CS 771A: Introduction to Machine Learning				**Endsem Exam** (18 Nov 2019)
Name	SAMPLE SOLUTIONS			**80 marks** Page **1** of **4**
Roll No		Dept.		

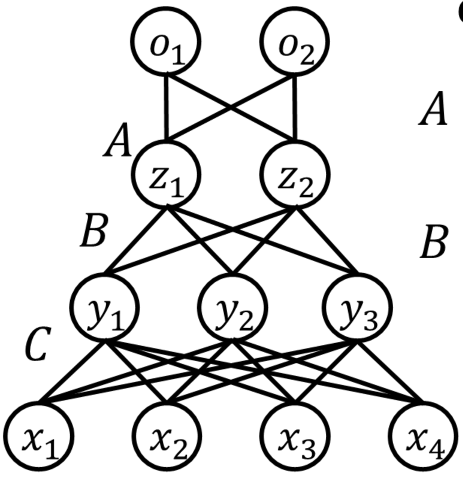
Instructions:

1. This question paper contains 2 pages (4 sides of paper). Please verify.
2. Write your name, roll number, department in **block letters neatly** with ink **on each page** of this question paper.
3. If you don't write your name and roll number on **all** pages, **pages may get lost** when we unstaple to scan pages
4. Write your final answers neatly **with a blue/black pen**. Pencil marks may get smudged.
5. Don't overwrite/scratch answers especially in MCQ and T/F. We will entertain no requests for leniency.

Q1. Write **T** or **F** for True/False (write **only in the box on the right hand side**) **(16x2=32 marks)**

1	If $f, g: \mathbb{R}^2 \to \mathbb{R}$ are two convex fns, then the fn h defined as $h(\mathbf{x}) = f(\mathbf{x}) \cdot g(\mathbf{x})$ can never be a convex fn no matter which two convex functions f, g we choose	F
2	It is possible to derive a Lagrangian dual problem for the L_2-regularized logistic regression problem even though there are no constraints in the primal formulation	T
3	If X, Y are two real-valued r.v.s (not necessarily independent) such that at least one of them has zero variance i.e. $\mathbb{V}[X] = 0$ or $\mathbb{V}[Y] = 0$ then $\text{Cov}(X, Y) = 0$	T
4	The LwP algorithm when used on a binary classification problem, results in a linear decision boundary no matter how many prototypes we use per class	F
5	The time it takes to make a prediction for a test data point with a decision tree with n leaf nodes is always $\mathcal{O}(\log n)$ no matter what the structure of the tree.	F
6	If we have 10000 red and 20 green points, then best option to deal with imbalance is to find 20 red points closest to the green points and throw the rest 9980 away	F
7	Reinf. learning is a good technique to build a RecSys if we suspect that tastes of users are changing (possibly due to our own recommendations to them)	T
8	Bandit algorithms are named so since they operate in settings where a malicious adversary can sometimes corrupt the feedback/response given to the algorithm	F
9	The binary relevance method in recommendation systems is best suited (in terms of prediction time/model size) when the number of items/labels is extremely large	F
10	A NN with a three hidden layers and a single output node with all nodes except input layer nodes using sigmoid activation will always learn a continuous function	T
11	If our goal in RecSys is to quickly find out the most liked item(s) by a certain user, then we should adopt the UCB method rather than pure exploration method	T
12	The EM algorithm is a special case of Q-learning (recall Q learning is used in reinf. learning) since the EM algorithm also optimizes a function known as the Q function	F
13	If we are training an ensemble of k classifiers, then it is very simple to train all of them in parallel when using bagging but not that simple when using boosting	T
14	If we have n data points with d-dimensional feature vectors, then kernel PCA with the Gaussian kernel can learn only at most d components from this data if $d < n$	F
15	If $A \in \mathbb{R}^{n \times n}$ is an orthonormal matrix i.e. $A^\top A = I_n = AA^\top$, then it can never be the case that A is symmetric i.e. we must have $A^\top \neq A$	F
16	Let X be a real valued r.v. that always takes values in the interval $[-1,1]$. Then we must have $\mathbb{V}[\mathbb{E}[X]] = 0$ i.e. if we define $Y = \mathbb{E}[X]$ then we must have $\mathbb{V}[Y] = 0$	T

