Date: Sunday, January 10, 2021. 9:00 am - 11:30 am

There are 9 questions.You need to attempt any 8 of the questions. Each Question carries 6 points. Max Points: 48

Answer to each question must start on a new page. You need to justify all your answers. Answers without justification may not get full points.

If you are answering all 9 questions, you need to clearly specify the question that you DO NOT want to be graded. Failure to do so will result in randomly discarding one of your answers.

1. **VC-Dim of Axis-parallel Rectangles in 2-D** Consider a hypothesis class given by axis parallel rectangles in 2-D. Each rectangle can be fully specified by the coordinates of the points at the end of its diagonal. Given a rectangle $R_{||}$, let the corresponding hypothesis be defined by $h^+_R(x) = 1$ if x lies inside the ~~circle~~ rectangle, 0 otherwise. Let the corresponding hypothesis class be given as $\mathcal{H}^+$.

   Similarly, consider another hypothesis class consisting of axis parallel rectangles, such that given a rectangle $R_{||}$, the corresponding hypothesis be defined by $h^-_R(x) = 0$ if x lies inside the rectangle, 1 otherwise. Let the corresponding hypothesis class be given as $\mathcal{H}^-$. What is the VC-dimension of the hypothesis class specified by $\mathcal{H}^+ \cup \mathcal{H}^-$? You need to fully justify your answer.

2. **Kernelized Perceptron:** Consider a learning problem where the training data is not linearly separable. Let the training data be given as $\{x^{(i)}, y^{(i)}\}_{i=1}^m$. Assume that class labels are Boolean valued, and each $x^{(i)} \in \mathcal{R}^n$. Assume that there exists a feature transformation $\phi : \mathcal{R}^n \rightarrow \mathcal{R}^N$, such that the data when transformed through the function ϕ becomes linearly separable, e.g., the ϕ could correspond to a polynomial transformation for instance. In general, $N >> n$. Consider learning a perceptron based classifier for modeling this data. You can assume a sigmoid activation function. One way to learn this classifier is to first compute the input features in the transformed space, and then learn a linear classifier using gradient descent over perceptron based loss. But this would be involve computing the ϕ based transformation explicitly which may be computationally expensive. Here, we describe an alternate (possibly more efficient) way of doing this under certain assumptions:

   (a) Explicitly write the loss function for learning a perceptron based classifier in the transformed feature space.

   (b) Assume that $\exists$ a function $K(x, z)$, where $K : \mathcal{R}^n \times \mathcal{R}^n \rightarrow \mathcal{R}$ such that $K(x, z) = \phi(x)^T \phi(z)$. Show that entire parameter update algorithm for learning the perceptron can be written in terms of $\phi(x)^T \phi(z)$, without ever explicitly computing $\phi(x)$ for any given input x.

   (c) Give one practical advantage of the scheme as described above in terms of the computational complexity of the learning algorithm (as opposed to explicitly computing $\phi(x)$'s).

Hint: You will have to think of a scheme where you can implicitly represent θ's in terms of the coefficients of input features $x^{(i)}$'s. You can assume that weights are initialized to $\vec{0}$ in the beginning of learning.

3. **Convex Functions and Jensen's Inequality:** Consider a function $f : \mathcal{R}^n \rightarrow \mathcal{R}$ such that f is convex.

   (a) Formally write the conditions under which we say that function f is a convex function; your conditions should not make any use of derivatives of the function f.

   (b) Let $x^{(1)}, x^{(2)}, \cdots x^{(k)}$ be a set of k points where $x^{(i)} \in \mathcal{R}^n$, $1 \leq i \leq k$. Assuming f is convex, show that ~~$\sum_k \alpha_k f(x_k) \geq f(\sum_k \alpha_k x_k)$ where $\sum_k \alpha_k = 1$~~ $\sum_{i=1}^k \alpha_i f(x_i) \geq f(\sum_{i=1}^k \alpha_i x_i)$ where $\sum_{i=1}^k \alpha_i = 1$, and $\alpha_i \geq 0, \forall i$. You should show this for any given finite value of k.

   (c) Consider a distribution $p(X = x)$ where $x \in \mathcal{R}^n$. Write down the Jensen's inequality involving function f (assumed to be convex) and distribution p defined over X. Using part (b) above, derive a proof of Jensen's inequality.

4. **Unconstrained SVMs:** Consider learning an SVM based classifier over a set of training points $\{(x^{(i)}, y^{(i)})\}_{i=1}^m$. Assume that we are learning a linear SVM. Assume a binary classification problem, and that $x^{(i)} \in \mathcal{R}^n$. Recall that an SVM is characterized by the parameters $w \in \mathcal{R}^n$, $b \in \mathcal{R}$, and the corresponding separating hyperplane is given as : $w^T x + b = 0$. Recall that the soft-margin formulation for SVMs is given as:

$$\min_{w.b} \frac{1}{2} w^T w + C \sum_i \xi_i$$
$$y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi_i, \forall i$$
$$\xi_i \geq 0, \forall i$$

   In this problem, instead of solving of the SVM based optimization via the use of Langrangians as done in the class, we will take an alternate approach.

   (a) We introduce a new kind of loss, called the Hinge loss, defined as: $L_H(z) = 1 - z$ if $z < 1$, and 0 otherwise ($z \in \mathcal{R}$). Use this Hinge loss to convert your constrained optimization problem in the part (a) above into an unconstrained optimization problem. Hint: you will need to appropriately define the Hinge loss over each of your constraints, and then define an objective directly incorporating this loss, such that we can get rid of the constraints and get an equivalent unconstrained formulation.

