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

QUESTION 1

Assignment Number: 2
Student Name: Siddharth Agrawal
Roll Number: 150716
Date: October 10, 2017

Part 1

No, the first attribute (*name*) is **not** useful in learning a binary classifier from this data.
This is because it is impossible to find any correlation between the name of a professor and his/her ability to advise.
Mathematically, *name* attribute can have infinitely many possible values and hence is not suitable to train the classifier.

Part 2

No, it is **not** possible to perfectly classify this data without using name attribute. Otherwise it is possible.

There are two data points which have all the attributes (except *name*) as similar but their labels are different (Example: Prof. S. Snape and Prof. H. Slughorn have the same attribute values, viz., medum-no-heavy-(0-1) but have different label - yes and no respectively). In this case our classifier will not be able to properly distinguish between the two.

Part 3

As per the `ID3 Decision Algorithm`-

- Entropy of a set S over c outcomes is defined as -

$$\text{Entropy}(S) = -\sum_{c} P(I) \log_2 P(I)$$

  where $P(I)$ is the proportion of S belonging to class I.

- $\text{Gain}(S, A)$ is the information gain of example S on attribute A, defined as -

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum_{v} \left(\frac{|S_v|}{|S|} \times \text{Entropy}(S_v) \right)$$

  where:

  - S is each value v of all possible values of attribute A.
  - S_v = subset of S for which attribute A has value v
  - $|S_v|$ = number of elements in S_v
  - $|S|$ = number of elements in S

For the entire sample set S
$P(\text{Yes}) = \frac{5}{15}, P(\text{No}) = \frac{10}{15} \implies \text{Entropy}(S) = 0.9183$

The value of gain for the following attributes are -

- Attribute: # of meetings per week
  Entropy$(0-1) = 1.0$
  Entropy$(2-3) = 0.0$
  Entropy$(>3) = 0.0$
  Gain $= 0.2516$

- Attribute: Average workload?
  Entropy(average) $= 1.0$
  Entropy(heavy) $= 0.7219$
  Entropy(light) $= 0.8113$
  Gain $= 0.0613$

- Attribute: Like the research area?
  Entropy(Yes) $= 1.0$
  Entropy(No) $= 0.8454$
  Gain $= 0.0317$

- Attribute: Size of research group
  Entropy(small) $= 0.6500$
  Entropy(medium) $= 0.9710$
  Entropy(large) $= 1.0$
  Gain $= 0.0680$

Since the Gain is maximum for Attrbute: # of meetings per week, we choose this as the decision attribute in root node.
Also, since for this attribute - Entropy$(2-3)$ = Entropy(>3) = 0, hence the set with these two values are *classified perfectly* and are assigned label 'no'.
For the remaining set with value $1-3$, we calculate gain as follows -

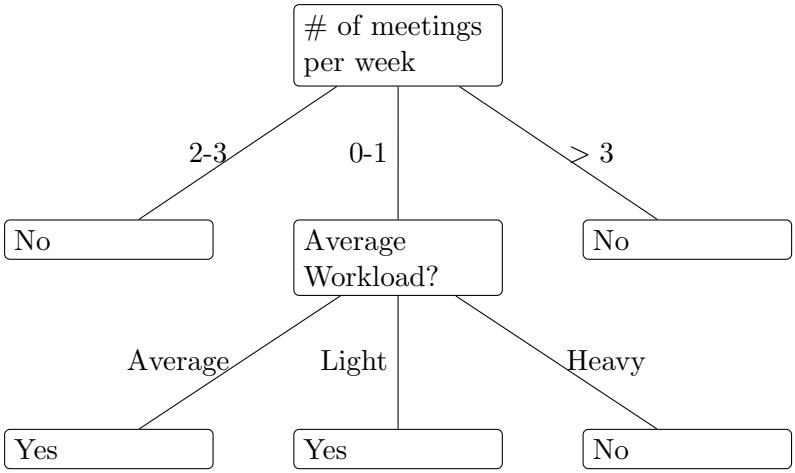
- Attribute: Average workload?
  Entropy(average) = 0.0
  Entropy(heavy) = 0.7219
  Entropy(light) = 1.0
  Gain = 0.4391

- Attribute: Like the research area?
  Entropy(Yes) = 0.9183
  Entropy(No) = 0.9852
  Gain = 0.0349

- Attribute: Size of research group
  Entropy(small) = 0.0
  Entropy(medium) = 0.9710
  Entropy(large) = 1.0
  Gain = 0.1145

Since the Gain is maximum for Attrbute: Average workload?, we choose this as the decision attribute in second level node.
Since we don't need to go beyond second level, hence we assign the majorly dominant labels to the sets.
Also, since for this attribute - Entropy(average) = 0, hence the set with this value is *classified perfectly* and is assigned the label 'yes'.
For the set with value average, since it has only 'yes' as the labels for all its data points so we assign its label as 'yes'.
For the set with value heavy, we assign the majorly dominant label 'no' to this set.
For the set with value light, there is one 'yes' and one 'no' for its two data points. So, we arbitrarily choose 'yes' as its label.