Q2 Consider the NN with 2 hidden layers – all nodes use the identity activation function. This NN is clearly equivalent to a network with no hidden layers since all activation functions are linear. Find the weights of this new network and write them down in the space provided. **(4 marks)**



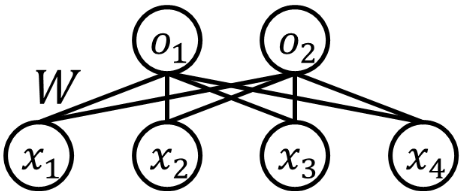
$$\mathbf{o} = A\mathbf{z}, \mathbf{z} = B\mathbf{y}, \mathbf{y} = C\mathbf{x},$$

$$A = \begin{bmatrix} 2 & 0 \\ 2 & 2 \end{bmatrix}, C = \begin{bmatrix} 1 & 3 & 0 & 1 \\ 4 & 1 & 4 & 0 \\ 0 & 2 & 0 & 2 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 3 & 1 \end{bmatrix}$$

$$\mathbf{o} = W\mathbf{x}$$

$$W^\top = \begin{bmatrix} 18 & 50 \\ 22 & 56 \\ 16 & 40 \\ 14 & 26 \end{bmatrix}$$



Q3 Define $f: \mathbb{R}^2 \times \mathbb{R}^{2\times 3} \times \mathbb{R}^3 \to \mathbb{R}$ as $f(\mathbf{x}, W, \mathbf{y}) = \mathbf{x}^\top W \mathbf{y}$ where $\mathbf{x} \in \mathbb{R}^2, \mathbf{y} \in \mathbb{R}^3, W \in \mathbb{R}^{2\times 3}$. Let $\mathbf{x}^0 = [1,2]^\top, \mathbf{y}^0 = [3,4,5]^\top, W^0 = \begin{bmatrix} 1,2,1 \\ 2,1,2 \end{bmatrix}$. Define $p: \mathbb{R}^2 \to \mathbb{R}$ as $p(\mathbf{x}) = \mathbf{x}^\top W^0 y^0$, $q: \mathbb{R}^{2\times 3} \to \mathbb{R}$ as $q(W) = (\mathbf{x}^0)^\top W \mathbf{y}^0$ and $r: \mathbb{R}^3 \to \mathbb{R}$ as $r(\mathbf{y}) = (\mathbf{x}^0)^\top \mathrm{W}^0 \mathbf{y}$. Write the Jacobians of p, q, r below. Note that to avoid clutter, we are asking you to write J^q as a 2×3 matrix. **(2+3+3=8 marks)**

$$J^p = [16 \quad 20], J^r = [5 \quad 4 \quad 5]$$

$$J^q = \begin{bmatrix} 3 & 4 & 5 \\ 6 & 8 & 10 \end{bmatrix}$$

Q4 We wish to use C-SVM to learn a binary classifier. We have 100000 train points half of which are red and the other half green. Briefly outline a way to tune the C parameter and justify your reasons for the same. You may use the 100000 training points in any way you wish. **(4 marks)**

Since the dataset is balanced, we need not resort to class-weighted classification tactics. We may set aside a fair number of randomly chosen points (say 30000) as a held-out validation set, then perform a grid search over a reasonable range of values of C say 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50 and choose the value for which the SVM trained using that value of C gives us maximum classification accuracy on the validation dataset. Note that other methods like k-fold validation etc are also admissible. Also, we are able to use classification accuracy as a performance measure on the validation dataset only because the dataset is balanced. Had the dataset been unbalanced, we should have used F-measure etc instead.

CS 771A: Introduction to Machine Learning				**Endsem Exam** (18 Nov 2019)
Name	SAMPLE SOLUTIONS			**80 marks** Page **3** of **4**
Roll No		Dept.		

