# COL 774: Machine Learning. Major Examination

Thursday April 10, 2018

**Notes:**

- **Time: 10:30 am to 12:30 pm. Total Questions: 8. Max Score: 35**
- **Each question carries 5 points. Attempt any seven.**
- **This exam is closed book/notes. Start each answer on a new page.**
- **Justify all your answers. Answers without justification may not get full points.**

1. Consider a simple Bernoulli experiment performed $m$ times. Let $\phi \in (0,1)$ denote the corresponding Bernoulli parameter. Let the data (set of values) obtained in these $m$ trials be given as $D = \{x^{(i)}\}_{i=1}^{m}$, where each $x^{(i)} \in \{0,1\}$. Then, we define the maximimum-likelihood (ML) estimate as: $\phi_{ML} = \arg\max_{\phi} P(D|\phi)$. Note that in this expression, $\phi$ is treated as a random variable. It is easy to show that the expression for $\phi^{ML}$ can be written as:

$$\phi_{ML} = \frac{\sum_{i=0}^{m} \mathbb{1}\{x^{(i)} = 1\}}{m} \tag{1}$$

We have seen in class this estimate may not work well especially when $m$ is small (think of what happens when $m = 1$). Therefore, a smoothing estimate is often used, defined as:

$$\phi^{s}_{ML} = \frac{(\sum_{i=0}^{m} \mathbb{1}\{x^{(i)} = 1\}) + c}{m + 2c} \tag{2}$$

where $\mathbb{1}$ denotes the indicator function. The term $c$ is the smoothing parameter and controls the amount of that data we effectively model to have seen apriori. When $c = 0$, this becomes exactly the ML estimate. In this problem, we will give a principled justification for making the smoothing approximation. Let us define another quantity, known as the MAP estimate as, $\phi_{MAP} = \arg\max_{\phi} P(\phi|D)$. Then:

(a) Using Bayes rule, write the expression for $\phi_{MAP}$ only in terms of $P(\phi)$ and $P(D|\phi)$.

(b) Show that $\phi_{ML} = \phi_{MAP}$ when $P(\phi)$ is uniform.

(c) Show that $\phi^{s}_{ML} = \phi_{MAP}$ with an appropriately defined distribution $P(\phi)$. Find the expression for $P(\phi)$ (as a function of $c$ parameter) such that this equivalence holds. It is ok if you leave $P(\phi)$ unnormalized.

2. (a) Consider a binary classification problem over linearly separable data. You can't make any further assumptions about the data distribution. For each pair below, state which of the two algorithms do you expect to have higher bias. Justify your answer in each case.

   i. multilayer network vs single perceptron.

   ii. Gaussian naive Bayes vs logistic regression. Recall: Gaussian naive Bayes is naive Bayes model over real-valued attributes where the distribution of each attribute given the class is Gaussian.

   (b) In this part, we relax the assumption of linearity made in the previous part - so that data could come from an arbitrary distribution. For each pair below, state which of the two algorithms do you expect to have higher variance. Justify your answer in each case.

   i. linear regression vs locally-weighted linear regression.

   ii. random forest vs decision tree.

   iii. hard (kernel) SVM vs soft (kernel) SVM. Recall: hard SVM gives $\infty$ penalty for margin-violators; soft SVM gives finite penalty for margin violators.

3. Consider an (unsupervised) learning problem where we are given the data of the form $\{x^{(i)}\}_{i=1}^{m}$. Further, assume that we know how to model $P(x^{(i)}, z^{(i)}; \Theta)$, where $z^{(i)}$'s are hidden variables and $\Theta$ denotes the set of parameters of the model. Let $L(\Theta)$ denote the log-likelihood of the (observed) data for a given set of parameters $\Theta$. In class, we showed that the local-maxima of $L(\Theta)$ can be obtained using the EM algorithm, which alternates between an $E$ step computation and an $M$ step computation. Show that EM can be equivalently shown to be performing block coordinate ascent over an appropriately defined function $f(\Theta, Q)$, where $\Theta$ is the set of parameters of the original model, and $Q = \{Q_i\}_{i=1}^{m}$ is a set of $m$ distribution where each $Q_i$ defines a distribution over $z^{(i)}$. Write down the expression for the function $f$. You need to clearly show the equivalence between two steps of the EM algorithm (E and M), and the two block coordinate descent steps.

4. Consider the following (modified) SVM problem:

$$\min_{w,\xi} \frac{1}{2}|w| + C\sum_{i=1}^{m}\xi_i$$
$$\forall i.\, y^{(i)}(w^T x^{(i)}) \geq 1 - \xi_i$$
$$\forall i, \xi_i \geq 0$$

Here, the symbols are as described in class. Note that in this modified formulation, the two norm square, i.e., $w^T w$, has been replaced by one-norm $|w|$. Further, we have assumed the intercept term $b$ to be zero. We would like to solve this problem directly in the primal formulation.

(a) Derive the equivalent unconstrained minimization problem for the above formulation. You need to justify the steps in your derivation.

(b) How would you use sub-gradient based optimization to compute the optimal value of $w$ parameters? You also need to compute the values of any relevant gradients/sub-gradients required to solve the problem.