   (b) Note that the Hinge loss as defined in part (a) above, is non-differentiable at $z = 1$. In Assignment 3, we looked at ReLU activation unit, which was non-differentiable, but we could still back-propagate using sub-gradients. Compute the sub-gradient for the unconstrained loss (non-differentiable) in the part (b) above. Write down the expression for performing gradient descent for computing the parameters of the SVM in the above formulation.

5. **Logistic Regression:** Using matrix calculus, show that the loss function of logistic regression model is a convex function of its parameters. You can assume training data to be given as $\{(x^{(i)}, y^{(i)})\}_{i=1}^{m}$ where $x^{(i)} \in \mathcal{R}^n$ and $y^{(i)}$'s are Boolean valued.

6. **Markov Networks:** In class, we looked at the Bayesian networks, which are directed graphical models. Their undirected counterparts are defined by Markov networks. Given a set of variables $X = \{X_1, \cdots X_n\}$ of discrete valued variables, and a Graph G over nodes corresponding to variables in the set X, a Markov network defines a distribution $P(X) = \frac{1}{Z}\Pi_k \Phi_k(X_{C_k})$, where X_{C_k} represents a clique (i.e., fully connected subset of nodes) in G, and each Φ_k represents a non-negative valued function defined over variables in the clique C_k. Each of the terms $\Phi_k(X_{C_k})$ is referred to as a factor. Z is defined in such a manner that the probabilities add up to 1.

   (a) Compute the value of the normalization constant Z in the distribution defined by a Markov network. Describe the computational complexity to compute it, assuming all the variables in the set X are Boolean valued.

   (b) Using the fact that factors in a Markov networks are defined over cliques in the graph, give an algorithm to convert a Bayesian network into an "equivalent" Markov network such that conditional distribution of a node given its parent corresponds to a factor in the resultant Markov network, and vice-versa. You should justify the steps of your algorithm. This establishes an equivalence between the two forms of Graphical Models.

   (c) Recall that in a Bayesian network B, the set of independences are given as $X_j \perp \text{Non-Desc}(X_j)|\text{Pa}(X_j)$. Whereas in a Markov network M, the set of independences are given as $X_j \perp \text{Non-Nbrs}(X_j)|\text{Nbr}(X_j)$. Here, $\text{Pa}(X_j)$ and $\text{Non-Desc}(X_j)$: parents and non-descendants of node X_j, respectively, in the Bayesian network graph; $\text{Nbr}(X_j)$ and $\text{Non-Nbrs}(X_j)$: neighbors and non-neighbors of node X_j, respectively, in the Markov network graph. With the help of an example, show that the construction of an equivalent Markov network as done in part (b) above can result in loss of some independences in the original Bayesian network.

7. **Understanding Basic Concepts:** Answer the following questions.

   (a) Consider the soft-margin SVM formulation as given below.

$$\min_{w.b} \frac{1}{2} w^T w + C \sum_i \xi_i$$
$$y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi_i, \forall i$$
$$\xi_i \geq 0, \forall i$$

   Here, the symbols are as described in the class. We would like to understand the behaviour of the model with varying values of the C parameter. Specifically, as the value of C parameter is increased (and made ∞ in the limit), is the model more likely to result in overfitting or underfitting? Justify your answer.

(b) Assume that you are learning a model M using a training algorithm. You have a separate training set t_s and a validation set v_s. Suppose that the learned model M gives low accuracy on both t_s and ~~t_v~~ v_s. (a) Do you think it is case of under-fitting or over-fitting? (b) As a first-step remedy, which one of the following would you choose and why (i) Move to a more sophisticated learning model (ii) Gather more training data. You need to justify your answer in each case.

(c) An unstable learning is a learning model which has the potential to change abruptly with minor changes in the training data distribution. Similarly, a stable learner is a learning model which is robust to minor changes in the training data distribution. Describe for each of the following if they are stable or unstable learners (a) Decision Trees (b) Naïve Bayes. Justify your answer in each case.

8. **Principal Component Analysis** Consider a set of data points given as $\{x^{(i)}\}_{i=1}^{m}$. Assume that $x^{(i)} \in \mathcal{R}^n$. You can assume the data to be normalized to zero mean and unit variance. Consider applying principal component analysis (PCA) on this data. Recall that PCA tries to find a set of orthonormal vectors $u_1, u_2, \cdots u_k$ such that variance of the data when projected on the subspace by $u_1, u_2 \cdots u_k$ is maximised.

   (a) Using the first principles, derive the expression for the objective function for PCA expressed in terms of the empirical co-variance matrix Σ of the data.

   (b) Assume $k = 2$. Show that vectors u_1 and u_2 correspond to the top two eigenvectors of the matrix Σ. Hint: You may need to appropriately define a constrained optimization problem, and work with the theory of Langrangians.

9. **Decision Tree Post Pruning:** In reduced error pruning of decision trees, we first fully grow a tree on the training data, and greedily prune the nodes (and sub-tree below it) using a validation set, until there is no further increase in the validation set accuracy. Show that using this greedy strategy for pruning is the optimal strategy. In other words, let T denote the tree grown from the training data, T_p denote the tree obtained by pruning T using the greedy strategy, and T' be some tree obtainable from T after pruning of nodes, then, it must be the case that ~~$\epsilon_{val}(T_p) \leq \epsilon_{val}(T'), \forall T'$, where the function $\epsilon_{val}(.)$~~ $\mathcal{A}_{val}(T_p) \geq \mathcal{A}_{val}(T'), \forall T'$, where the function $\mathcal{A}_{val}(.)$ computes the validation set accuracy of the tree given as its argument.