The learnt decision tree is:-


# of meetings per week
2-3
0-1
> 3
No
Average Workload?
No
Average
Light
Heavy
Yes
Yes
No

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

Assignment Number: 2
Student Name: Siddharth Agrawal
Roll Number: 150716
Date: October 10, 2017

QUESTION

2

Part 1

Consider the Cost Vector $\mathbf{C}$ (dimension: $L \times 1$) where $C^i = c_i$, cost of i^{th} item.
Let $\mathbf{z}^i$ be the latent vector of i^{th} customer and z^i_j denotes whether i^{th} customer buys j^{th} item or not (value = 0 or 1). Let $\mathbf{Z} = [\mathbf{z}^1, \mathbf{z}^2, ..., \mathbf{z}^n]$.

Now, for any i^{th} customer, $\mathbf{x}^i$ denotes the attributes of that customer. So, we have to find a model $\mathbf{w}$ (dimension: $L \times d$), such that $f(\mathbf{w}_j^\top \mathbf{x}^i)$ tells us whether that customer buys the j^{th} item or not.
Using linear regression model, probability that i^{th} customer buys j^{th} item is given by:

$$\mathbb{P}[j \in S^i] = \frac{1}{1 + \exp(-\mathbf{w}_j^\top \mathbf{x}^i)}$$

Let $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_L]$. Therefore our model, $\mathbf{\Theta} = \{\mathbf{W}, \sigma\}$. Using Naive Bayes Assumption:

$$\mathbb{P}[\mathbf{z}^i = \mathbf{z}_0 | \mathbf{x}^i, \mathbf{\Theta}] = \prod_{j=1}^{L} \frac{1}{1 + \exp((-1)^{z_{j,0}} \mathbf{w}_j^\top \mathbf{x}^i)}$$

Now, $b^i = \sum_{j=1}^{L} z^i_j c_j + \epsilon^i \simeq \mathcal{N}(\sum_{j=1}^{L} z^i_j c_j, \sigma^2)$, hence:

$$\mathbb{P}[b^i \,|\, \mathbf{z}^i, \mathbf{x}^i, \mathbf{\Theta}] = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(- \frac{\left(b^i - \sum_{j=1}^{L} z^i_j c_j\right)^2}{2\sigma^2} \right)$$

Part 2

Complete Likelihood expression for b^i given $\mathbf{x}^i$ and $\mathbf{\Theta}$:
Let S = set of all possible values of $\mathbf{z}^i$ ($|S| = 2^L$).

$$\mathbb{P}[b^i \,|\, \mathbf{x}^i, \mathbf{\Theta}] = \sum_{\mathbf{z}^i \in S} \mathbb{P}[b^i, \mathbf{z}^i \,|\, \mathbf{x}^i, \mathbf{\Theta}]$$

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

$$\mathbb{P}[b^i, \mathbf{z}^i \,|\, \mathbf{x}^i, \mathbf{\Theta}] = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\left(b^i - \sum_{j=1}^{L} z_j^i c_j\right)^2}{2\sigma^2} \right) \prod_{j=1}^{L} \frac{1}{1 + \exp((-1)^{z_j^i} \mathbf{w}_j^\top \mathbf{x}^i)}$$

Let $\mathbf{B} = [b^1, b^2, ..., b^n]$ and $\mathbf{X} = [\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^n]$
Complete Likelihood expression for B given $\mathbf{X}$ and $\mathbf{\Theta}$:

$$\mathbb{P}[\mathbf{B}, \mathbf{Z} \,|\, \mathbf{X}, \mathbf{\Theta}] = \prod_{i=1}^{n} \left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\left(b^i - \sum_{j=1}^{L} z_j^i c_j\right)^2}{2\sigma^2} \right) \prod_{j=1}^{L} \frac{1}{1 + \exp((-1)^{z_j^i} \mathbf{w}_j^\top \mathbf{x}^i)} \right)$$

