Indian Institute of Technology Kanpur
CS771 Introduction to Machine Learning, 2017-18-a

Assignment Number: 2
Student Name: Raktim Mitra
Roll Number: 150562
Date: October 10, 2017

QUESTION 1

Part 1:

Row number 4 and 6 has exactly same attributes except the name and one is **Yes** while other one is **No**. That means the 'Name' field will be required to separate them. Hence, that field will be useful. Same applies for 6 and 14.

Part 2:

If we do not consider the **Name** attribute then it is not possible to classify the data no matter how complicated the algorithm is, because as said earlier there are pairs of rows for which all attributes are same except name but they belong to different classes.

If we consider the name attribute then it is posssible to perfectly classify the data since no two names are same. Although **Name** is very less informative as a decision node, but to perfectly classify it is needed.

Part 3:

From the ID3 algorithm : Information gain by choosing an attribute is calculated as:

$$Gain(S,A) = Entropy(S) - \Sigma_i \frac{|S_i|}{|S|} Entropy(S_i) \tag{1}$$

where $|S_i|$ is set of elements of S with same value of Attribute A. Entropy of S(5 Yes, 10 No):

$$Entropy(S) = -\frac{5}{15} log \frac{5}{15} - \frac{10}{15} log \frac{10}{15} = 0.9183 \tag{2}$$

Now, values of entropy :
We have used $lim_{x \to 0} xlog(x) = 0$

$$\begin{aligned}
Entropy(S_{size=small}) &= 0.6500 \\
Entropy(S_{size=medium}) &= 0.9710 \\
Entropy(S_{size=large}) &= 1.0 \\
Entropy(S_{likearea=yes}) &= 1.0 \\
Entropy(S_{likearea=no}) &= 0.8454 \\
Entropy(S_{workload=light}) &= 0.8113 \\
Entropy(S_{workload=average}) &= 1.0 \\
Entropy(S_{workload=heavy}) &= 0.7219 \\
Entropy(S_{meetings=0-1}) &= 0.9710 \\
Entropy(S_{meetings=2-3}) &= 0.0 \\
Entropy(S_{meetings>3}) &= 0.0
\end{aligned}$$

Gain calculation for root node:

$$\begin{aligned}
Gain(S, size) &= 0.0680 \\
Gain(S, like) &= 0.0317 \\
Gain(S, workload) &= 0.0613 \\
Gain(S, meetings) &= 0.2710
\end{aligned}$$

Conditioning on **number of meetings** attribute gives maximum gain, so root will use this attribute to separate.

$$\begin{aligned}
Gain(S_{0-1,workload}) &= 0.41 \\
Gain(S_{0-1,researcharea}) &= 0.0587 \\
Gain(S_{0-1}, size) &= 0.085
\end{aligned}$$

Hence the 0-1 branch from root will use **workload** as next attribute to branch upon.

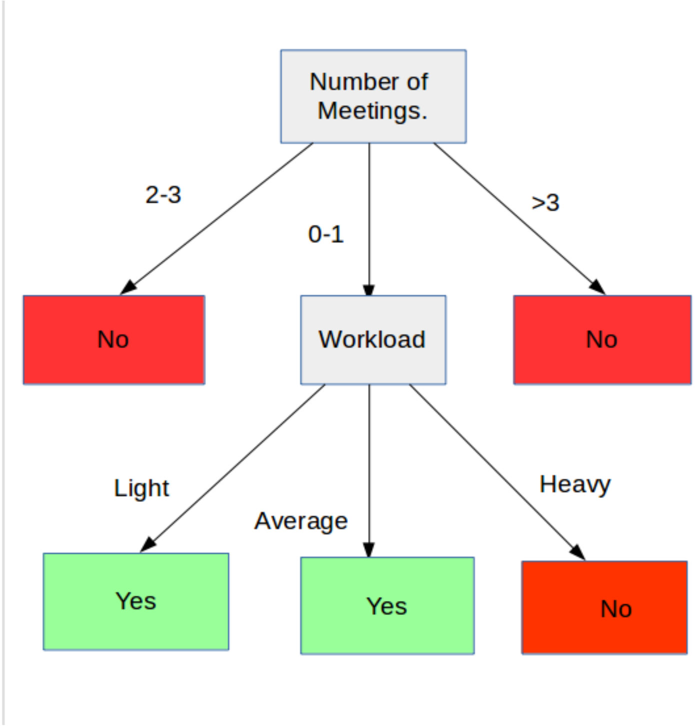

Figure 1: Decision Tree

Number of Meetings 2-3 and NUmber of Meetings ¿3 results into two pure leaf nodes with No labels.

