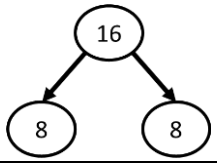
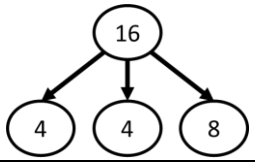
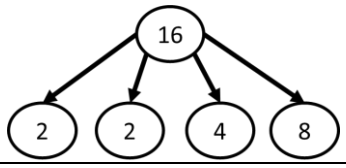
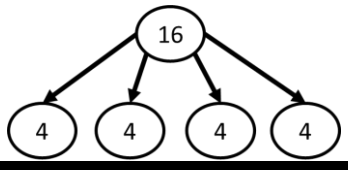
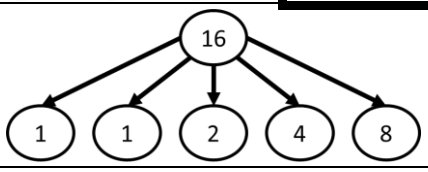
CS 771A: Intro to Machine Learning, IIT Kanpur				**Midsem Exam** (24 Sep 2022)
Name	Melbo			**40 marks** Page **1** of **4**
Roll No	000001	Dept.	AWSM	

Instructions:

1. This question paper contains 2 page (4 sides of paper). Please verify.
2. Write your name, roll number, department in **block letters** with **ink** on **each page**.
3. If you don't do this, your pages may get lost when we unstaple your paper to scan pages
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 – such cases will get straight 0 marks.

Q1. For the hangman problem, 5 decision tree splits at a node are given. For each split, write down the information gain (entropy reduction) in the bold border boxes border next to the diagrams as a single fraction or decimal number. Use logarithms with base 2 in the definition of entropy. The numbers written in the nodes indicate how many words reached that node. **(5 marks)**

	1.0 or 1		1.5 or 3/2
	1.75 or 7/4		2.0 or 2
		1.875 or 15/8	

Q2. (Intriguing entropy) For a random variable X with support $\{-1,1\}$ with $\mathbb{P}[X=1]=p$, define its entropy as $H(X) \stackrel{\text{def}}{=} -p\ln p-(1-p)\ln(1-p)$ (use natural logarithms for sake of simplicity). Find **(a)** a value of $p \in [0,1]$ where the entropy $H(X)$ is largest and **(b)** a value $p \in [0,1]$ where the entropy $H(X)$ is smallest. Show brief calculations/arguments for both parts. **(3 + 2 = 5 marks)**

$\frac{dH}{dp} = -1-\ln p+1+\ln(1-p) = \ln\left(\frac{1}{p}-1\right)$ which vanishes at $p=\frac{1}{2}$. Confirming with the second derivative tells us that $\frac{d^2H}{dp^2} = -\frac{1}{p(1-p)} < 0$ at $p=\frac{1}{2}$ confirming that it is a local maxima candidate. Since this value of p also lies in the feasible set $[0,1]$, we conclude that $H(X)$ is largest at $p=\frac{1}{2}$.

$H(X) \geq 0$ for all values of p and we see that for $p=0$ as well as for $p=1$, we get $H(X)=0$ since $1\ln 1 = 0 = 0\ln 0$. Since these values are also in the feasible set $[0,1]$ we conclude that $H(X)$ is the smallest at these values.

Q3. (At a loss for names) Consider the following loss function where $\tau > 0$ and $z \in \mathbb{R}$.

$$\ell_\tau(z) = \begin{cases} 1 - z & z < 1 - \tau \\ -\frac{(1-z)^4}{16\tau^3} + \frac{3(1-z)^2}{8\tau} + \frac{1-z}{2} + \frac{3\tau}{16} & z \in [1-\tau, 1+\tau] \\ 0 & z > 1 + \tau \end{cases}$$

