CS 771A: Intro to Machine Learning, IIT Kanpur				**Endsem Exam** (18 Nov 2025)
Name	MELBO			**40 marks**
Roll No	250007	Dept.	AWSM	Page **1** of **4**

Instructions:
1. This question paper contains 2 pages (4 sides of paper). Please verify.
2. Write your name, roll number, department in **block letters** with **ink** on **each page**.
3. Write your final answers neatly **with a blue/black pen**. Pencil marks may get smudged.
4. Don't overwrite/scratch answers especially in MCQ – ambiguous cases will get 0 marks.

Q1 (Total Confusion) Melbu learnt a linear model to solve a binary classification problem with two classes $-1, 1$ as $\text{sign}(\mathbf{w}^\top\mathbf{x} + b)$ with $\mathbf{w} \in \mathbb{R}^{100}$ and $b \in \mathbb{R}$. The classifier was evaluated on 10 test data points $(\mathbf{x}^i, y^i)$ that gave the confusion matrix on the right. y is the true label of a test point and $\hat{y}$ is the label predicted by the classifier. The entries in the matrix show how many points of a given class were classified in a certain way by the classifier (e.g. 7 points whose true label was $y = -1$ were (mis)predicted as $\hat{y} = 1$). Calculate the following quantities for the classifier based on its test performance (no derivations needed) **(4 x 0.5 + 2 + 2 = 6 marks)**

	$\hat{y} = 1$	$\hat{y} = -1$
$y = 1$	1	1
$y = -1$	7	1

Accuracy $\mathbb{P}[\hat{y} = y]$	$\frac{2}{10} = 0.2$	False discovery rate $\mathbb{P}[y \neq 1 \mid \hat{y} = 1]$	$\frac{7}{8} = 0.875$
False Omission Rate $\mathbb{P}[y = 1 \mid \hat{y} = -1]$	$\frac{1}{2} = 0.5$	Neg. predictive value $\mathbb{P}[\hat{y} = y \mid \hat{y} = -1]$	$\frac{1}{2} = 0.5$

The table below shows the classifier's prediction scores $\mathbf{w}^\top\mathbf{x}^i + b$ on the 10 test points. Melbu wants to change the model parameters $\mathbf{w}, b$ to improve the test accuracy on these 10 points. Retraining the model or changing the training algorithm is not allowed. The test feature vectors $\mathbf{x}^i \in \mathbb{R}^{100}, i \in [10]$ are not available either. All we are allowed to do is make simple changes directly to the model parameters $\mathbf{w}, b$ learnt by Melbu (e.g. scale or shift them). Help Melbu achieve this goal. What is the best test accuracy you get after the modifications? Briefly justify.

$\mathbf{w}^\top\mathbf{x}^i + b$	-3	-1	1	3	5	7	9	11	13	15
true label y	-1	$+1$	-1	-1	-1	-1	-1	-1	-1	$+1$

Test accuracy of modified classifier:
Give details of modifications below

at least 0.9 if using bias correction
at least 0.8 if using inversion

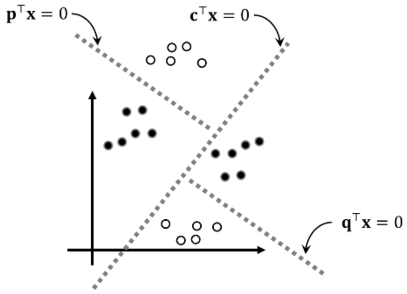
Method 1 (Inversion): A binary classifier that is mostly wrong is actually mostly right. By flipping all predictions by setting $\tilde{\mathbf{w}} = -\mathbf{w}, \tilde{b} = -b$, the resulting model achieves an accuracy of 0.8.

Method 2 (Bias correction): Notice that the model gives large (positive) weights to most points whereas most points are negatively labeled. This suggests that the model has a bias problem. Setting $\tilde{b} = b - 14$ and keeping $\mathbf{w}$ unchanged will result in the following prediction scores

$\mathbf{w}^\top\mathbf{x}^i + \tilde{b}$	-17	-15	-13	-11	-9	-7	-5	-3	-1	1
true label y	-1	$+1$	-1	-1	-1	-1	-1	-1	-1	$+1$

The resulting model achieves an accuracy of 0.9.