Part 3

Hard Assignment Alternating Optimization Algorithm

1. Initialize $\mathbf{\Theta}^0 = \mathbf{W}^0, \sigma^0$
2. For $i \in [n]$, update $\mathbf{z}^{i,t}$ using $\mathbf{\Theta}^t$
   (a) For $j \in [d]$, let $z_j^{i,t} = \begin{cases} 1, & \frac{1}{1+\exp(-\mathbf{w}_j^{t\top}\mathbf{x}^i)} \geq 0.5 \\ 0, & \text{Otherwise} \end{cases}$
3. Update $\mathbf{\Theta}^{t+1} as$:

$$\mathbf{\Theta}^{t+1} = \arg\max_{\mathbf{\Theta}} \prod_{i=1}^{n} \left(\frac{1}{\sqrt{2\pi(\sigma^t)^2}} \exp\left(-\frac{\left(b^i - \sum_{j=1}^{L} z_j^{i,t} c_j\right)^2}{2(\sigma^t)^2} \right) \prod_{j=1}^{L} \frac{1}{1+\exp((-1)^{z_j^{i,t}} \mathbf{w}_j^\top \mathbf{x}^i)} \right)$$

4. Repeat until Convergence.

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

Assignment Number: 2
Student Name: Siddharth Agrawal
Roll Number: 150716
Date: October 10, 2017

QUESTION
3

Part 1

To prove that the given constraints $\xi_i \geq 0$ are vacuous, we just need to show that for any optimal solution $(\mathbf{w}, \{\xi_i\})$, $\nexists i \in [n]$ such that $\xi_i < 0$.

Claim: $\nexists k \in [n]$ *such that* $\xi_k < 0$ *where* $(\mathbf{w}, \{\xi_i\})$ *is the optimal solution.*
Proof by Contradiction:
Let the optimized solution be $(\mathbf{w}, \{\xi_i\})$ such that for some $k \in [n]$, $\xi_k < 0$ and $\xi_i \geq 0 \ \forall i \in [n] \backslash \{k\}$.
Let $\xi_k = -\epsilon$ where $\epsilon > 0$.
Hence, $y^k \left\langle \mathbf{w}, \mathbf{x}^k \right\rangle \geq 1 - \xi_k = 1 + \epsilon$.
$\because \ \epsilon > 0 \implies y^k \left\langle \mathbf{w}, \mathbf{x}^k \right\rangle > 1$
Now, consider the pair $(\mathbf{w}, \{\xi_i'\})$ such that $\xi_k' = 0$ and $\xi_i' \geq 0 \ \forall i \in [n] \backslash \{k\}$.
Clearly $\forall i \in [n] \backslash \{k\}, \ y^i \left\langle \mathbf{w}, \mathbf{x}^i \right\rangle \geq 1 - \xi_i'$. Also, $y^k \left\langle \mathbf{w}, \mathbf{x}^k \right\rangle > 1 \implies 1 - \xi_k'$. Thus $\{\xi_i'\}$ satisfies the given constraint.

$$
\begin{aligned}
f(\mathbf{w}, \{\xi_i'\}) &= \|\mathbf{w}\|_2^2 + \sum_{i=1}^{n} \xi_i'^2 \\
&= \|\mathbf{w}\|_2^2 + \sum_{\substack{i=0 \\ i \neq k}}^{n} \xi_i'^2 + \xi_k'^2 \\
&= \|\mathbf{w}\|_2^2 + \sum_{\substack{i=0 \\ i \neq k}}^{n} \xi_i'^2 && \because \ \xi_k' = 0 \\
&= \|\mathbf{w}\|_2^2 + \sum_{\substack{i=0 \\ i \neq k}}^{n} \xi_i^2 \\
&< \|\mathbf{w}\|_2^2 + \sum_{\substack{i=0 \\ i \neq k}}^{n} \xi_i^2 + \xi_k^2 && \because \ \xi_k < 0 \implies \xi_k^2 > 0 \\
&= \|\mathbf{w}\|_2^2 + \sum_{i=1}^{n} \xi_i^2 \\
&= f(\mathbf{w}, \{\xi_i\})
\end{aligned}
$$

Thus $f(\mathbf{w}, \{\xi_i'\}) < f(\mathbf{w}, \{\xi_i\})$.
But we assumed $(\mathbf{w}, \{\xi_i'\})$ to be the optimal solution – which is a clear Contradiction.
Hence our original assumption was wrong. Thus our claim is proved by contradiction.

