Howework 1

Lingyu Zhou, Chang Chen

2024.2.13

1.

i.

a.

Shuffle two datasets and randomly pick 70% from both it as training dataset, 20% as validating and 10 % as testing dataset.

b.

For each database, randomly split 70% as training, 20% as validating and 10 % as testing dataset. And mix the corresponding dataset together. This prevents the model based only or mainly on high school student's scripts or employee's scripts:
Training (70 % total): 50% students' 50% employees' randomly
Validation (20 % total): 50% students' 50% employees' randomly
Testing (10 % total): 50% students' 50% employees' randomly

ii.

Process the image (contrastness, etc.) to make it black and white. Therefore 0 means white pixel, 1 means black pixel and everything between 0 and 1 means different shades of grey. Then make them into a matrix, with the size same as the resolution of the original image ($\mathbb{R}^{28\times 28}$). The output would be a vector $O \in \mathbb{R}^{10\times 1}$ s.t. O_{i+1}=1 if it is predicted as number i and 0 o/w. E.g. predicted as digit 7 would be $[0,0,0,0,0,0,0,1,0,0]^T$ (first index is digit 0)
To evaluate the accuracy, suppose we have label vectors L's, the accuracy is defined by:

$$1-\frac{1}{70000}\sum_{O,L}||O-L||^2$$

iii.

Split all the post contents 7 : 2 : 1 based on time (no overlap and chronologically), to make it use for train, validation and test set, correspondingly. Then make a vector I that represents the existence of words that appear in certain posts from a word list containing all words that appeared in the post, like {a,`apple`,...,`zebra`}, so that $I_1 = [0,1,\cdots,0]^T$ means post only contains the word `apple`. And the output set is just 0 or 1. First fit the model with training dataset. Then according to the results from validation dataset, fine-tune the hyperparameters in the model. In the really final stage, after the model reaches a reasonable accuracy, test the data to get the accuracy ($\frac{\#\text{right}}{\#\text{wrong}}$)

2

i.

Since $\mathscr{L}_{sq}(h) = \frac{1}{n}\sum_{i=1}^{n}(h(\mathbf{x}_i)-y_i)^2$, and we are calculating for $h(\cdot)$, therefore minimum loss occurs at when the partial differentiation of $h(x_i)$ equal to 0:

$$\frac{\partial(\mathscr{L}_{sq}(h))}{\partial h(x_i)} = \frac{\partial}{\partial(h(x_i))}\frac{1}{n}\sum_{i=1}^{n}(h(\mathbf{x}_i)-y_i)^2 = \frac{1}{n}\frac{\partial}{\partial(h(x_i))}\sum_{i=1}^{n}h(x_i)^2 - 2h(x_i)y_i + y_i^2$$
$$= \frac{1}{n}\sum_{i=1}^{n}2h(x_i) - 2y_i = 0$$
$$= 2\sum_{i=1}^{n}h(x_i) - 2\sum_{i=1}^{n}2y_i = 0$$
$$\sum_{i=1}^{n}h(x_i) = \sum_{i=1}^{n}y_i$$
$$nh(x_i) = \sum_{i=1}^{n}y_i \text{(cause we are solve for a general } m\text{, namely } h(\cdot))$$
$$m = h(x_i) = \frac{1}{n}\sum_{i=1}^{n}y_i$$

□

ii.

Claim:

$$\frac{\partial|h(x_i)-y_i|}{\partial h(x_i)} = \text{sgn}(h(x_i)-y_i)$$

Proof:
$\forall h(x_i) : |h(x_i)-y_i|$, if $|h(x_i)-y_i| > 0$, then $\frac{\partial|h(x_i)-y_i|}{\partial h(x_i)} = \frac{\partial h(x_i)-y_i}{\partial h(x_i)} = 1$ and if $|h(x_i)-y_i| < 0$, then $\frac{\partial|h(x_i)-y_i|}{\partial h(x_i)} = \frac{\partial -h(x_i)+y_i}{\partial h(x_i)} = -1$
□
Therefore min. of abs. loss function occurs at $\frac{\partial\mathscr{L}_{abs}(h)}{\partial h(\cdot)} = 0$:

$$\frac{\partial(\mathscr{L}_{abs}(h))}{\partial h(x_i)} = \frac{\partial}{\partial(h(x_i))}\frac{1}{n}\sum_{i=1}^{n}|h(x_i)-y_i|$$
$$= \frac{1}{n}\sum_{i=1}^{n}\text{sgn}(h(x_i)-y_i) := 0$$
$$\Rightarrow \sum_{i=1}^{n}\text{sgn}(h(x_i)-y_i) := 0,$$

which means that $\forall i, h(x_i)-y_i$ must have equal number of positive and negative results, with the max. number of $\frac{n+1}{2}$, which is exactly the definition of median.
□

iii.

