Date: Sunday, November 21, 2021. 9:00 am - 11:20 am
There are 7 questions. All questions are compulsory. Each Question carries 6 points. Max Points: 42. There are four printed pages.
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.

1. Consider learning a soft-margin SVM model over the training data $\{x^{(i)}, y^{(i)})\}_{i=1}^{m}$. Recall that SVM problem can be written as:

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_i \xi_i$$

$$y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi_i, \forall i$$

Let the optimal weight vector obtained by solving the above problem in the dual space be given as $w = \sum_i \alpha_i y^{(i)} x^{(i)}$, where α_i are Langrange multipliers as defined in the class. Assume that $\exists i'$, $y^{(i')} = 1$ and $x^{(i')}$ is a support vector ($\alpha_{i'} > 0$) and $x^{(i')}$ lies on the margin boundary.

(a) Show that $\exists i'', y^{(i'')} = -1$ and $y^{(i'')}(w^T x^{(i'')} + b) \leq 1$. You should prove your result mathematically.

(b) Describe the procedure for computing the value of the b parameter purely in terms of the value of the α_i parameters, and $x^{(i)}$ and $y^{(i)}$'s. You should not simply say "pick a point on the margin boundary", rather characterize the procedure in terms of the values of the langrange multiplier α_i's.

(c) Show that above computation of the parameters w and b is amenable to learning with a Kernel function K. In other words, if ϕ represents the feature transformation corresponding to the Kernel K, then the prediction for a new point x can be computed solely using the kernel function, and does not require explicitly computing the feature transformation ϕ.

2. Consider an RNN architecture for solving a sequence labeling task - where an output needs to be produced for each input token (one token is processed at each time step in the RNN). Assume that all examples have the same number of tokens, given by T. Further, assume that the output at each time step is Boolean valued. Let the input to the RNN be given as a sequence $(x^1, x^2, .., x^t, .., x^T)$ where each $x^t \in R^n$. Let $h^t \in R^p$ denote the output of the unit at time step t which is fed into the unit at time $t+1$. Therefore, the output of the hidden unit at time t is given as: $h^t = \sigma(W_I x^t + W_h h^{t-1} + b)$, where $W_I \in \mathcal{R}^{p \times n}$, $W_h \in \mathcal{R}^{p \times p}$, and $b \in \mathcal{R}^p$. Here, the application of the σ (sigmoid) function over a vector simply denotes an element wise application. Further, let the prediction output by the network at time step t be given by y^t, such that $y^t = \sigma({w_o}^T x^t + {w'_o}^T h^t + b_o)$, where $w_o \in \mathcal{R}^n$, $w'_o \in \mathcal{R}^p$, and $b_o \in \mathcal{R}$. Note that all the parameters are tied across time steps. Assume $h^0 = 0$. Assume squared loss over m training examples of the form $\{x^{(i)}, y^{(i)}\}_{i=1}^{m}$, derive the expression for the gradient of the loss with respect to the W_I and W_h parameters using backpropagation. Note

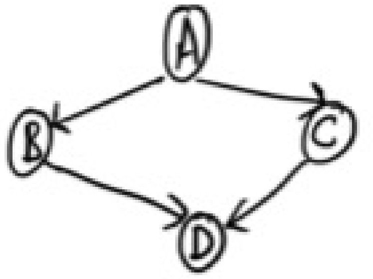

Figure 1: A simple Bayesian network

that here, each $x^{(i)}$ denotes a sequence of the form $(x^{(i)1}, x^{(i)2}, .., x^{(i)t}, .., x^{(i)T})$ where each $x^{(i)t} \in \mathcal{R}^n$. Similarly $y^{(i)}$ denotes a sequence $(y^{(i)1}, y^{(i)2}, .., y^{(i)t}, .., y^{(i)T})$ where each $y^{(i)t} \in \{0, 1\}$.

3. (a) Consider a hypothesis class $\mathcal{H}$, and an instance space given by $\mathcal{X}$. Consider a binary classification problem over the instance space $\mathcal{X}$. Given some training data $D = \{x^{(i)}, y^{(i)}\}_{i=1}^m$, drawn from an underlying distribution, we say that a hypothesis $h \in \mathcal{H}$ is consistent with D, if the training error $\hat{\epsilon}(h) = 0$, i.e., $\forall i, h(x^{(i)}) = y^{(i)}$. If $\epsilon(h^*)$ denotes the generalization error achieved by the optimal hypothesis $h^* \in \mathcal{H}$, what is the probability that $\exists h \in \mathcal{H}$, such that h is consistent with D.