1. Write down expressions for $\frac{d\ell_\tau(z)}{dz}$ and $\frac{d^2\ell_\tau(z)}{dz^2}$. No need to show calculations.
2. Write down an expression for $\nabla_{\mathbf{w}} f(\mathbf{w})$ where $\mathbf{w}, \mathbf{x}^i \in \mathbb{R}^d, y^i \in \{-1, +1\}$. You can use terms such as $\ell'_\tau(\cdot)$ in your expression to denote the first derivative of $\ell_\tau(\cdot)$ to avoid clutter.
$$f(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|_2^2 + \sum_{i=1}^{n} \ell_\tau\left(y^i \cdot \mathbf{w}^\top \mathbf{x}^i\right)$$
3. Write down an expression for what the loss function $\ell_\tau(\cdot)$ would look like as $\tau \to 0^+$.

Give brief calculations for the 2nd part and brief justification for the 3rd part. **(2+2+3+1 = 8 marks)**

$$\frac{d\ell_\tau(z)}{dz} = \begin{cases} -1 & z < 1 - \tau \\ \frac{(1-z)^3}{4\tau^3} - \frac{3(1-z)}{4\tau} - \frac{1}{2} & z \in [1-\tau, 1+\tau] \\ 0 & z > 1 + \tau \end{cases}$$

$$\frac{d^2\ell_\tau(z)}{dz^2} = \begin{cases} 0 & z < 1 - \tau \\ -\frac{3(1-z)^2}{4\tau^3} + \frac{3}{4\tau} & z \in [1-\tau, 1+\tau] \\ 0 & z > 1 + \tau \end{cases}$$

By applying the chain rule, we get

$$\nabla_{\mathbf{w}} f(\mathbf{w}) = \mathbf{w} + \sum_{i=1}^{n} \ell'_\tau\left(y^i \cdot \mathbf{w}^\top \mathbf{x}^i\right) \cdot y^i \cdot \mathbf{x}^i$$

As $\tau \to 0^+$ (i.e., approaches 0 from the right), the interval $[1-\tau, 1+\tau]$ vanishes as well as we have $1 - \tau \to 1^-$ and $1 + \tau \to 1^+$. Thus, the limiting behaviour of this function can be described as $\lim_{\tau \to 0^+} \ell_\tau(z) = [1 - z]_+$ i.e., it approaches the hinge loss function.

The loss function $\ell_\tau(\cdot)$ is a doubly differentiable version of the hinge loss first proposed by Kamalika Chaudhuri, Claire Monteleoni and Anand D. Sarwate in their 2011 JMLR paper entitled "*Differentially Private Empirical Risk Minimization*". Other such "smoothed" hinge loss variants also exist.

Q4. (A regularized median) Given a set of real numbers $a^1, a^2, \ldots, a^n \in \mathbb{R}$ (all are distinct but may be positive, negative or zero), we wish to find its "regularized median" by solving: $\min_x \frac{1}{2}x^2 + \sum_{i=1}^{n} |x - a^i|$. However, to design a solver, it would be helpful if we first rewrite this objective function as shown on the right-hand side by artificially introducing constraints

$$\begin{aligned} \min_{x, c^i} \quad & \frac{1}{2}x^2 + \sum_{i=1}^{n} c_i \\ & x - a^i \le c^i \\ \text{s.t.} \quad & x - a^i \ge -c^i \\ & c^i \ge 0 \end{aligned}$$

1. Write down the Lagrangian of this problem by introducing dual variables for the constraints.
2. Using the Lagrangian, create the dual problem (show brief derivation). Simplify the dual as much as you can otherwise the next part may get more cumbersome for you.
3. Give an expression for deriving the primal solution x in terms of the dual variables
4. Give peudocode for a coordinate ascent/descent method to solve the dual. Use any coordinate selection method you like. Give precise expressions in pseudocode on how you would process a chosen coordinate taking care of constraints. **(2 + 4 + 2 + 4 = 12 marks)**