Part 2

The Lagrangian for $(P1)$ is:

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

Part 3

Primal Problem:

$$(\widehat{\mathbf{w}}_P, \{\widehat{\xi_i}\}_P) = \underset{\mathbf{w},\{\xi_i\}}{\arg\min} \underset{\substack{\alpha_i \geq 0 \\ i \in [n]}}{\arg\max} \ \mathcal{L}(\mathbf{w}, \{\xi_i\}, \boldsymbol{\alpha})$$

Dual Problem:

$$(\widehat{\mathbf{w}}_D, \{\widehat{\xi_i}\}_D) = \underset{\substack{\alpha_i \geq 0 \\ i \in [n]}}{\arg\max} \underset{\mathbf{w},\{\xi_i\}}{\arg\min} \ \mathcal{L}(\mathbf{w}, \{\xi_i\}, \boldsymbol{\alpha})$$

Derivation:
Differentiating (1) w.r.t. $\mathbf{w}$:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{w}} &= 2\mathbf{w} - \sum_{i=1}^{n} \alpha_i y^i \mathbf{x}^i \\ &= 0 \\ \implies \mathbf{w} &= \frac{1}{2} \sum_{i=1}^{n} \alpha_i y^i \mathbf{x}^i \end{aligned} \tag{2}$$

Substituting (2) in (1):

$$\begin{aligned} \mathcal{L}(\{\xi_i\}, \boldsymbol{\alpha}) &= \frac{1}{4} \sum_{i=1}^{n} \sum_{j=1}^{n} y^i y^j \alpha_i \alpha_j \langle \mathbf{x}^i, \mathbf{x}^j \rangle + \sum_{i=1}^{n} \left(\xi_i^2 + \alpha_i (1 - \xi_i) \right) - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y^i y^j \alpha_i \alpha_j \langle \mathbf{x}^i, \mathbf{x}^j \rangle \\ &= -\frac{1}{4} \sum_{i=1}^{n} \sum_{j=1}^{n} y^i y^j \alpha_i \alpha_j \langle \mathbf{x}^i, \mathbf{x}^j \rangle + \sum_{i=1}^{n} \left(\xi_i^2 + \alpha_i (1 - \xi_i) \right) \end{aligned} \tag{3}$$

$\forall i \in [n]$, Differentiating (1) w.r.t ξ_i:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \xi_i} &= 2\xi_i - \alpha_i \\ &= 0 \\ \implies \xi_i &= \frac{\alpha_i}{2} \end{aligned} \tag{4}$$

Substituting (4) in (3):

$$\begin{aligned} \mathcal{L}(\boldsymbol{\alpha}) &= -\frac{1}{4} \sum_{i=1}^{n} \sum_{j=1}^{n} y^i y^j \alpha_i \alpha_j \langle \mathbf{x}^i, \mathbf{x}^j \rangle + \sum_{i=1}^{n} \left(\left(\frac{\alpha_i}{2} \right)^2 + \alpha_i \left(1 - \frac{\alpha_i}{2} \right) \right) \\ &= -\frac{1}{4} \sum_{i=1}^{n} \sum_{j=1}^{n} y^i y^j \alpha_i \alpha_j \langle \mathbf{x}^i, \mathbf{x}^j \rangle + \sum_{i=1}^{n} \left(\alpha_i - \frac{\alpha_i^2}{4} \right) \end{aligned} \tag{5}$$

Thus the Lagrangian Dual Problem for $(P1)$ is:

$$\underset{\substack{\alpha_i \geq 0 \\ i \in [n]}}{\arg\min} \ \mathcal{L}(\boldsymbol{\alpha}) = \underset{\substack{\alpha_i \geq 0 \\ i \in [n]}}{\arg\min} \left(-\frac{1}{4} \sum_{i=1}^{n} \sum_{j=1}^{n} y^i y^j \alpha_i \alpha_j \langle \mathbf{x}^i, \mathbf{x}^j \rangle + \sum_{i=1}^{n} \left(\alpha_i - \frac{\alpha_i^2}{4} \right) \right)$$

Part 4

Differences are:

1. In the original SVM problem, α_i's are restricted between 0 and C, this restriction arises because of removing the variables β_i's. But in this problem, there is no such restriction present.
2. In this problem, there is a $\sum_i \alpha_i^2$ term that arises due to squaring of slack ariable ξ in SVM equation. But there is no such term present in the original SVM problem.