Q2 (Probabilistic DT) Melbo wants to solve a binary classification problem using two classification models $\mathbf{p}, \mathbf{q} \in \mathbb{R}^d$. A classifier $\mathbf{c} \in \mathbb{R}^d$ decides which model to use at test time (see figure). For a point $\mathbf{x} \in \mathbb{R}^d$, if $\mathbf{c}^\top \mathbf{x} \geq 0$, Melbo will predict $\text{sign}(\mathbf{p}^\top \mathbf{x})$. If $\mathbf{c}^\top \mathbf{x} < 0$, predict $\text{sign}(\mathbf{q}^\top \mathbf{x})$. Bias terms are hidden inside the models. Note that this is simply a decision tree with one root and two leaves. **(4 x 4 = 16 marks)**



Melbo has train data $(\mathbf{x}^i, y^i), i \in [N]$ with $\mathbf{x}^i \in \mathbb{R}^d, y^i \in \mathbb{R}$ but doesn't know which model, $\mathbf{p}$ or $\mathbf{q}$, should handle which point, so Melba advises using latent variables. For each point $i \in [N]$, Melbo uses latent variables $z^i \in \{-1, +1\}$, a naïve prior $\mathbb{P}[z^i \mid \mathbf{p}, \mathbf{q}, \mathbf{c}] = 0.5$ and (conditional) likelihood functions $\mathbb{P}[z^i \mid \mathbf{x}^i, \mathbf{p}, \mathbf{q}, \mathbf{c}] = \sigma(z^i \cdot \mathbf{c}^\top \mathbf{x}^i)$ and $\mathbb{P}[y^i \mid z^i = +1, \mathbf{x}^i, \mathbf{p}, \mathbf{q}, \mathbf{c}] = \sigma(y^i \cdot \mathbf{p}^\top \mathbf{x}^i)$ and $\mathbb{P}[y^i \mid z^i = -1, \mathbf{x}^i, \mathbf{p}, \mathbf{q}, \mathbf{c}] = \sigma(y^i \cdot \mathbf{q}^\top \mathbf{x}^i)$, where $\sigma(t) \stackrel{\text{def}}{=} \frac{1}{(1+\exp(-t))}$ is the sigmoid function.

Derive an expression for total likelihood $\mathbb{P}[y^i \mid \mathbf{x}^i, \mathbf{p}, \mathbf{q}, \mathbf{c}]$ in terms of $y^i, \mathbf{x}^i, \mathbf{p}, \mathbf{q}, \mathbf{c}$ (no z^i allowed).

The law of total probability gives us

$$\mathbb{P}[y^i \mid \mathbf{x}^i, \mathbf{p}, \mathbf{q}, \mathbf{c}] = \sum_{z^i \in \{-1,+1\}} \mathbb{P}[y^i \wedge z^i \mid \mathbf{x}^i, \mathbf{p}, \mathbf{q}, \mathbf{c}]$$

$$= \sum_{z \in \{-1,+1\}} \mathbb{P}[y^i \mid \mathbf{x}^i, z, \mathbf{p}, \mathbf{q}, \mathbf{c}] \cdot \mathbb{P}[z \mid \mathbf{x}^i, \mathbf{p}, \mathbf{q}, \mathbf{c}]$$

$$= \left(\sigma(\mathbf{c}^\top \mathbf{x}^i) \cdot \sigma(y^i \cdot \mathbf{p}^\top \mathbf{x}^i) + \left(1 - \sigma(\mathbf{c}^\top \mathbf{x}^i)\right) \cdot \sigma(y^i \cdot \mathbf{q}^\top \mathbf{x}^i)\right)$$

As exact MLE is hard, Melbo instead tries to solve $\underset{\mathbf{p},\mathbf{q}}{\arg\max}\, \underset{\{z^i\}}{\arg\max}\, \underset{\mathbf{c}}{\arg\max} \left\{\mathcal{L}(\mathbf{p}, \mathbf{q}, \mathbf{c}, \{z^i\})\right\}$ with $\mathcal{L}(\mathbf{p}, \mathbf{q}, \mathbf{c}, \{z^i\}) \stackrel{\text{def}}{=} \sum_{i \in [N]} \ln(\mathbb{P}[y^i \mid z^i, \mathbf{x}^i, \mathbf{p}, \mathbf{q}, \mathbf{c}]) + \sum_{i \in [N]} \ln(\mathbb{P}[z^i \mid \mathbf{x}^i, \mathbf{p}, \mathbf{q}, \mathbf{c}])$ using alternating optimization. **You are free to use simple operations like least squares, logistic regression directly**.

Step 1: Freeze $\mathbf{c}, \{z^i\}$ and give brief derivation on how to find $\arg\max_{\mathbf{p},\mathbf{q}} \mathcal{L}(\mathbf{p}, \mathbf{q}, \mathbf{c}, \{z^i\})$.

