expectation we have

$$E\left[\frac{\partial \mathcal{L}_d}{\partial w_i}\right] = 2 * \sum_{n=1}^{N} \left(E[-y_n * m_{ni} * x_{ni}] + E[w_i * m_{ni}^2 * x_{ni}^2] + \sum_{d=1, d \neq i}^{D} (E[w_d * m_{nd} * x_{nd} * m_{ni} * x_{ni}]) \right)$$

From the results of Bernoulli we know $E[m_{ni}]$=p and $E[m_{ni}^2]$=p=$Var(m_{ni}) + E[m_{ni}^2]$. Also, $m_{ni} and m_{nj}$ are independent for all j $\neq$ i.

$$E\left[\frac{\partial \mathcal{L}_d}{\partial w_i}\right] = 2 * \sum_{n=1}^{N} \left(-y_n * p * x_{ni} + w_i * (Var(m_{ni}) + p^2) * x_{ni}^2 + \sum_{d=1, d \neq i}^{D} (w_d * p * x_{nd} * p * x_{ni}) \right)$$

$$= 2 * \sum_{n=1}^{N} \left(-y_n * p * x_{ni} + w_i * Var(m_{ni}) * x_{ni}^2 + w_i * p^2 * x_{ni}^2 + \sum_{d=1, d\neq i}^{D} (w_d * p * x_{nd} * p * x_{ni}) \right) \tag{3}$$

Now suppose in the normal squared loss function we use p***w** in place of **w**. The function is defined as

$$\mathcal{L}' = \sum_{n=1}^{N} (y_n - p * \mathbf{w}^T \mathbf{x}_n)^2 = \sum_{n=1}^{N} \left(y_n - \sum_{d=1}^{D} p * w_d * x_{nd} \right)^2$$

Now, the ith term of its gradient is given by

$$\frac{\partial \mathcal{L}'}{\partial w_i} = \sum_{n=1}^{N} \left(2 * (y_n - \sum_{d=1}^{D} p * w_d * x_{nd}) * (-1 * p * x_{ni}) \right)$$

$$\frac{\partial \mathcal{L}'}{\partial w_i} = 2 * \sum_{n=1}^{N} \left(-y_n * p * x_{ni} + \sum_{d=1}^{D} (w_d * p * x_{nd} * p * x_{ni}) \right)$$

$$\frac{\partial \mathcal{L}'}{\partial w_i} = 2 * \sum_{n=1}^{N} \left(-y_n * p * x_{ni} + w_i * p^2 * x_{ni}^2 + \sum_{d=1, d\neq i}^{D} (w_d * p * x_{nd} * p * x_{ni}) \right)$$

Substituting the value of $\frac{\partial \mathcal{L}}{\partial w_i}$ and $Var(m_{ni}) = p * (1 - p)$ in 3, we got

$$E\left[\frac{\partial \mathcal{L}_d}{\partial w_i}\right] = \frac{\partial \mathcal{L}'}{\partial w_i} + 2 * \sum_{n=1}^{N} (w_i * p * (1-p) * x_{ni}^2) \tag{4}$$

From eqn 4 it is clear that dropp out is equivalent to minimizing the regularised function. the correponding regularised loss function can be written as

$$\mathcal{L}_{new} = \sum_{n=1}^{N} \left((y_n - p * \mathbf{w}^T \mathbf{x}_n)^2 + p * (1-p) * \sum_{d=1}^{D} w_d^2 * x_{nd}^2 \right)$$

$$\mathcal{L}_{new} = \sum_{n=1}^{N} \left((y_n - \mathbf{w}_p^T \mathbf{x}_n)^2 + \lambda * \sum_{d=1}^{D} w_{pd}^2 * x_{nd}^2 \right), \quad \text{where } \lambda = \frac{1}{p} - 1$$

Here $\mathbf{w}_p = p\mathbf{w}$ with it's dth element as w_{pd}.

$$\mathcal{L}_{new} = \sum_{n=1}^{N} (y_n - \mathbf{w}_p^T \mathbf{x}_n)^2 + \lambda * \sum_{n=1}^{N} \sum_{d=1}^{D} w_{pd}^2 * x_{nd}^2, \quad \text{where } \lambda = \frac{1}{p} - 1$$

