ML Exam-2

Name: Vyom Pathak
I.D.: 96703101

Q1 (a) It is important to reserve some images for the test dataset in order to evaluate the performance of the model on unseen data. As a whole, we do this to check the generalization capability of the model.

(b) One possible explanation that the model performs well on the training and poorly on the test set is that the model is overfitting on the training dataset i.e. model shows high variance.

Q2 (a) False (b) False (c) True
(d) False (e) False

(Q5) → Considering n vectors:

$\phi_1, \phi_2, \ldots, \phi_n$.

→ Some of the vectors are pre-assigned to a specific group.

→ The first step of k-means algorithm is partitioning of the vectors in k groups.

→ We assign each ϕ_i to a group with the nearest representation value.

→ Now, we overwrite this assignment of the vectors which are pre-assignment to a specific group.

→ This helps in faster convergence.

→ Eg: Let there be patients with m medical features.

→ For some of these patients, the diagnosis is confirmed irrespective of their medical features, hence we already know the corresponding group is already known based on the diagnosis.

→ Now, the first step of the K-means algorithm will be to partition the patients in K-groups with O_i as the features having close representation of a group classified by a disease.

→ After this, the patients with definitive diagnosis are over-written and are assigned to their pre-set diagnosed disease irrespective of their features.

→ (Q6) (a) SVD - We use SVD instead of SVDS because the size of the data matrix is small (100x500) as well as the dataset is described as dense.

→ It returns the singular values of Φ in descending order.

→ Firstly we will center the given data matrix :

```
→ n = size(Φ, 1);
→ center = mean(Φ, 1);
→ Φ = X - repmat(center, n, 1);
```

→ Then, we will perform SVD to obtain V as the projections

$\Rightarrow [U, S, \Theta] = \text{svd}(\underline{\underline{\Phi}})$;

$\Rightarrow$ Then we use the projections to find the embedding matrix Y.

$\Rightarrow Y = \underline{\underline{\Phi}}\,\Theta(:, 1:10)$;

(b) SVDS - We use SVDS instead of SVD because the size of the data matrix is large ($10^5 \times 10^6$) and the dataset is described as sparse (10^7 non-zeros). The function SVDS performs faster calculations under these conditions.

→ The steps are similar to the above we just use the svds() function instead of svd().

→ Firstly we will center the given data matrix:

⇒ n = size(X,1);

⇒ center = mean(X,1);

⇒ $\tilde{X}$ = X - repmat(center,n,1);

→ Then, we will perform SVD to obtain V as the projections

⇒ [U,S,V] = svds($\tilde{X}$,100);

→ Then we use the projections to find the embedding matrix Y.

$\Rightarrow \psi = \tilde{\Phi}\theta$;

(Q5) We need to prove the given inequality using the given assumption

PGD is given as:

$$\theta^{(t+1)} = \text{Prox}_{\gamma(t)\lambda r}(\theta^{(t)} - \gamma^{(t)} \nabla L(\theta^{(t)}))$$

→ We define $G_\gamma(\theta)$ which is a gradient at iteration t,

$$G_\gamma(\theta) \triangleq \frac{1}{\gamma}\left(\theta - \text{Prox}_{\gamma\lambda r}(\theta - \gamma \nabla L(\theta))\right)$$

→ At every iteration,

$$\theta^{t+1} = \theta^t - \gamma^{(t)} G_{\gamma(t)}(\theta^t)$$

→ At optimum, $G_r(\theta^t) = 0$.

→ We require that the gradient $L(\cdot)$ is Lipschitz continous (which is also given)

→ Now we use the Central Lemma & the Descent Lemma (which is also given).

→ By convexity of $L(\cdot)$ we have,

$$L(z) \geq L(\theta) + \langle \nabla L(\theta), z - \theta \rangle.$$

$\theta = \theta^t$ & $\theta' = \theta^{t+1}$. When we combine this with the Descent Lemma we get the following:

$$L(\theta') \leq L(z) - \langle \nabla L(\theta), z - \theta \rangle + \langle \nabla L(\theta), \theta' - \theta \rangle + \frac{M}{2} \|\theta' - \theta\|$$

where $L(z) = L(\theta) + \lambda x(\theta^{t})$

$\Rightarrow$ This simplifies to the following:

$$L(\theta^{(t+1)}) + \lambda r(\theta^{(t+1)}) \leq L(\theta) + \lambda r(\theta)$$
$$+ \frac{1}{r^{(t)}} (\theta^{(t)} - \theta^{(t+1)})^T (\theta^{(t+1)} - \theta)$$
$$+ \frac{M}{2} \| \theta^{(t)} - \theta^{(t+1)} \|^2$$

(Q3) (a) Yes, Huber loss is differentiable

$$l(\varepsilon) = \begin{cases} \varepsilon^2/2 & |\varepsilon| \leq a \\ a(|\varepsilon| - a) & |\varepsilon| > a \end{cases}$$

$$= \begin{cases} \varepsilon^2/2 & |\varepsilon| \leq a \\ a(\varepsilon - a) & |\varepsilon| > a \;\&\; |\varepsilon| > 0 \\ a(-\varepsilon - a) & |\varepsilon| > a \;\&\; |\varepsilon| < 0 \end{cases}$$

$$\nabla l(\varepsilon) = \begin{cases} \varepsilon & |\varepsilon| \leq a \\ a & \varepsilon > a \\ -a & \varepsilon < -a \end{cases}$$

$$g = \nabla\left(\lambda(\phi_i^T\theta - y_i)\right) = \begin{cases} \phi_i^T(\phi_i^T\theta - y_i) & |\varepsilon| \le a \\ a & \varepsilon > a \\ -a & \varepsilon < -a \end{cases}$$

(b) Algo. for PGD,

Initialize $\theta^{(0)}$, V (step size)

for $t = 1, \ldots,$ do

  for $i = 1, \ldots,$ do

    compute $\theta_c = \text{argmin}_i\left(\lambda(\phi_{ci}^T\theta_j - y_i)\right)$

  end for

$\theta_c^{(t+1)} \leftarrow \theta^{(t)} + \gamma(a)$

$\forall |\theta_c| > a \; \varepsilon | \theta_c > 0$

$\theta_i^{(t+1)} \leftarrow \theta^{(t)} - \gamma(a)$

$\forall |\theta_c| > a \; \varepsilon | \theta_c < 0$

for $i = 1$ to m

  $v = \|\theta_i\|$

  if $v \le \beta\lambda$,

    $\theta_i^{(t+1)} = 0$

else
$\theta_j^{(t+1)} \leftarrow \left(1 - \frac{r\lambda}{v}\right) \theta_j^{(t+1)}$
endif
end for

(c) Stochastic PGD,
$\theta^{(0)}$, V (step size)

$g_i \in \nabla l(\theta)$

$\theta_i \leftarrow \arg\min_{\theta} \left(f(\theta_i) + \frac{1}{2} \|\theta_i - \theta_i^{(t)} - r g_i\| \right)$

for $j = 1$ to m
$v \leftarrow |\theta_j|$
if $v \le r\lambda$ then $\theta_j^{(t+1)} = 0$
else $\theta_j^{(t+1)} \leftarrow \left(1 - \frac{r\lambda}{v}\right) \theta_j^{(t+1)}$
endif
end for