Since $\mathscr{L}_{bin}(h) = \frac{1}{n}\sum_{i=1}^{n}\delta(h(x_i)\neq y_i)$, it is actually measuring the proportion of the wrong prediction over n data. Therefore, if we need to choose a general $h_m(\cdot)$ with a fixed value, then we need to choose the correct answer that appears most frequently in $\{y_i|i\in[1,n]\}$, namely the mode. Because it's obvious that $\forall\{y_i|i\in[1,n]\} \ni m'\neq m = \text{mode}(\{y_i|i\in[1,n]\}) : |m'|m'\neq y_i| > |m|m\neq y_i|$, therefore, $\forall i$:

$$\sum_{i=1}^{n}\delta(m'\neq y_i) > \sum_{i=1}^{n}\delta(m\neq y_i)$$
$$\frac{1}{n}\sum_{i=1}^{n}\delta(m'\neq y_i) > \frac{1}{n}\sum_{i=1}^{n}\delta(m\neq y_i) (\because n>0)$$
$$\mathscr{L}_{bin}(h'_m) > \mathscr{L}_{bin}(h_m), \forall m'$$

Hence proved that mode minimizes the binary loss function.
□

3

i.

No because $\exists y \vee x \vee h$ s.t. $h(x) - y < 0$

ii.

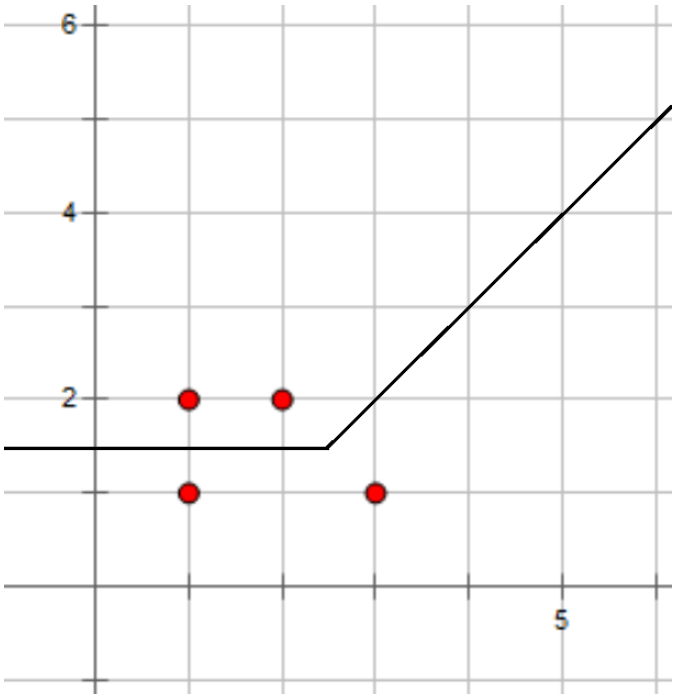
Continuity: Yes since log function is smooth over $\mathbb{R}$
Differentiability: Yes, similar to continuity
Non-negative: Yes since $\log(x) < 0$ only for $x < 1$ but $\cosh(x) \geq 1, \forall x$
Identity:
$\Rightarrow \because \log(\cosh(h(x) - y)) = 0 \therefore \cosh(h(x) - y) = 1 \therefore h(x) - y = 0 \therefore h(x) = y$
$\Leftarrow \because h(x) = y \therefore h(x) - y = 0 \therefore \log(\cosh(h(x) - y)) = 0$
Therefore yes.

iii.