. Hence, the regularised term in corresponding regularised function ($\mathcal{L}_{new}$) is
$\lambda * \sum_{n=1}^{N} \sum_{d=1}^{D} w_{pd}^2 * x_{nd}^2$ with $\lambda = \frac{1}{p} - 1$

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

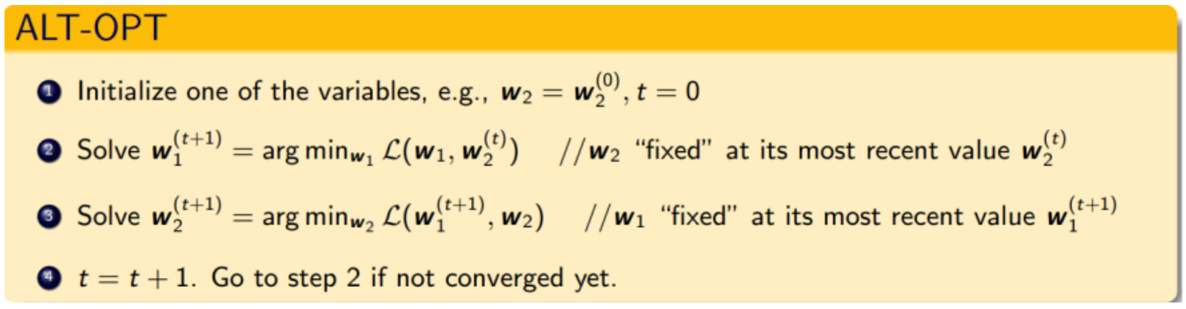
Student Name: Abhay
Roll Number: 180014
Date: October 30, 2020

QUESTION
3

We will use the notations given in the question problem. We are considering the squared error loss $\sum_{n=1}^{N}\sum_{m=1}^{M}(y_{nm}-\mathbf{w}_m^T\mathbf{x}_n)^2$. It is quite clear that it is equivalent to trace$[(\mathbf{Y}-\mathbf{XW})^T(\mathbf{Y}-\mathbf{XW})]$. This can be easily verified by visualising each diagonal term of the multiplication of the matrices whose summation is equal to squared loss. Now according to question $\mathbf{W}=\mathbf{BS}$, where $\mathbf{B}$ is a $D\times K$ and $\mathbf{S}$ is $K\times M$ matrix. Now the final objective function is trace$[(\mathbf{Y}-\mathbf{XBS})^T(\mathbf{Y}-\mathbf{XBS})]$. The problem can be written as

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

From the notes we have defined ALT-OPT as shown in Fig 1


ALT-OPT

1. Initialize one of the variables, e.g., $\boldsymbol{w}_2=\boldsymbol{w}_2^{(0)}, t=0$
2. Solve $\boldsymbol{w}_1^{(t+1)}=\arg\min_{\boldsymbol{w}_1}\mathcal{L}(\boldsymbol{w}_1,\boldsymbol{w}_2^{(t)})$ $\quad//\boldsymbol{w}_2$ "fixed" at its most recent value $\boldsymbol{w}_2^{(t)}$
3. Solve $\boldsymbol{w}_2^{(t+1)}=\arg\min_{\boldsymbol{w}_2}\mathcal{L}(\boldsymbol{w}_1^{(t+1)},\boldsymbol{w}_2)$ $\quad//\boldsymbol{w}_1$ "fixed" at its most recent value $\boldsymbol{w}_1^{(t+1)}$
4. $t=t+1$. Go to step 2 if not converged yet.

Figure 1: ALT-OPT

Here, we consider $\mathbf{S}$ as $\mathbf{w}_1$ and $\mathbf{B}$ as $\mathbf{w}_2$. For the updates we need to differentiate the function. Our objective function is

$$\mathcal{L}(B,S)=Tr[(\mathbf{Y}-\mathbf{XBS})^T(\mathbf{Y}-\mathbf{XBS})]$$

$$\mathcal{L}(B,S)=Tr[(\mathbf{Y}^T-\mathbf{S}^T\mathbf{B}^T\mathbf{X}^T)(\mathbf{Y}-\mathbf{XBS})]$$

$$\mathcal{L}(B,S)=Tr[\mathbf{Y}^T\mathbf{Y}-\mathbf{Y}^T\mathbf{XBS}-\mathbf{S}^T\mathbf{B}^T\mathbf{X}^T\mathbf{Y}+\mathbf{S}^T\mathbf{B}^T\mathbf{X}^T\mathbf{XBS}]$$