Q5 Let $K_1 : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a Mercer kernel with feature map $\phi_1 : \mathcal{X} \to \mathbb{R}^D$ for some finite $D > 0$. Define a new kernel $K_2 = K_1^2$ i.e. $K_2(\mathbf{x}, \mathbf{y}) = K_1(\mathbf{x}, \mathbf{y})^2$ for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$. Design a feature map for K_2 i.e. $\phi_2 : \mathcal{X} \to \mathbb{R}^L$ for some $L > 0$ s.t. $K_2(\mathbf{x}, \mathbf{y}) = \langle \phi_2(\mathbf{x}), \phi_2(\mathbf{y}) \rangle$ for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$. **(6 marks)**

The properties of trace tell us that $\phi_1(\mathbf{x})^\top \phi_1(\mathbf{y}) = \text{trace}\left(\phi_1(\mathbf{x})^\top \phi_1(\mathbf{y})\right) = \text{trace}\left(\phi_1(\mathbf{x})\phi_1(\mathbf{y})^\top\right)$. Also, $c \cdot trace(X) = trace(c \cdot X)$ for all $c \in \mathbb{R}$. Thus we write $K_2(\mathbf{x}, \mathbf{y}) = K_1(\mathbf{x}, \mathbf{y})^2 = \left(\phi_1(\mathbf{x})^\top \phi_1(\mathbf{y})\right)^2 = \text{trace}\left(\phi_1(\mathbf{x})\phi_1(\mathbf{x})^\top \phi_1(\mathbf{y})\phi_1(\mathbf{y})^\top\right)$. If we use $\phi_2(\mathbf{x}) = \phi_1(\mathbf{x})\phi_1(\mathbf{x})^\top \in \mathbb{R}^{D \times D}$, then we have $K_2(\mathbf{x}, \mathbf{y}) = \langle \phi_2(\mathbf{x}), \phi_2(\mathbf{y}) \rangle$ for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$.

Instead of a matrix-valued feature map, we may have a vector feature map as well $\phi_2(\mathbf{x}) \in \mathbb{R}^{D^2}$ i.e. $L = D^2$ by creating coordinates of the form $\mathbf{v}_i \mathbf{v}_j : i, j \in [D]$ where we denote $\mathbf{v} = \phi_1(\mathbf{x})$ (note that this essentially stretches out the $D \times D$ matrix we created earlier as a long vector).

Q6 Derive the Lagrangian dual for the following weighted CSVM problem (for use in Adaboost)

$$\min_{\mathbf{w} \in \mathbb{R}^d} \frac{1}{2}\|\mathbf{w}\|_2^2 + \sum_{i=1}^n c_i \cdot \left[1 - y^i \cdot \mathbf{w}^\top \mathbf{x}^i\right]_+$$

Write down the problem as a constrained opt. problem, write down the Lagrangian, and show main steps in the derivation of the dual. Assume $y^i \in \{-1,1\}, \mathbf{x}^i \in \mathbb{R}^d, c_i > 0$. **(3+1+2=6marks)**

Constrained prob: $\min_{\mathbf{w}, \boldsymbol{\xi}} \frac{1}{2}\|\mathbf{w}\|_2^2 + \sum_{i=1}^n c_i \cdot \xi_i$ s.t. $y^i \cdot \mathbf{w}^\top \mathbf{x}^i \geq 1 - \xi_i$ and $\xi_i \geq 0 \; \forall i$

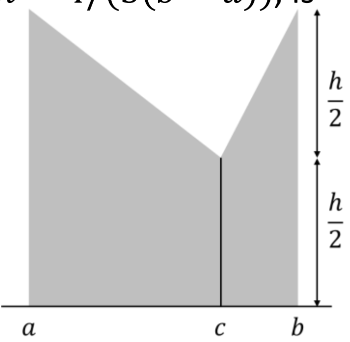
Lagrangian: $\mathcal{L}(\mathbf{w}, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\|\mathbf{w}\|_2^2 + \sum_{i=1}^n c_i \cdot \xi_i + \alpha_i\left(1 - \xi_i - y^i \cdot \mathbf{w}^\top \mathbf{x}^i\right) - \beta_i \xi_i$

Setting $\frac{\partial \mathcal{L}}{\partial \xi_i} = 0$ gives us $\alpha_i + \beta_i = c_i$ whereas $\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \mathbf{0}$ gives us $\mathbf{w} = \sum_{i=1}^n \alpha_i y^i \mathbf{x}^i$.