No, since $\forall h(x) - y = k\pi \neq 0, k \in \mathbb{Z}, \sin(h(x) - y) = 0$

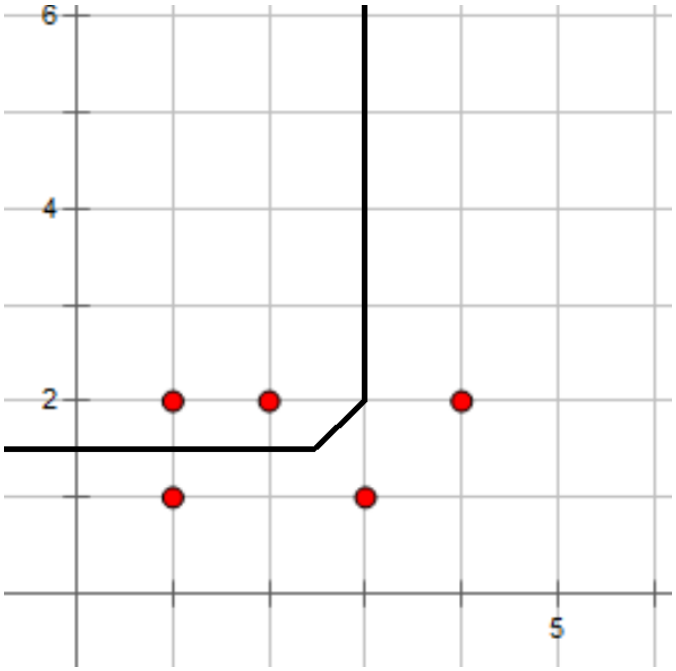
4.

i.



(Turning point is $(2.5, 1.5)$)

ii.



(Turning point are (2.5,1.5) and (3,2))
The new point falls into the positive side. The boundary changes because a new tie situation occurs so that we need to redraw the boundary around that new point.

iii.

No since the closest point to (2,400) is not (2,300), but (4,400), which is positive.
If we scale it the points become:
Positive: (0.8,0.4),(0.8,0.8),(1,0.8)
Negative: (0.2,0.2),(0.4,0.6), where the scaled target becomes (0.4,0.8), with its closest point being (0.4,0.6), which is negative. So, **yes**.

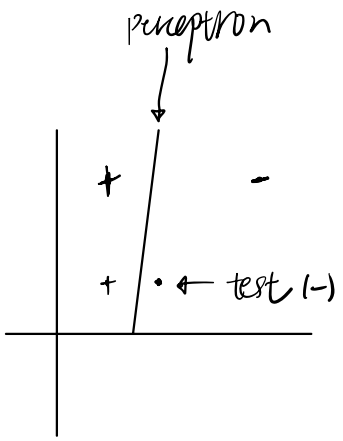
iv.

Testing, when there are millions of points.
During training, the KNN algorithm just stores the existing data points without building any complicated models or doing time-consuming calculations. This process is relatively quick, even with millions of points stored in RAM. However, during testing, it will apply KNN. It compares the new point to every data point by calculating the distance between them, and then sorting them to get the closest K ones in the training set to find the K nearest neighbors, which is time-consuming and still has a minimum time complexity of $O(n\log(n))$.

5(i)

Train: $(1,1)^{+}$, $(1,3)^{+}$, $(4,3)^{-}$

Test: $(2,1)^{-}$



$(1,1) \Rightarrow +1 \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} = 0$

$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$

$(1,3) \Rightarrow +1 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 3 & 1 \end{bmatrix} = 5$

$(4,3) \Rightarrow -1 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 4 & 3 & 1 \end{bmatrix} = -8$

$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 4 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} -3 \\ -2 \\ 0 \end{bmatrix}$

$\begin{bmatrix} -2 \\ -1 \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} -1 \\ 0 \\ 2 \end{bmatrix}$

$-1 \begin{bmatrix} -3 \\ -2 \\ 0 \end{bmatrix} \begin{bmatrix} 2 & 1 & 1 \end{bmatrix} = 8$

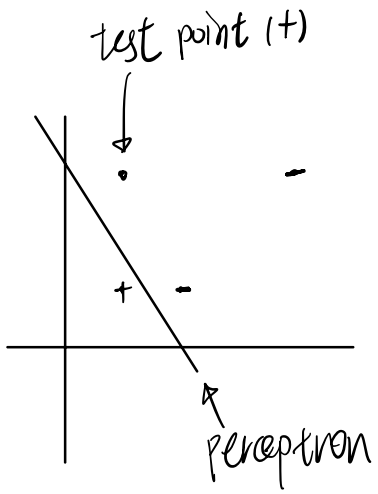
Test Accuracy

$\therefore$ Percep > kNN

5(ii)

Train: $(1,1)^+$, $(2,1)^-$, $(4,3)^-$

Test: $(1,3)^+$



$(1,1) \Rightarrow + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} = 0$

$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$

$(2,1) \Rightarrow - \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 1 \end{bmatrix} = -4$

$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}$

$(4,3) \Rightarrow - \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 4 & 3 & 1 \end{bmatrix} = -4$

$\begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 3 & 1 \end{bmatrix} = -1$

Test Accuracy

$\therefore$ Percep < KNN