Using identities from Matrix Cookbook Section 2.5, treating $\mathbf{B}$ as a constant, we have

$$\frac{\partial\mathcal{L}}{\partial S}=0-(\mathbf{Y}^T\mathbf{XB})^T-\mathbf{B}^T\mathbf{X}^T\mathbf{Y}+(\mathbf{B}^T\mathbf{X}^T\mathbf{XB})^T\mathbf{S}+\mathbf{B}^T\mathbf{X}^T\mathbf{XBS}$$

In order to minimise it, we put $\frac{\partial\mathcal{L}}{\partial S}=0$, hence we got

$$0=0-2\mathbf{B}^T\mathbf{X}^T\mathbf{Y}+2\mathbf{B}^T\mathbf{X}^T\mathbf{XBS}$$

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

Hence the update for S is given by

$$\mathbf{S}=(\mathbf{B}^T\mathbf{X}^T\mathbf{XB})^{-1}\mathbf{B}^T\mathbf{X}^T\mathbf{Y}$$

Now we calculate update for B when S is fixed. Differentiating $\mathcal{L}(B, S)$ w.r.t to B we have

$$\frac{\partial \mathcal{L}}{\partial B} = 0 - (\mathbf{Y}^T\mathbf{X})^T\mathbf{S}^T - \mathbf{X}^T\mathbf{Y}\mathbf{S}^T + \mathbf{X}^T\mathbf{X}\mathbf{B}\mathbf{S}\mathbf{S}^T + (\mathbf{X}^T\mathbf{X})^T\mathbf{B}\mathbf{S}\mathbf{S}^T$$

In order to minimize it, we put $\frac{\partial \mathcal{L}}{\partial B} = 0$, hence we got

$$0 = 0 - 2\mathbf{X}^T\mathbf{Y}\mathbf{S}^T + 2\mathbf{X}^T\mathbf{X}\mathbf{B}\mathbf{S}\mathbf{S}^T$$

$$\mathbf{X}^T\mathbf{X}\mathbf{B}\mathbf{S}\mathbf{S}^T = \mathbf{X}^T\mathbf{Y}\mathbf{S}^T$$

$$\mathbf{B} = (\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{Y}\mathbf{S}^T)(\mathbf{S}\mathbf{S}^T)^{-1}$$

Thus algo APT-OPT can be summarised as shown

```
Procedure
1. Initialise $\mathbf{B}$ as $\mathbf{B}_0$.
2. For t= 0 to $\infty$
    • update $\mathbf{S}_{t+1} = (\mathbf{B}_t^T\mathbf{X}^T\mathbf{X}\mathbf{B}_t)^{-1}\mathbf{B}_t^T\mathbf{X}^T\mathbf{Y}$
    • update $\mathbf{B}_{t+1} = (\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{Y}\mathbf{S}_{t+1}^T)(\mathbf{S}_{t+1}\mathbf{S}_{t+1}^T)^{-1}$
    • if converged : break from the loop
3. End procedure
```

Since the update of S involves calculation of only one inverse of a matrix $(K \times K)$ while the update of B involves calculation of 2 inverses ($D \times D$ matrix and $K \times K$ matrix), thus the sub problem to calculate B is more difficult than calculating the sub-problem for calculating S.

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

Student Name: Abhay
Roll Number: 180014
Date: October 30, 2020

QUESTION 4

The objective function for ridge regression as given in question is

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

From class, we know that Newton's update equation is given by

$$\mathbf{w}_{t+1} = \mathbf{w}_t - H_t^{-1}\nabla\mathcal{L}(\mathbf{w}_t)$$

, where H is hessian matrix ($\nabla^2 L(w)$). Thus we need to find $\frac{\partial\mathcal{L}}{\partial\mathbf{w}}$ first. We have

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

Now using rules of vector differentiation, we have

$$\frac{\partial\mathcal{L}}{\partial\mathbf{w}} = \frac{1}{2}(0 - (\mathbf{y}^T\mathbf{X})^T - \mathbf{X}^T\mathbf{y} + \mathbf{X}^T\mathbf{X}\mathbf{w} + (\mathbf{X}^T\mathbf{X})^T w) + \frac{\lambda}{2}(\mathbf{w} + \mathbf{w})$$

$$\frac{\partial\mathcal{L}}{\partial\mathbf{w}} = -\mathbf{X}^T\mathbf{y} + \mathbf{X}^T\mathbf{X}\mathbf{w} + \lambda\mathbf{w} \tag{5}$$