From 0-1 path at workload node 3 branches are generated from the node using workload as attribute. The average node becomes pure leaf node. The light node has a tie with 1 Yes and 1 No. Arbitrarily assigning it a Yes leaf node. The Heavy node has a majority of No, so making it a No leaf node.

Indian Institute of Technology Kanpur
CS771 Introduction to Machine Learning, 2017-18-a

Assignment Number: 2
Student Name: Raktim Mitra
Roll Number: 150562
Date: October 10, 2017

QUESTION 2

Part 1:

We want to model $\mathbb{P}[b^i|\mathbf{x^i},\mathbf{\Theta}]$. We can write it using conditional probability, summed over all possible combinations , as follows:

$$\mathbb{P}[b^i|\mathbf{x^i},\mathbf{\Theta}] = \sum_{j=1}^{2^L} \mathbb{P}[b^i|S^j,\mathbf{x^i},\mathbf{\Theta}].\mathbb{P}[S^j|\mathbf{x^i},\mathbf{\Theta}]$$

Probability $\mathbb{P}[b^i|S^j,\mathbf{x^i},\mathbf{\Theta}]$ is modeled from $\mathcal{N}(\mu_i,\sigma^2)$.

Now we assume the customers choice among products are independent and hence we get $\mathbb{P}[S^j|\mathbf{x^j},\mathbf{\Theta}] = \prod_{k=1}^{L} \mathbb{P}[S_k^j|\mathbf{x^j},\mathbf{\Theta}]$.

Using hint, we model $\mathbb{P}[S_k^j = 1|\mathbf{x^j},\mathbf{\Theta}] = sigmoid\langle w_k, x^j\rangle$

Our Model $\Theta = \{\mathbf{W} = \mathbf{w}_i \forall i \epsilon [L], \mu_i \forall i \epsilon [n], \sigma\}$

S^i's are our latent variables in this problem. Using all these finally our model becomes:

$$\mathbb{P}[b^i|x^i,\Theta] = \prod_{i=1}^{n} \sum_{j=1}^{2^L} \mathbb{P}[b|S^j,\mathbf{x},\mathbf{\Theta}].\Big[\prod_{k=1}^{L} \mathbb{P}[S_k^j|\mathbf{x},\mathbf{\Theta}]\Big] \tag{3}$$

Part 2:

Assuming each customer is independent, we can write $\mathbb{P}[\mathbf{b}|\mathbf{x},\mathbf{\Theta}]$ as products of each customer probabilities: Here $\mathbf{b}$ is a vector of b^i for each customer i.

$$\begin{aligned}
\mathbb{P}[\mathbf{b}|\mathbf{x},\mathbf{\Theta}] &= \prod_{i=1}^{n} P[b^i \mid \mathbf{x}^i, \mathbf{\Theta}] \\
\mathbb{P}[\mathbf{b}|\mathbf{x},\mathbf{\Theta}] &= \prod_{i=1}^{n} \sum_{j=1}^{2^L} \mathbb{P}\left[b^i|\mathbf{x}^i, S^j, \mathbf{\Theta}\right] \left[\prod_{k=1}^{L} \mathbb{P}\left[S_k^j|\mathbf{x}^i,\mathbf{\Theta}\right]\right] \\
&= \prod_{i=1}^{n} \sum_{j=1}^{2^L} \mathcal{N}\left(\mu_j,\sigma^2\right) \left[\prod_{k=1}^{L} \mathbb{P}\left[S_k^j|\mathbf{x}^i,\mathbf{\Theta}\right]\right]
\end{aligned}$$

Now this gives our MLE estimate of the model: $\mathbf{\Theta}_{MLE} = \arg\max_{\mathbf{\Theta}} P[\mathbf{b}|\mathbf{X},\mathbf{\Theta}]$

Part 3:Hard Assignment Alternating Optimization:

> Initialize $\mathbf{\Theta}^0$.
>Update S^i $\forall i$. We do this by finding S^i such that $P[S \,|\, \mathbf{x}^i, b^i, \mathbf{\Theta}^t]$ is maximised.