v

Part 5

No, the positivity constraints $\xi_i \geq 0$ are **not** vacuous for the original SVM problem.
This is because in the original SVM problem, the expression to be minimized has the term $\sum \xi_i$, thus negative and positive value combination for ξ_i can help minimize the expression. Thus the positivity constraint needs to be *explicitly* mentioned.
For the given SVM problem, since the expression to be minimized has the term $\sum \xi_i^2$, thus positive and negative value combination of xi_i will give the same expression (since sign gets lost while squaring), hence the positivity constraint need *not* be *explicitly* mentioned.

Part 3

For the GD solver, the averaged iterate (Accuracy=71.95%) gave a better performance than the current iterate (Accuracy=68.69%). Also, the objective value $f(\mathbf{w})$ achieved a lesser value in case of averaged iterate (As seen from the graph given below).

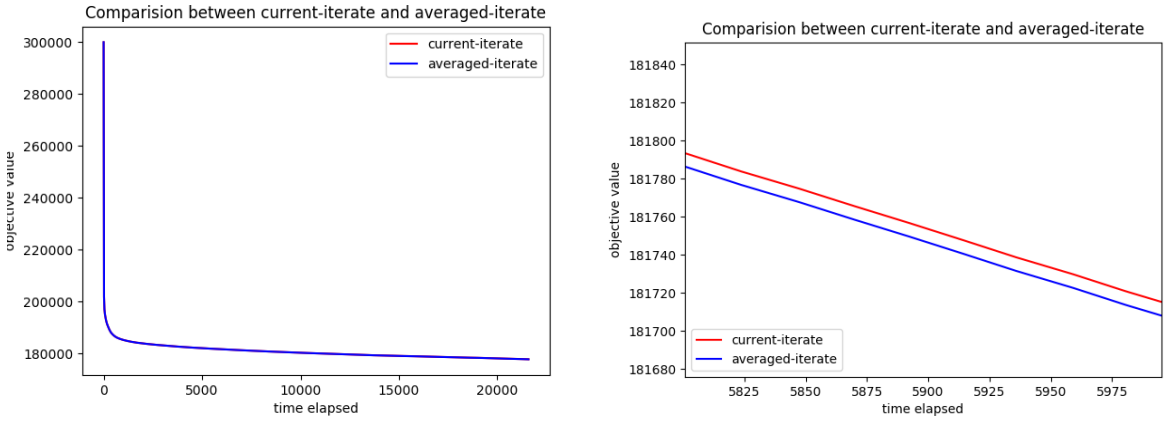


Figure 1: the figure on the left shows the comparison of performance between the current iterate model and the averaged iterate model. The figure on the right shows the zoomed in version of the graph.

Part 4

I assumed the step length to be proportional to $\frac{1}{\sqrt{t}}$ (as mentioned in the skeleton code).
Then for the dependency on n, I tried multiplying different functions: $1, n, \frac{1}{n}, \frac{1}{n}, \frac{1}{n^2}, \frac{1}{\sqrt{n}}$.
Out of all these, $\frac{1}{n}$ was found to be the most efficient.
Then I tried multiplying different constants to the obtained function (such as 1,2,3,10,11,12).
Out of them, 2 came out to be the most optimal.
Thus, I followed the hit-and-trial strategy to obtain the optimal step-length for GD and got:

$$\eta = 2 \times \frac{1}{n} \times \frac{1}{\sqrt{t}}$$

Part 5

From the graph, we can see that the GD algorithm reduces the objective function faster than the SCD algorithm (for which high fluctuations are observed).