Now for optimal $\mathbf{w}$,say $\mathbf{w}_{opt}$ the term $\frac{\partial\mathcal{L}}{\mathbf{w}}$ in eqn 5 must be equal to 0. Hence we have

$$\mathbf{X}^T\mathbf{X}\mathbf{w}_{opt} + \lambda\mathbf{w}_{opt} = \mathbf{X}^T\mathbf{y}$$

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

Now for Hessian matrix we again need to differentiate this term, thus we have

$$H = \frac{\partial^2\mathcal{L}}{\partial\mathbf{w}^2} = 0 + (\mathbf{X}^T\mathbf{X})^T + \lambda\mathbf{I}_D$$

$$H = \frac{\partial^2\mathcal{L}}{\partial\mathbf{w}^2} = (\mathbf{X}^T\mathbf{X}) + \lambda\mathbf{I}_D \tag{6}$$

Here I_D represents the identity matrix for $D \times D$ dimensions. Putting values from eqn 5 and 6 into update eqn we have

$$\mathbf{w}_{t+1} = \mathbf{w}_t - (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_D)^{-1}(-\mathbf{X}^T\mathbf{y} + \mathbf{X}^T\mathbf{X}\mathbf{w}_t + \lambda\mathbf{w}_t)$$

After first iteration i.e for t=0 we have

$$\mathbf{w}_1 = \mathbf{w}_0 - (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_D)^{-1}(-\mathbf{X}^T\mathbf{y} + \mathbf{X}^T\mathbf{X}\mathbf{w}_0 + \lambda\mathbf{w}_0)$$

$$\mathbf{w}_1 = \mathbf{w}_0 - (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_D)^{-1}(-\mathbf{X}^T\mathbf{y} + (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_D)\mathbf{w}_0)$$

$$\mathbf{w}_1 = \mathbf{w}_0 - ((\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_D)^{-1})(-\mathbf{X}^T\mathbf{y}) - \mathbf{w}_0$$

$$\mathbf{w}_1 = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_D)^{-1}(\mathbf{X}^T\mathbf{y})$$

Since $\mathbf{w}_1$ is equal to $\mathbf{w}_{opt}$.hence proved that convergence occurs after the first iteration itself.

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

Student Name: Abhay
Roll Number: 180014
Date: October 30, 2020

QUESTION 5

For N rolls of a die(6-faced), given the probability vector $\pi = [\pi_1, \pi_2, \ldots, \pi_6]$,$(\sum \pi_i = 1)$ suppose we obtain output as $\mathbf{y}$ which is represented as $N_1, N_2, \ldots, N_6$ appearances of 6 faces respectively. Now the likelihood i.e. PMF is given by

$$p(\mathbf{y}|\boldsymbol{\pi}) = \frac{N!}{N_1!N_2!\ldots N_6!}(\pi_1^{N_1}\pi_2^{N_2}\ldots\pi_6^{N_6}) = \frac{\Gamma(\sum\limits_{i=1}^{6} N_i + 1)}{\prod\limits_{i=1}^{6}\Gamma(N_i+1)}\prod_{i=1}^{6}\pi_i^{N_i}$$

The above expression is multi nomial distribution whose conjugate prior is Dirichlet distribution. Thus for the probability vector π, we will consider Dirichlet distribution.

$$p(\boldsymbol{\pi}) = \frac{1}{\mathbf{B}(\boldsymbol{\alpha})}\prod_{i=1}^{6}\pi_i^{\alpha_i-1} = \frac{\Gamma(\sum\limits_{i=1}^{6}\alpha_i)}{\prod\limits_{i=1}^{6}\Gamma(\alpha_i)}\prod_{i=1}^{6}\pi_i^{\alpha_i-1}$$

Now the posterior is given by

$$p(\boldsymbol{\pi}|\mathbf{y}) = \frac{p(\boldsymbol{\pi})p(\mathbf{y}|\boldsymbol{\pi})}{p(\mathbf{y})} \tag{7}$$