The second set of terms in $\mathcal{L}(\mathbf{p}, \mathbf{q}, \mathbf{c}, \{z^i\})$ i.e. $\sum_{i \in [N]} \ln(\mathbb{P}[z^i \mid \mathbf{x}^i, \mathbf{p}, \mathbf{q}, \mathbf{c}])$ do not participate here as these do not depend on $\mathbf{p}, \mathbf{q}$. For the first set of terms, the summation can be broken into terms that depend on $\mathbf{p}$ and that depend on $\mathbf{q}$ to give (after taking NLL)

$$\arg\max_{\mathbf{p},\mathbf{q}} \mathcal{L}(\mathbf{p}, \mathbf{q}, \mathbf{c}, \{z^i\}) = \arg\min_{\mathbf{p},\mathbf{q}} \sum_{i \in [N]} -\ln(\mathbb{P}[y^i \mid z^i, \mathbf{x}^i, \mathbf{p}, \mathbf{q}, \mathbf{c}])$$

$$= \arg\min_{\mathbf{p}} \sum_{z^i = +1} \ln(1 + \exp(-y^i \cdot \mathbf{p}^\top \mathbf{x}^i)) + \arg\min_{\mathbf{q}} \sum_{z^i = -1} \ln(1 + \exp(-y^i \cdot \mathbf{q}^\top \mathbf{x}^i))$$

The above can be solved by invoking a logistic regression solver twice, once over points with $z^i = +1$ to give us $\mathbf{p}$ and another time over points with $z^i = -1$ to give us $\mathbf{q}$.

Step 2: Freeze $\mathbf{p}, \mathbf{q}, \mathbf{c}$ and give brief derivation on how to find $\arg\max_{\{z^i\}} \mathcal{L}(\mathbf{p}, \mathbf{q}, \mathbf{c}, \{z^i\})$.

We show the process for a single z^i which can be repeated for all others.

$$\arg\max_{z^i} \mathcal{L}(\mathbf{p}, \mathbf{q}, \mathbf{c}, z^i) = \arg\min_{z^i} \{-\ln(\mathbb{P}[y^i \mid z^i, \mathbf{x}^i, \mathbf{p}, \mathbf{q}, \mathbf{c}]) - \ln(\mathbb{P}[z^i \mid \mathbf{x}^i, \mathbf{p}, \mathbf{q}, \mathbf{c}])\}$$

For $z^i = +1$, the objective value is $V_i^+ = \ln(1 + \exp(-y^i \cdot \mathbf{p}^\top \mathbf{x}^i)) + \ln(1 + \exp(-\mathbf{c}^\top \mathbf{x}^i))$

For $z^i = -1$, the objective value is $V_i^- = \ln(1 + \exp(-y^i \cdot \mathbf{q}^\top \mathbf{x}^i)) + \ln(1 + \exp(\mathbf{c}^\top \mathbf{x}^i))$

Thus, the optimal value of $z_i = \text{sign}(V_i^- - V_i^+)$ with ties broken arbitrarily.

Note: unlike the mixed regression case we discussed in lectures, where we chose the latent variable simply by looking at which model gave us smaller regression error, the choice here also must take care that the chosen latent variable value is easy to learn by the classifier $\mathbf{c}$.

Step 3: Freeze $\mathbf{p}, \mathbf{q}, \{z^i\}$ and give brief derivation on how to find $\arg\max_{\mathbf{c}} \mathcal{L}(\mathbf{p}, \mathbf{q}, \mathbf{c}, \{z^i\})$.

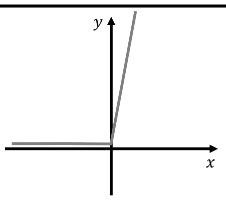
The first set of terms in $\mathcal{L}(\mathbf{p}, \mathbf{q}, \mathbf{c}, \{z^i\})$ i.e. $\sum_{i \in [N]} \ln(\mathbb{P}[y^i \mid z^i, \mathbf{x}^i, \mathbf{p}, \mathbf{q}, \mathbf{c}])$ do not participate here as these do not depend on $\mathbf{c}$. For the second set of terms, we get (after taking NLL)

$$\arg\max_{\mathbf{c}} \mathcal{L}(\mathbf{p}, \mathbf{q}, \mathbf{c}, \{z^i\}) = \arg\min_{\mathbf{c}} \sum_{i \in [N]} -\ln(\mathbb{P}[z^i \mid \mathbf{x}^i, \mathbf{p}, \mathbf{q}, \mathbf{c}])$$

$$= \arg\min_{\mathbf{c}} \sum_{i \in [N]} \ln(1 + \exp(-z^i \cdot \mathbf{c}^\top \mathbf{x}^i))$$

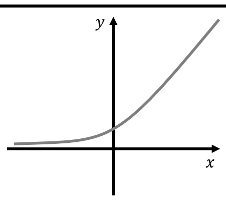
The above can be solved by invoking a logistic regression solver using z^i as the binary "labels".