   (b) Consider an instance space $\mathcal{X} = \mathcal{R}^2$. Assume a binary classification problem. Show that no set of four points in $\mathcal{X}$ can be shattered by a linear hypothesis class.

4. Bayesian networks are a generalization of Naïve bayes, encoding a more general set of conditional independences among a set of variables given as $X = (X_1, X_2, \cdots, X_n)$. Specifically, a Bayesian network is represented by a DAG (Directed Acyclic Graph) G, where there is a node in the DAG for every variable in the set X, and the set of conditional independences is given as $(X_j \perp \textit{Non-Desc}(X_j))|Pa(X_j)$, where $Pa(X_j)$ denotes the set of parents of X_i in G, and $\textit{Non-Desc}(X_j)$ is the set of all non-descendants of X_j in G.

   (a) Prove that Naíve bayes model is a special kind of Bayesian network.

   (b) Show that the joint distribution defined by a Bayesian network over a set of variables (given by X) can be represented as $\prod_{j=1}^{n} P(X_j|Pa(X_j))$. The parameters specifying the conditional distribution $P(X_j|Pa(X_j)$ for each variable node X_j become the parameters of the underlying Bayesian network.

   (c) Consider the Bayesian network shown in Figure 1. Derive the expression for $P(D|A, B)$ in terms of the conditional probabilities of the form $P(X_j|Pa(X_j))$, where $X_j \in \{A, B, C, D\}$. Is it necessarily the case that $(D \perp A)|B$. If yes, prove. If not, construct a counter example.

5. (a) Show that K-means algorithm can be seen as block coordinate descent over a suitably defined loss function $\mathcal{L}_m$, where the descent happens over two blocks (sets) of variables.

   (b) Show that each step of block coordinate descent in K-means can be solved analytically and hence, the corresponding optimization problems have a unique minima.

   (c) For a mixed (discrete + continuous) variable optimization problem, we define a local minima to be an assignment to the variables, such that any movement of the continuous variables in a local neighbourhood does not improve the loss, independent of the assignment to the discrete variables. With the help of an example, show that $\mathcal{L}_m$ can have multiple local minimas.

6. (a) Given two distributions P and Q over a K valued discrete random variable u, cross entropy between P and Q is defined as: $-\sum_{k=1}^{K} Q(u_l) log P(u_l)$. For a multi-valued classification problem (i.e., where the target variable y can take values from a discrete set), show that maximum (log)-likelihood based objective under any given learning model can be equivalently written as a minimizing the cross entropy between two suitably defined distributions. You should clearly specify the forms of the distributions P and Q for this equivalence to hold.

   (b) Regularization is popular technique to avoid over-fitting. In $L1$-regularization, 1-norm of the parameters is added to the loss function to define a regularized objective, which is then optimized to find the set of regularized parameters. Consider a learning problem with training data $\{x^{(i)}, y^{(i)}\}_{i=1}^{m}$. Let $x^{(i)} \in \mathcal{R}^n$, and $y^{(i)} \in \{0, 1\}$. Write down the L1-regularized loss for logistic regression in this setting. Compute the (sub)-gradient of the L1-regularized loss. Finally, compute the expression for the hessian matrix H (wherever gradient is defined), and show that L1-regularized loss for logistic regression is a convex at all such points (using properties of H). Hint: To recall the definition of sub-gradients, refer to Assignment 3. What can you say about convexity property for the set of points where the function is non-differentiable? Justify your answer.

7. Recall that in the standard decision tree implementation, we used the majority class to assign the target (predicted) value to the leaf node. In Naïve bayes decision trees, we instead learn an independent Naïve bayes classifier at each leaf and use the classifier to the assign the target (predicted) value. Below, you will work out some technical details involved in building a Naïve bayes decision tree.

   (a) Describe the changes in the entropy function computation which is used for picking the "best" attribute to split on next for implementing Naïve bayes decision trees. For notational convenience, you can assume that you are splitting on an internal node with data $\{x^t, y^t\}_{t=1}^{T}$. Further, assume that each attribute X_j $(1 \leq j \leq n)$ can take values from a fixed domain $\mathcal{D}$ of discrete values. You can assume a binary classification problem.

   (b) Clearly describe the suitable modifications to the convergence criteria for implementing Naïve bayes decision trees.

(c) Prove that Naïve bayes decision trees are representationally strictly more powerful (in terms of decision boundaries that they can learn) compared to standard decision trees.