Simplifying gives $\max_{\boldsymbol{\alpha}} \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i,j=1}^n \alpha_i \alpha_j y^i y^j \langle \mathbf{x}^i, \mathbf{x}^j \rangle$ s.t. $\alpha_i \in [0, c_i] \; \forall i \in [n]$

Q7 Consider the valley distribution with three parameters $\mathcal{V}(a,b,c)$ where $a < b$ and $a \leq c \leq b$ (no other restrictions on a, b, c). The PDF of this distribution, with $h = 4/(3(b-a))$, is

$$\mathbb{P}[x \mid a,b,c] = \mathcal{V}(x;a,b,c) \triangleq \begin{cases} 0 & x < a \\ h - \dfrac{h(x-a)}{2(c-a)} & a \leq x < c \\ h - \dfrac{h(b-x)}{2(b-c)} & c \leq x \leq b \\ 0 & x > b \end{cases}$$



Given n indep. samples $x^1, \ldots, x^n \in \mathbb{R}$ (not all samples are the same) we wish to learn a valley distribution as a generative distribution using MLE i.e. find $\arg\max\limits_{a<b, a\leq c\leq b} \mathbb{P}[x^1, \ldots, x^n \mid a,b,c]$. Give a brief description + derivation of an algorithm to find $\hat{a}_{\text{MLE}}, \hat{b}_{\text{MLE}}, \hat{c}_{\text{MLE}}$. **(5+5+10=20 marks)**

Observation 1: let $m \triangleq \min\limits_i x^i$ and $M \triangleq \max\limits_i x^i$. Then if $a > m$ or $b < M$ then the likelihood would vanish and thus, we must have $a \leq m, b \geq M$.

Observation 2: if $c \in [m, M]$ then if $a < m$ or $b > M$ or both, then we can increase likelihood by keeping c the same and setting $a = m, b = M$. This is because doing so causes $(b-a) \downarrow$ so $h \uparrow$ which causes PDF to go up in the entire interval $[a,b] = [m,M]$ i.e. likelihood of all data points goes up.

Observation 3: if $c < m$, then we can similarly see that setting $a = c = m$ will strictly increase likelihood. Similarly if $c > M$, we may set $b = c = M$.

The above observations tell us that $\hat{a}_{\text{MLE}} = m, \hat{b}_{\text{MLE}} = M$ and $\hat{c}_{\text{MLE}} \in [m, M]$. In general there need not be a closed form solution for $\hat{c}_{\text{MLE}}$. A sensible workaround is to perform search in the interval $[m, M]$. W.l.o.g. assume that $x^1 \leq x^2 \ldots \leq x^n$. Then for all values of $c \in [x^i, x^{i+1}]$, we have the NLL expression as

$$\ell^i(c) = -\sum_{j=1}^{i} \ln\left(1 - \frac{x^j - a}{2(c-a)}\right) - \sum_{j=i+1}^{n} \ln\left(1 - \frac{b - x^j}{2(b-c)}\right)$$

Note that we removed terms involving h above as they do not affect the optimum. The above function may be (approximately) minimized in the range $c \in [x^i, x^{i+1}]$ using GD. The same process needs to be repeated for all $i \in [n-1]$ to obtain an (approximation) of the globally optimal value of c.

Pseudo Algo for estimating $\hat{c}_{\text{MLE}}$:

For $i = 1, \ldots, n-1$, let $\hat{c}^i = \arg\min\limits_{c \in [x^i, x^{i+1}]} \ell^i(c)$ approximated using GD

Output $\hat{c}^k$ where $k = \arg\min\limits_{j \in [n-1]} \ell^j(\hat{c}^j)$