Introducing dual variables $\alpha_i, \beta_i, \gamma_i$ for the 3 sets of constraints and compactly writing them and the a^i, c^i values as vectors gives the Lagrangian as (note that $\mathbf{1}$ denotes the all-ones vector)

$$\mathcal{L}(x, \mathbf{c}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}) = \frac{1}{2}x^2 + \mathbf{c}^\top\mathbf{1} + \boldsymbol{\alpha}^\top(x \cdot \mathbf{1} - \mathbf{a} - \mathbf{c}) - \boldsymbol{\beta}^\top(x \cdot \mathbf{1} - \mathbf{a} + \mathbf{c}) - \boldsymbol{\gamma}^\top\mathbf{c}$$

Setting $\frac{\partial \mathcal{L}}{\partial \mathbf{c}} = 0$ gives us $\mathbf{1} - \boldsymbol{\alpha} - \boldsymbol{\beta} - \boldsymbol{\gamma} = \mathbf{0}$ and therefore, $\boldsymbol{\alpha} + \boldsymbol{\beta} \leq \mathbf{1}$ (as $\boldsymbol{\gamma} \geq 0$)

Setting $\frac{\partial \mathcal{L}}{\partial x} = 0$ gives us $x + \boldsymbol{\alpha}^\top\mathbf{1} - \boldsymbol{\beta}^\top\mathbf{1} = 0$ i.e., $x = (\boldsymbol{\beta} - \boldsymbol{\alpha})^\top\mathbf{1} = \sum_{i-1}^{n}(\beta_i - \alpha_i)$

Putting these back gives us $\mathcal{L} = \frac{1}{2}x^2 + x \cdot (\boldsymbol{\alpha} - \boldsymbol{\beta})^\top\mathbf{1} - (\boldsymbol{\alpha} - \boldsymbol{\beta})^\top\mathbf{a} = -\frac{1}{2}x^2 - (\boldsymbol{\alpha} - \boldsymbol{\beta})^\top\mathbf{a}$

Inverting the sign (optional) gives us the dual problem

$$\min_{\substack{\boldsymbol{\alpha}\geq\mathbf{0},\boldsymbol{\beta}\geq\mathbf{0} \\ \boldsymbol{\alpha}+\boldsymbol{\beta}\leq\mathbf{1}}} \frac{1}{2}((\boldsymbol{\alpha} - \boldsymbol{\beta})^\top\mathbf{1})^2 + (\boldsymbol{\alpha} - \boldsymbol{\beta})^\top\mathbf{a}$$

for $i = 1,2, \ldots, n$ (cyclic coordinate choice)

Terms involving only α_i, β_i are

$$\frac{1}{2}(\alpha_i - \beta_i)^2 + (\alpha_i - \beta_i)u_i$$

$$= \frac{1}{2}\alpha_i^2 + \frac{1}{2}\beta_i^2 - \alpha_i\beta_i + (\alpha_i - \beta_i)u_i$$

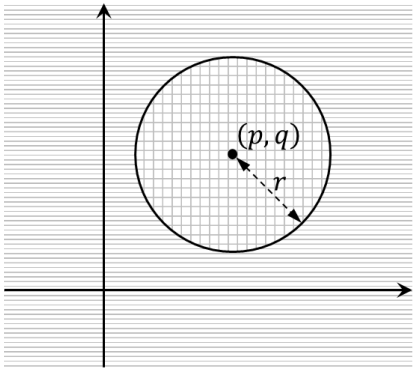
with $u_i \stackrel{\text{def}}{=} \left(a_i + \sum_{j\neq i}(\alpha_j - \beta_j)\right)$ shorthand. **Note**: u_i will keep changing across iterations

While updating α_i, we must ensure $\alpha_i \in [0, 1 - \beta_i]$ as $\alpha_i + \beta_i \leq 1$. The unconstrained optimum is $s_i \stackrel{\text{def}}{=} \beta_i - u_i$ but since quadratics are unimodal functions, the constrained optimum is s_i if $s_i \in [0,1 - \beta_i]$ else if $s_i < 0$, the constrained optimum is 0 else it is $1 - \beta_i$.