$$S^i = \arg\max_{S^i} \mathbb{P}[S^i|\mathbf{x}^i, b^i, \mathbf{\Theta}]$$

where

$$\begin{aligned}\mathbb{P}\left[S \,|\ \mathbf{x}^i, b^i, \mathbf{\Theta}\right] &\propto P[S^i|\mathbf{x}^i, \mathbf{\Theta}] \prod_{j=1}^{L} \mathbb{P}\left[S^i_j|\mathbf{x}^i, b^i, \mathbf{\Theta}\right] \\ &= \mathcal{N}(\mu^i, \Sigma) \prod_{j=1}^{L} \sigma(\mathbf{w}_j^T \mathbf{x}^i)^{S^i_j} [1 - \sigma(\mathbf{w}_j^T \mathbf{x}^i)]^{1-S^i_j}\end{aligned}$$

We set S^i_j as 1 or 0 depending on whether it's probability is greater than 0.5 or not.

> Now we update $\mathbf{\Theta}^{t+1}$ using the new obtained S^i.

$$\begin{aligned}\mathbf{\Theta}^{t+1} &= \arg\max_{\mathbf{\Theta}} \mathbb{P}[\mathbf{b}, S|X, \mathbf{\Theta}] \\ &= \arg\max_{\mathbf{\Theta}} \prod_{i=1}^{n} \mathbb{P}[b^i \,|\, S^i, x^i, \mathbf{\Theta}] \prod_{j=1}^{L} \mathbb{P}[S^i_j \,|\, \mathbf{x}^i, \mathbf{\Theta}] \\ &= \arg\max_{\mathbf{\Theta}} \prod_{i=1}^{n} \left(\mathcal{N}(\mu^i, \sigma^2) \prod_{j=1}^{L} \sigma(\mathbf{w}_j^T \mathbf{x}^i)^{S^i_j} [1 - \sigma(\mathbf{w}_j^T \mathbf{x}^i)]^{1-S^i_j} \right)\end{aligned}$$

It can be easily seen using first order derivative method we find: $\mu_i = \sum_{j|S^i_j=1} C_j$
$\sigma_i^2 = \frac{\sum_{i=1}^{n}(x^i - \mu^i)^2}{n}$ as expected in the problem scenario.

> Repeat until Convergence.

Part 4:soft Alternating optimization algorithm:

>For $i \in [n]$ create 2^L copies of datapoint x^i. $x^i to\{x^{i,1}, x^{i,2}, ..., x^{i,2^L}\}$
>Initialize $\mathbf{\Theta}^0$.Let,$\pi_j = P[S^i = j|\mathbf{x}^i, \mathbf{\Theta}]$ which is product of linear logistics desribed in part 1. Update weights $\gamma^{i,k,t}$ using $\mathbf{\Theta}^0$

$$\begin{aligned}\gamma^{i,k,t} &= P[S^i|b^i, \mathbf{x}^i, \mathbf{\Theta}^t] \\ &= \frac{P[S^i = k \,|\, \mathbf{x}^i, \mathbf{\Theta}] \ P[b^i \,|\, x^i, S^i = k, \mathbf{\Theta}]}{\sum_{j=1}^{j=2^L} P[S^i = j \,|\, \mathbf{x}^i, \mathbf{\Theta}] \ P[b^i \,|\, x^i, S^i = j, \mathbf{\Theta}]} \\ &= \frac{\boldsymbol{\pi}_k^t \ \mathcal{N}(\mu_i, \sigma^2)}{\sum_{j=1}^{j=2^L} \boldsymbol{\pi}_j^t \ \mathcal{N}(\mu_i, \sigma^2)}\end{aligned}$$

> Update $\mathbf{\Theta}^{t+1}$,

$$\mathbf{\Theta}^{t+1} = \arg\max_{\mathbf{\Theta}} P[\{\mathbf{x}^{i,k}\}, \{b^{i,k}\}, \{\gamma^{i,k,t}\}]$$

where $P[\mathbf{x}, b, \gamma \ |\mathbf{\Theta}] = P[x, b|\mathbf{\Theta}]^{\gamma}$.
> Repeat until Convergence

Indian Institute of Technology Kanpur
CS771 Introduction to Machine Learning, 2017-18-a

Assignment Number: 2
Student Name: Raktim Mitra
Roll Number: 150562
Date: October 10, 2017

QUESTION 3

Part 1:

Claim: $\xi_i \geq 0$ constraints are vacuous.
Proof: Let there is an optimal solution with some $\xi_i < 0$. now, since $\xi_i < 0$ it follows that $1 - \xi_i > 1 - 0$ i.e. $y^i\langle \mathbf{w}, \mathbf{x}^{\mathbf{i}}\rangle$ strictly greater than 1. This means we can replace negative ξ_i's with 0. That will also reduce the objective funstion by an amount ξ_i^2. Therefore, we have a contradiction as the solution we took as optimal is not optimal.