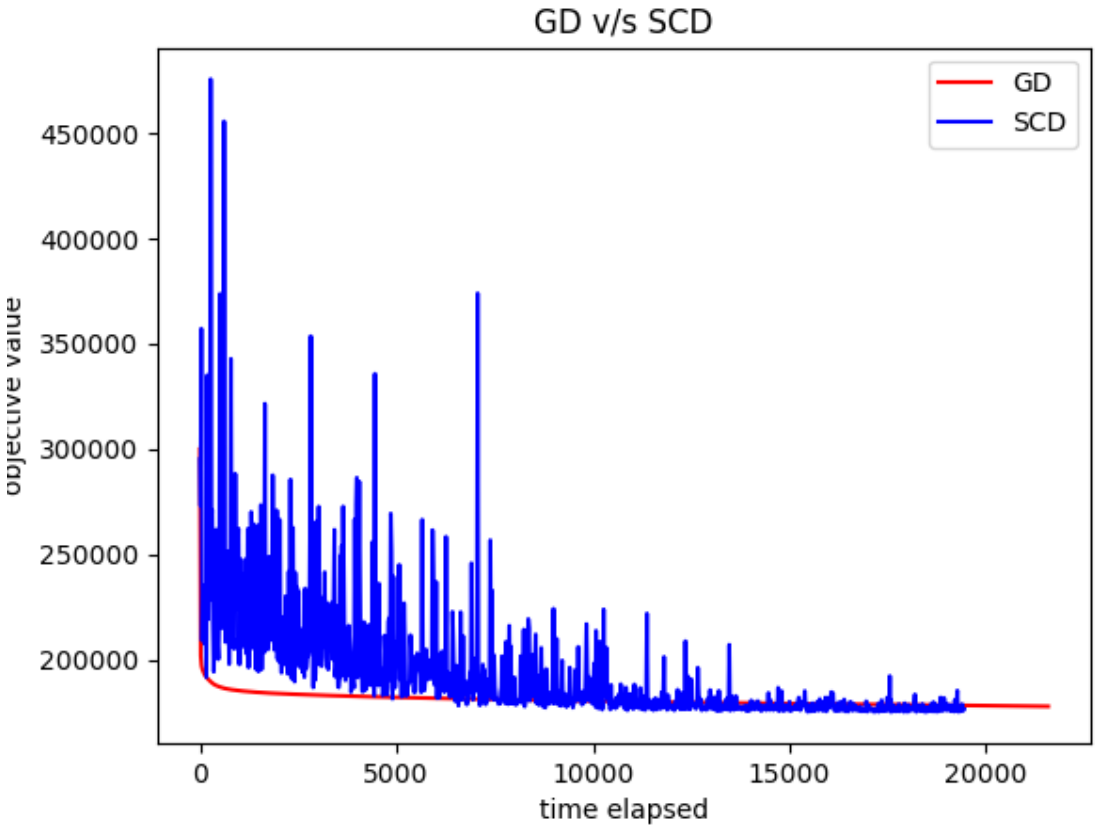


Figure 2: Comparison of performance between GD and SCD algorithms.

Part 6

From the graph, we can see that the *theoretical* performance of SCD algorithmn is WAY BETTER than that of the GD algorithm (SCD algorithm finishes even before $2 \times (spacing)$ iterations of the GD algorithm!!).

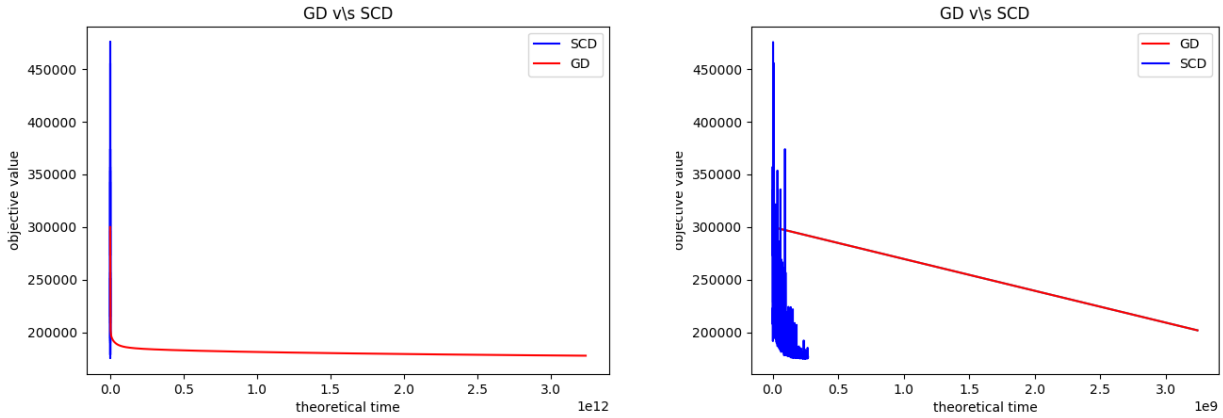


Figure 3: the figure on the left shows the comparison of *theoretical* performance between the GD and SCD algorithms. The figure on the right shows the graph after taking only 2 points for the GD plot.