5. Consider an $n \times n$ grayscale image. Let $x_{ij}$ denote the value of the $(i,j)^{th}$ pixel in the image. Consider applying the convolution operation over this image using a $r \times r$ two dimensional kernel. Let $K_{uv}$ denote the $(u,v)^{th}$ element of this kernel $(1 \leq u, v \leq r)$. Assuming a stride of 1, the size of the output obtained after applying the convolution operation is given as $(n-r+1) \times (n-r+1)$. Let $z_{pq}$ denote the $(p,q)^{th}$ element of this output. Each $z_{pq}$ is processed using a sigmoidal activation unit, and let $y_{pq}$ denote the corresponding output element. Consider a neural network which applies the convolution operation as defined above followed by the non-linearity. Let $J$ denote the loss function of this network. Compute the expression for $\frac{\partial J}{\partial K_{11}}$ in terms of $\frac{\partial J}{\partial y_{pq}}$'s.

6. (a) Let $\mathcal{H}$ denote a finite hypothesis class over some instance space $\mathcal{X}$. Show that VC-dimension of $\mathcal{H}$ is upper bounded by $log(|\mathcal{H}|)$. You can assume a binary label space.

(b) Consider the instance space $\mathcal{X} = \mathcal{R}$. Let $I = \bigcup_k (a_k, b_k)$ denote a union of open intervals of the form $(a_k, b_k)$ defined over the real-line. Assume that $a_k < b_k$ and that the intervals are non-overlapping. Corresponding to $I$, we define a hypothesis $h_I$ such that for $x \in \mathcal{R}$, $h_I(x) = 1$ if $\exists k$ such that $x \in (a_k, b_k)$, and $h_I(x) = 0$ otherwise. Given an integer $r > 0$, let $\mathcal{H}_r$ denote the class consisting of hypotheses defined over union of exactly $r$ number of open intervals. Given a set of $d$ distinct points $\{x_1, x_2, \cdots, x_d\}$, what is the minimum value of $r$ such that $\mathcal{H}_r$ shatters this set. Justify your answer. As before, you can assume a binary label space.

7. Consider a binary classification problem over a set of $m$ examples $\{x^{(i)}, y^{(i)}\}_{i=1}^m$. Let $y^{(i)} \in \{-1, 1\}, \forall i$. Consider learning a classifier $h$ over this set of examples. Note that $h(x^{(i)}) \in \{-1, 1\}, \forall i$. Let the error rate of the classifier be given as: $\epsilon = \frac{\sum_{i=1}^m \mathbb{1}\{h(x^{(i)}) \neq y^{(i)}\}}{m}$, where $\mathbb{1}$ is the indicator function. The idea in boosting is to build another classifier over the same set of training examples, but where $i^{th}$ example is assigned a weight $w^{(i)}$. While computing the (weighted) log-likelihood, each example is multiplied by its corresponding weight before taking sum over all the examples. We use $w^{(i)} = \exp(-y^{(i)} h(x^{(i)})\alpha)$ where $\alpha$ is a constant independent of the specific example.

(a) **Determining weights:** Let us define the weighted error of the original classifier $h$ as: $\epsilon^w = \frac{\sum_{i=1}^m w^{(i)} \mathbb{1}\{h(x^{(i)}) \neq y^{(i)}\}}{\sum_{i=1}^m w^{(i)}}$. We will choose the constant $\alpha$ such that the net weighted error $\epsilon^w$ of the hypothesis $h$ becomes 0.1. Derive the expression for $\alpha$ (only) in terms of unweighted error $\epsilon$.

(b) **Building a new Classifier:** Next we train a new classifier $h_1$ using the weights derived in part (a) above. Let $\epsilon_1^w$ denote its error on the reweighted set of examples, i.e., $\epsilon_1^w = \frac{\sum_{i=1}^m w^{(i)} \mathbb{1}\{h_1(x^{(i)}) \neq y^{(i)}\}}{\sum_{i=1}^m w^{(i)}}$. Define an aggregate classifier $h^*$ to be a linear combination of $h$ and $h_1$, i.e., $h^*(x^{(i)}) = \alpha h(x^{(i)}) + \alpha_1 h_1(x^{(i)})$ where $\alpha$ is as determined in the part above and $\alpha_1$ is another constant. Define an aggregate (exponential) error metric $AE = \sum_{i=1}^m \exp(-y^{(i)} h^*(x^{(i)}))$. Find the value of $\alpha_1$ which minimizes AE as defined above. You should express $\alpha_1$ only as a function of the weighted error $\epsilon_1^w$.

8. Consider a set of $m$ instances $\{x^{(i)}\}_{i=1}^m$. Let $x^{(i)} \in \mathcal{R}^n$. Let $\Sigma$ denote the empirical co-variance matrix, i.e., $(j,k)^{th}$ entry of $\Sigma$ denotes the (empirical) co-variance among the $j^{th}$ and $k^{th}$ attributes.

(a) Show that $\Sigma$ can be written as a sum of outer products, i.e., $\Sigma = \sum_l a_l b_l^T$, where $a_l, b_l$ are suitably defined vectors in $\mathcal{R}^n$. You need to state how many terms are there in the sum and also give the exact expression for $a_l$'s and $b_l$'s.

(b) Show that $\Sigma$ is positive semi-definite (do not forget to show that it is symmetric).

(c) State two different learning algorithms (supervised or unsupervised) which explicitly make use of the $\Sigma$ matrix in their computation. Briefly describe where $\Sigma$ is used in the computation in each case.