Hence ξ_i cannot be negative in an optimal solution of the problem. Hence the $\xi_i \geq 0$ conditions are vacuous.

Now, let there is an optimal solution of the problem with out the constraints. Then we essentially get an optimal value of ξ_i^2 which has anyway a positive root ξ_i which is greater than or equal to 0. Hence it holds. [Proved]

Part 2:

$y^i \left\langle \mathbf{w}, \mathbf{x}^i \right\rangle \geq 1 - \xi_i \implies 1 - \xi_i - y^i \left\langle \mathbf{w}, \mathbf{x}^i \right\rangle \leq 0$ and $\xi_i \geq 0 \implies -\xi_i \leq 0$

We introduce lagrange parameters α_i's and β_i's. The langrangian becomes:

$$L(\mathbf{w}, \{\xi_i\}, \{\alpha_i\}, \{\beta_i\}) = \|\mathbf{w}\|_2^2 + \sum_{i=1}^{n} \xi_i^2 + \sum_{i=1}^{n} \alpha_i(1 - \xi_i - y^i \left\langle \mathbf{w}, \mathbf{x}^i \right\rangle) - \sum_{i=1}^{n} \beta_i \xi_i$$

Part 3:

We want to solve objective

$$\underset{\mathbf{w},\{\xi_i\}}{\arg\min}\ \underset{\alpha_i \geq 0, \beta_i \geq 0}{\arg\max}\quad L(\mathbf{w}, \{\xi_i\}, \{\alpha_i\}, \{\beta_i\})$$

Derivative w.r.t $\mathbf{w}$:

$2\mathbf{w} + 0 + \sum_{i=1}^{n} \alpha_i * (-y^i \mathbf{x}^i) = 0 \implies \mathbf{w} = \frac{\sum_{i=1}^{n} \alpha_i y^i \mathbf{x}^i}{2}$

Derivative w.r.t ξ_i:

$2\xi_i - \alpha_i - \beta_i = 0 \implies \xi_i = \frac{\alpha_i + \beta_i}{2}$

Substituting these values in Lagrangian:

$$\begin{aligned} &f(\{\alpha_i\}, \{\beta_i\}) \\ &\quad = \frac{1}{4}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y^i y^j (\mathbf{x}^i)^T \mathbf{x}^j + \frac{1}{4}\sum_{i=1}^{n}(\alpha_i + \beta_i)^2 + \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \alpha_i y^i \left\langle \frac{1}{2}\sum_{j=1}^{n} \alpha_j y^j \mathbf{x}^j, \mathbf{x}^i \right\rangle \\ &\quad\quad - \frac{1}{2}\alpha_i(\alpha_i + \beta_i) - \frac{1}{2}\beta_i(\alpha_+ \beta_i) \\ &\quad = \sum_{i=1}^{n} \alpha_i - \frac{1}{4}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y^i y^j (\mathbf{x}^i)^T \mathbf{x}^j - \sum_{i=1}^{n} \frac{(\alpha_i + \beta_i)^2}{4} \end{aligned}$$

Hence:

$$f(\{\alpha_i\}, \{\beta_i\}) = \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \frac{(\alpha_i + \beta_i)^2}{4} - \frac{1}{4}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y^i y^j (\mathbf{x}^i)^T \mathbf{x}^j$$

where $\alpha_i \geq 0, \beta_i \geq 0 \ \forall i$.
$(\alpha_i + \beta_i)^2$, being a non-negative quantity,it follows that the dual is maximised when it becomes as small as possible. No other term in the expression has β,hence we can set it to $0(\beta_i \geq 0)$. Therefore the constraint of $\xi_i \geq 0$ was clearly vacuous.
Hence, the Dual Problem is finally:

$$f(\{\alpha_i\}) = \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \frac{\alpha_i^2}{4} - \frac{1}{4}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y^i y^j (\mathbf{x}^i)^T \mathbf{x}^j$$

where $\alpha_i \geq 0 \ \forall i$

Part 4:

Dual problem for original SVM :

$$f(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y^i y^j (\mathbf{x}^i)^T \mathbf{x}^j$$

where $0 \leq \alpha_i \leq C$

1. In the original SVM we have an upper bound for α_i which is result of the constraints $\xi_i > 0$. Here we do not have that because those constraints are vacuous here.