For MAP we essentially calculate $\arg\max\limits_{\pi} p(\pi|\mathbf{y})$ which will depend upon only on numerator term. From notes we have,

$$\boldsymbol{\pi}_{MAP} = \arg\max_{\boldsymbol{\pi}}[\log p(\mathbf{y}|\boldsymbol{\pi}) + \log p(\boldsymbol{\pi}))]$$

but we also have a constraint $\sum \pi_i = 6$ so we will use Lagrangian method to find optima for the following problem...

$$\mathcal{L}(\boldsymbol{\pi}) = \log p(\mathbf{y}|\boldsymbol{\pi}) + \log p(\boldsymbol{\pi})) + \lambda(1 - \sum_{i=1}^{6}\pi_i)$$

$$\log p(\mathbf{y}|\boldsymbol{\pi}) = \log(\Gamma(\sum_{i=1}^{6} N_i + 1)) - \log(\prod_{i=1}^{6}\Gamma(N_i+1)) + \log(\prod_{i=1}^{6}\pi_i^{N_i})$$

$$\log p(\boldsymbol{\pi}) = \log(\Gamma(\sum_{i=1}^{6}\alpha_i)) - \log(\prod_{i=1}^{6}\Gamma(\alpha_i)) + \log(\prod_{i=1}^{6}\pi_i^{\alpha_i-1})$$

Now in order to optimise,

$$\nabla_{\pi_1,\ldots,\pi_6,\lambda}(L(\boldsymbol{\pi}),\lambda) = 0$$

. Differentiating $\mathcal{L}$ w.r.t to π_i for each i, we get

$$\frac{\partial\mathcal{L}}{\partial\pi_i} = 0 - 0 + \frac{N_i}{\pi_i} + 0 - 0 + \frac{\alpha_i - 1}{\pi_i} - \lambda$$

Since for each i, $\frac{\partial \mathcal{L}}{\partial \pi_i} = 0$,

$$\pi_i = \frac{N_i + \alpha_i - 1}{\lambda} \tag{8}$$

Differentiating $\mathcal{L}$ w.r.t to λ, we get

$$1 - \sum_{i=1}^{6} \pi_i = 0$$

Substituting value of π_i's in this we get,

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

Putting value of λ in eqn 8, we get

$$\pi_i = \frac{N_i + \alpha_i - 1}{N - 6 + \sum_{i=1}^{6}(\alpha_i)} \quad \forall i \in [1, 6]$$

The above expression represents the MAP estimate of the following problem using Dirichlet prior.

MAP better than MLE! When?

The MAP solution is better than the MLE one when the prior is not uniform(i.e $\alpha_i \neq 1 \forall i$) and number of training examples are small in number. The reason behind is that MLE solution tends to over fit in case of small size of training set. The prior term in the function $\mathcal{L}_{MAP}$ act as a regulariser. The MAP is also better when die hasn't rolled one or more than one number since in this case MLE will predict zero probability for those numbers.

Posterior

Now for calculation posterior, we have eqn 7. Hence

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

$$p(\boldsymbol{\pi}|\mathbf{y}) \propto \prod_{i=1}^{6} \pi_i^{N_i} . \frac{1}{\mathbf{B}(\boldsymbol{\alpha})} \prod_{i=1}^{6} \pi_i^{\alpha_i - 1}$$

$$p(\boldsymbol{\pi}|\mathbf{y}) \propto \prod_{i=1}^{6} \pi_i^{N_i + \alpha_i - 1}$$

Now, the above proportionality is the kernel for Dirichlet distribution(parameters: $N_i + \alpha_i > 0$ since $\alpha_i > 0$ and $N_i \geq 0$) which means if we find the constant of proportionality it will be same as the constant of Dirichlet expression. Thus the posterior is given by

$$p(\boldsymbol{\pi}|\mathbf{y}) = \frac{\Gamma \sum_{i=1}^{6}(N_i + \alpha_i)}{\sum_{i=1}^{6} \Gamma(N_i + \alpha_i)} \prod_{i=1}^{6} \pi_i^{N_i + \alpha_i - 1} \tag{9}$$

Calculate MAP and MLE from posterior - Yes it's possible

In order to get MAP estimate, we directly find the mode of the posterior distrubution. For MLE,which have a uniform prior, we can put $\alpha_i = 1 \; \forall i \in [1, 6]$ in posterior and then find its mode. We can also find MLE by directly putting $\alpha_i = 1 \; \forall i \in [1, 6]$ in the result of MAP estimate.