CS771: Practice Set 2

Problem 1

(A More Fancy Version of?) Consider a classification model where we are given training data $\{\boldsymbol{x}_n, y_n\}_{n=1}^N$ from K classes. Each input $\boldsymbol{x}_n \in \mathbb{R}^D$ and each class c is defined by two parameters, $\boldsymbol{w}_c \in \mathbb{R}^D$ and a $D \times D$ positive definite (PD) matrix $\mathbf{M}_c$, $c = 1, 2, \ldots, K$. Assume N_c denotes the number of training examples from class c. Suppose we estimate $\boldsymbol{w}_c$ and $\mathbf{M}_c$ by solving the following optimization problem

$$(\hat{\boldsymbol{w}}_c, \hat{\mathbf{M}}_c) = \arg\min_{\boldsymbol{w}_c, \mathbf{M}_c} \sum_{\boldsymbol{x}_n : y_n = c} \frac{1}{N_c} (\boldsymbol{x}_n - \boldsymbol{w}_c)^\top \mathbf{M}_c (\boldsymbol{x}_n - \boldsymbol{w}_c) - \log |\mathbf{M}_c|$$

(note that, in the above objective, the $\log |\mathbf{M}_c|$ term ensures positive definiteness of $\mathbf{M}_c$ because the determinant of a PD matrix is always non-negative)

For the given objective/loss function, find the optimal values of $\boldsymbol{w}_c$ and $\mathbf{M}_c$ using first-order optimality (you may use standard results of derivatives of functions w.r.t. vectors and matrices from the Matrix Cookbook[1]). Also, what will this model reduce to as a special case when $\mathbf{M}_c$ is an identity matrix?

Problem 2

(Corrective Updates) Consider the weight vector update equation of the Perceptron algorithm for binary classification: $\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} + y_n \boldsymbol{x}_n$. Assume $y_n \in \{-1, +1\}$.

Prove that these updates are “corrective” in nature, i.e., if the current weight vector $\boldsymbol{w}^{(t)}$ mispredicts $(\boldsymbol{x}_n, y_n)$ (i.e., if $y_n {w^{(t)}}^\top x_n < 0$) then after this update, the new weight vector $\boldsymbol{w}^{(t+1)}$ will mispredict this example by a ”lesser extent” (i.e.., $y_n {\boldsymbol{w}^{(t+1)}}^\top \boldsymbol{x}_n$ will be less negative than $y_n {w^{(t)}}^\top x_n < 0$ after this update).

Problem 3

(Arbitrary Choice?) Formally, show that changing the condition $y_n(\boldsymbol{w}^\top \boldsymbol{x}_n + b) \geq 1$ in SVM to a different condition $y_n(\boldsymbol{w}^\top \boldsymbol{x}_n + b) \geq m$ does not change the effective separating hyperplane that is learned by the SVM. Assume the hard-margin SVM for simplicity.

Problem 4

(Recover the Bias) Assuming hard-margin SVM, show that, given the solution for the dual variables α_n’s, the bias term $b \in \mathbb{R}$ can be computed as $b = y_s - t_s$ where s can denote the index of *any* of the support vectors, and t_s is a term that requires computing a summation defined over all the support vectors. (Hint: Use KKT conditions)

Problem 5

(Look Ma, No Subgradients!) Show that we can rewrite regression with *absolute* loss function $|y_n - \boldsymbol{w}^\top \boldsymbol{x}_n|$ as a *reweighted* least squares objective where the squared loss term for each example $(\boldsymbol{x}_n, y_n)$ is multiplied by an importance weight $s_n > 0$. Write down the expression for s_n, and briefly explain why this expression for s_n

[1] https://www.math.uwaterloo.ca/ hwolkowi/matrixcookbook.pdf

makes intuitive sense. Given N examples $\{(\boldsymbol{x}_n, y_n\}_{n=1}^N$, briefly outline the steps of an optimization algorithm that estimates the unknowns ($\boldsymbol{w}$ and the importance weights $\{s_n\}_{n=1}^N$) for this reweighted least squares problem.

Problem 6

(Linear Regression viewed as Nearest Neighbors) Show that, for the unregularized linear regression model, where the solution $\hat{\boldsymbol{w}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\boldsymbol{y}$, the prediction at a test input $\boldsymbol{x}_*$ can be written as a weighted sum of all the training responses, i.e.,

$$f(\boldsymbol{x}_*) = \sum_{n=1}^N w_n y_n$$

Give the expression for the weights w_n's in this case and briefly discuss (<50 words) in what way these weights are different from the weights in a *weighted version* of K nearest neighbors where each w_n typically is the inverse distance of $\boldsymbol{x}_*$ from the training input $\boldsymbol{x}_n$. **Note:** You do not need to give a very detailed expression for w_n (if it makes algebra messy) but you must give a precise meaning as to what w_n depends on and how it is different from the weights in the weighted K nearest neighbors.

Problem 7

(Feature Masking as Regularization) Consider linear regression model by minimizing the squared loss function $\sum_{n=1}^N (y_n - \boldsymbol{w}^\top\boldsymbol{x}_n)^2$. Suppose we decide to mask out or "drop" each feature x_{nd} of each input $\boldsymbol{x}_n \in \mathbb{R}^D$, independently, with probability $1-p$ (equivalently, retaining the feature with probability p). Masking or dropping out basically means that we will set the feature x_{nd} to 0 with probability $1-p$. Essentially, it would be equivalent to replacing each input $\boldsymbol{x}_n$ by $\tilde{\boldsymbol{x}}_n = \boldsymbol{x}_n \circ \boldsymbol{m}_n$, where $\circ$ denotes elementwise product and $\boldsymbol{m}_n$ denotes the $D \times 1$ binary mask vector with $m_{nd} \sim$ Bernoulli(p) ($m_{nd} = 1$ means the feature x_{nd} was retained; $m_{nd} = 0$ means the feature x_{nd} was masked/zeroed).

Let us now define a new loss function using these masked inputs as follows: $\sum_{n=1}^N (y_n - \boldsymbol{w}^\top\tilde{\boldsymbol{x}}_n)^2$. Show that minimizing the *expected* value of this new loss function (where the expectation is used since the mask vectors $\boldsymbol{m}_n$ are random) is equivalent to minimizing a **regularized** loss function. Clearly write down the expression of this regularized loss function. Note that showing this would require some standard results related to expectation of random variables, such as linearity of expectation, and expectation and variance of a Bernoulli random variable. Note that, so far in the course, we haven't talked much about probability ideas but, with this much information, you should be able to attempt this problem.