Similarly, the unconstrained optimal value for β_i is $t_i \stackrel{\text{def}}{=} \alpha_i + u_i$ but the constrained optimum is t_i if $t_i \in [0,1 - \alpha_i]$ else if $t_i < 0$, the constrained optimum is 0 else it is $1 - \alpha_i$.

Note: use new α_i value when updating β_i

Q5. (Circular argument) Given a circle in 2D plane with centre at $\mathbf{c} = (p, q) \in \mathbb{R}^2$ and radius $r > 0$, we wish to build a classifier that gives output $y = -1$ if a point $\mathbf{x} = (x, y) \in \mathbb{R}^2$ is inside the circle i.e., $(x-p)^2 + (y-q)^2 < r^2$ and $y = +1$ otherwise. Do not worry about points on the boundary. Give a feature map $\phi: \mathbb{R}^2 \to \mathbb{R}^D$ for some $D > 0$ and a corresponding classifier $\mathbf{W} \in \mathbb{R}^D$ such that for any $\mathbf{x} \in \mathbb{R}^2$, $\text{sign}\left(\mathbf{W}^\top \phi(\mathbf{x})\right)$ is the correct output. Your map ϕ must **not depend on** p, q, r but your classifier $\mathbf{W}$ may depend on p, q, r. **(2 + 2 = 4 marks)**



We wish to capture $\text{sign}((x-p)^2 + (y-q)^2 - r^2)$ – expanding the expression gives us

$$\text{sign}(x^2 - 2px + p^2 + y^2 - 2qy + q^2 - r^2)$$

This is readily done by the following combination of feature map and classifier

$$\phi\big((x, y)\big) \stackrel{\text{def}}{=} [x^2, x, y^2, y, 1] \in \mathbb{R}^5$$
$$\mathbf{W} = [1, -2p, 1, -2q, p^2 + q^2 - r^2]$$

Q6. Melbo has learnt a decision tree to solve a binary classification problem with 95 train points of which it gets 48 correct and 47 wrong. There are 10 real features for every training point and the first feature x_1 is an interesting one. x_1 takes only 3 values, namely $0, 1, 2$. Among the train points that Melbo classified correctly, a 5/12 fraction had $x_1 = 0$, a 1/6 fraction had $x_1 = 1$ and the rest had $x_1 = 2$. Melbo got 2/3rd of the training points that had $x_1 = 0$ wrong. Melbo got 1/5th of the training points that had $x_1 = 1$ wrong and 1/5th of the training points that had $x_1 = 2$ wrong. Find out how many train points had the feature value $x_1 = 0$, $x_1 = 1$ and $x_1 = 2$.

(Bonus) Do you notice anything funny about the way Melbo's decision tree gets answers right and wrong? Can you improve its classification accuracy on the training set? You cannot change the decision tree itself, but you can take the decision tree output on a data point and the value of the first feature for that data point x_1 and possibly change the output. What is the improved accuracy? For this part, you may assume that the binary labels are $+1$ and -1. **(2 x 3 = 6 + 3 marks)**

There are 60 points with $x_1 = 0$, 10 points with $x_1 = 1$ and 25 points with $x_1 = 2$.To see what is funny about this DT classifier, let us tabulate the breakup

	Correct	Wrong
$x_1 = 0$	20	40
$x_1 = 1$	8	2
$x_1 = 2$	20	5

➡

	Correct	Wrong
$x_1 = 0$	40	20
$x_1 = 1$	8	2
$x_1 = 2$	20	5

We note that the DT classifies many more points wrongly than correctly if $x_1 = 0$. Thus, if we invert the DT outputs for data points with $x_1 = 0$ i.e., predict $y = +1$ if the DT predicted $y = -1$ and predict $y = -1$ if the DT predicted $y = +1$, then we would get 68/95 points correct. We leave DT predictions on points with $x_1 = 1$ or $x_1 = 2$ untouched.