Q3 (Rapidly Rising ReLUs) The ReLU activation becomes more expressive if used with a sharpness parameter B as $\rho(x; B) \overset{\text{def}}{=} \ln((1 + \exp(-B \cdot x)))$. For each of the following five curves, select the value of B that best generates that curve. **Shade only one circle in each part.** **(5 x 1 = 5 marks)**



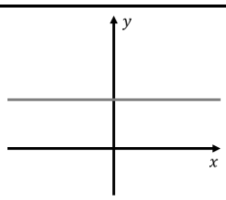
(a)

- ● $B \to -\infty$
- ○ $B = -1$
- ○ $B = 0$
- ○ $B = +1$
- ○ $B \to +\infty$



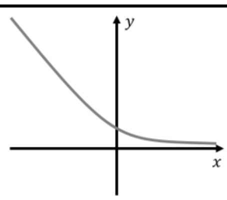
(b)

- ○ $B \to -\infty$
- ● $B = -1$
- ○ $B = 0$
- ○ $B = +1$
- ○ $B \to +\infty$



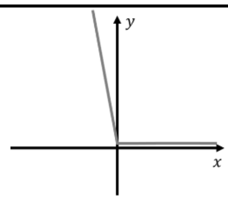
(c)

- ○ $B \to -\infty$
- ○ $B = -1$
- ● $B = 0$
- ○ $B = +1$
- ○ $B \to +\infty$



(d)

- ○ $B \to -\infty$
- ○ $B = -1$
- ○ $B = 0$
- ● $B = +1$
- ○ $B \to +\infty$



(e)

- ○ $B \to -\infty$
- ○ $B = -1$
- ○ $B = 0$
- ○ $B = +1$
- ● $B \to +\infty$

Q4 (The Perils of Pollution) Melba is studying the effect of factory output on pollution levels. It is known that if the factory output is p and the pollution probe is at distance q from the factory, then the pollution level measured by the probe is p/q. Melba conducted experiments with 4 output levels l, m, n, o and for each experiment, Melba placed the probe at 4 distances d, e, f, g from the factory, thus getting 16 readings. Melba recorded the readings in the above matrix but by mistake, Melba spilled coffee on the spreadsheet causing some of the entries (labeled x_{fl}, x_{gn} etc in gray) to get erased. Melba is in panic as not only did a lot of data get erased, but the values l, m, n, o, d, e, f, g used to conduct experiments are also gone. Help find these values so that Melba can repeat the experiment. It is known that l, m, n, o, d, e, f, g are all positive integers, $d + e + f + g = 50$ and $f < g$. Melba also recalls that if we arrange the readings in the first column as a vector $\mathbf{v} = [12, 30, x_{fl}, x_{gl}] \in \mathbb{R}^4$, then $\|\mathbf{v}\|_2^2 = 1169$, $\|\mathbf{v}\|_1 = 57$. Give brief derivation on how you obtained l, m, n, o, d, e, f, g. **(8 + 5 = 13 marks)**

	l	m	n	o
d	12	x_{dm}	x_{dn}	x_{do}
e	30	40.5	x_{en}	x_{eo}
f	x_{fl}	x_{fm}	1.25	x_{fo}
g	x_{gl}	x_{gm}	x_{gn}	4.5

$l = \boxed{120}$ $m = \boxed{162}$ $n = \boxed{15}$ $o = \boxed{108}$

$d = \boxed{10}$ $e = \boxed{4}$ $f = \boxed{12}$ $g = \boxed{24}$

Give brief derivation here

The constraints over the vector $\mathbf{v}$ tell us that $x_{fl}^2 + x_{gl}^2 = 125$ and $x_{fl} + x_{gl} = 15$. This tells us that $\{x_{fl}, x_{gl}\} = \{5, 10\}$. Since $f < g$ and readings are inversely proportional to the distance, we must have $x_{fl} > x_{gl}$ i.e. $x_{fl} = 10, x_{gl} = 5$. This tells us that $d^{-1}: e^{-1}: f^{1}: g^{-1} = 12: 30: 10: 5$ i.e. $d: e: f: g = 5: 2: 6: 12$. Using $d + e + f + g = 50$ gives us $d = 10, e = 4, f = 12, g = 24$.

Having obtained d, e, f, g, obtaining the l, m, n, o is simple since the matrix is a rank one matrix by construction. Direct calculations tell us that $l = 12 \times d = 120$. However, $l: m = 30: 40.5$ i.e. $m = 162$. Similarly, $l: n = x_{fl}: 1.25$ i.e. $n = 15$. Similarly, $l: o = x_{gl}: 4.5$ i.e. $o = 108$.