2. Also there is an extra summation term on α_i^2 in this problem which is not present in original problem.

Part 5:

Those $\xi_i > 0$ constraints are not vacuous for the original problem. There we basically used ξ_is to allow some points to lie between the expected margin of classifications. Removing $\xi_i \geq 0$ would then may change the solution because it would mean we are actually allowing to points to be present further awaya from the argin of classification. Which means we may not get the highest possible margin of classification which we get from the constraint problem except for some allowance of points to viaolate the margin. Hence we cannot remove those constraints in the original problem.

Indian Institute of Technology Kanpur
CS771 Introduction to Machine Learning, 2017-18-a

Assignment Number: 2
Student Name: Raktim Mitra
Roll Number: 150562
Date: October 10, 2017

QUESTION 4

Part 3:

In the GD implementation averaged for small number of iterations ”wbar” was giving slightly better objective values and for large number of iterations there was not much noticable difference. So, I kept final averaged model for my solution.

Part 4:

My best step-length selection strategy was
$eta = \frac{c}{n}\frac{1}{\sqrt{t^k}+1}$ where I found c = 50, k = 0.9 to work best. n is number of data points, t is iteration number.

part 5:

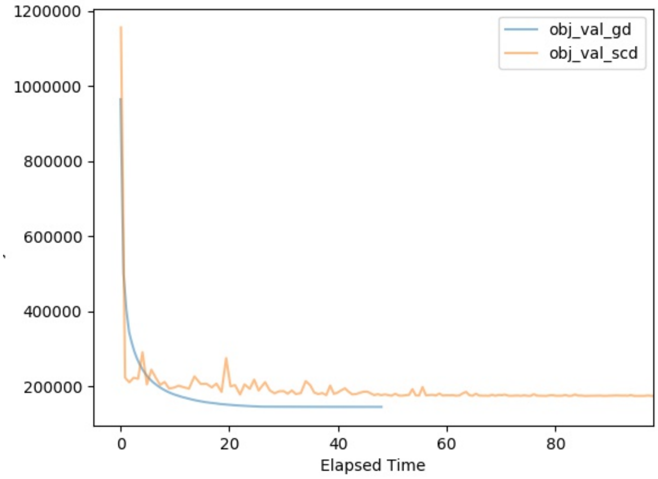

Figure 2: Objective Value against wall clock time for Gradient Decent and Stochastic Coordinate Descent

According to Wall Clock Time:
At the start SCD reduces the value faster but after initial fast reduction it becomes slower and GD reaches a better optimal value earlier.

Part 6:

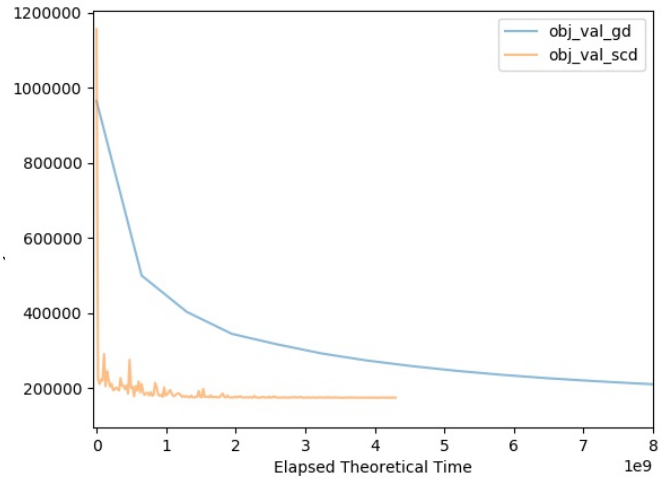

Figure 3: Objective Value against theoretical time for Gradient Decent and Stochastic Coordinate Descent

In theoretical time SCD is much much faster in converging ! I had to cut the graph along x-axis to even see the scd plot properly. This clearly shows how less computations SCD really takes.