MIT 9.520/6.860: Statistical Learning Theory and Applications **Fall 2022**

Problem Set 1

Student Name: Aniruddha Deshpande

Problem 1

Problem 1.1

In this problem, I will be using the same approach to simplify the integrals for all the loss functions. This will involve first, computing the inner integral over y. Since it is a binary classification problem, $y \in \{-1, 1\}$ and the integral over y is simply a sum over the 2 possible values it can take. Then to find the f that minimizes the integral over x , I will use the result from section 1.5.1 of the book wherein we have for $L_x(a) = l(y, a)dP(y|x)$it is possible to show that to minimise the integral below over f, it is sufficient to minimize $L_x(f(x))$ over f and that will give us the same optimal function.

$$L(f) = \int dP_X(x)L_x(f(x)) \tag{1.1}$$

Exponential loss $l(-yf(x)) = exp(-yf(x)$

$$\begin{aligned}\mathcal{E}(f) &= \int_{X\times Y} exp(-yf(x))p(x)p(y|x)\,dx\,dy \\ &= \int_X p(x)[exp(f(x))p(-1|x) + exp(-f(x)p(1|x)]\,dx\end{aligned}$$

Taking derivative of the term inside, over $f(x)$, and setting it to 0 , we get

$$exp(f(x))p(-1|x) - exp(-f(x))p(1|x) = 0$$

Solving this and then using the fact that $p(-1|x) = 1 - p(-1|x)$ we get the optimal function f_p given by

$$f_p(x) = \frac{1}{2}log\left(\frac{p(1|x)}{1 - p(1|x)}\right) \tag{1.2}$$

Logistic loss $l(-yf(x)) = log(1 + exp(-yf(x)))$

$$\begin{aligned}\mathcal{E}(f) &= \int_{X\times Y} log(1+exp(-yf(x)))p(x)p(y|x)\,dx\,dy \\ &= \int_X p(x)[log(1+exp(f(x)))p(-1|x) + log(1+exp(-f(x))p(1|x)]\,dx\end{aligned}$$

Taking derivative of the term inside, over $f(x)$, and setting it to 0 , we get

$$\frac{exp(f(x))}{1+exp(f(x))}p(-1|x) - \frac{exp(-f(x))}{1+exp(-f(x))}p(1|x) = 0$$

Solving this and then using the fact that $p(-1|x) = 1 - p(-1|x)$ we get the optimal function f_p given by

$$f_p(x) = log\left(\frac{p(1|x)}{1-p(1|x)}\right) \tag{1.3}$$

Hinge loss $l(-yf(x)) = |1-yf(x)|_+$

$$\begin{aligned}\mathcal{E}(f) &= \int_{X\times Y} |1-yf(x)|_+ p(x)p(y|x)\,dx\,dy \\ &= \int_X p(x)[|1+f(x)|_+ p(-1|x) + |1-f(x)|_+ p(1|x)]\,dx\end{aligned}$$

For this case, notice that the inner function is not a differentiable function of f and is piecewise linear. We can however minimize it by first partitioning the expected error term in the following way.

$$\begin{aligned}\mathcal{E}(f) = &\int_{X_1} (1-f(x))p(1|x)]\,dx \\ &+ \int_{X_2} (1+f(x))p(-1|x)]\,dx \\ &+ \int_{X_3} p(x)[(1+f(x))p(-1|x) + (1-f(x))p(1|x)]\,dx\end{aligned}$$

$$\begin{aligned}X_1 &= \{x|f(x) \leq -1\} \\ X_2 &= \{x|-1 < f(x) < 1\} \\ X_3 &= \{x|f(x) \geq 1\}\end{aligned}$$

Observe that in region X_1, irrespective of the value of $p(1|x)$, we would like our function to have a value of exactly -1 , as that minimizes the integral. Now, to decide what region X_1 is to be. Notice that for this region only the $p(1|x)$ remains, so we would want it to only be when $p(-1|x) > p(1|x)$. Similarly for X_3 where we want $p(-1|x) \leq p(1|x)$ and here we set the value of the optimal function to be 1.

$$f_p(x) = \begin{cases} -1 & p(1|x) \leq p(-1|x) \\ 1 & p(-1|x) \leq p(1|x) \end{cases}$$

Misclassification loss $\Theta(-yc(x))$

$$\begin{aligned} \mathcal{E}(f) &= \int_{X \times Y} \Theta(-yc(x)) p(x) p(y|x) \, dx \, dy \\ &= \int_X p(x)[\Theta(c(x))p(-1|x) + \Theta(-c(x))p(1|x)] \, dx \end{aligned}$$

For this case we can minimize the loss using the observation that we can send one of the terms to 0 if its corresponding multiplier ($p(y|x)$) is big. Using this, we get the minimizing c to be

$$c_p(x) = \begin{cases} -1 & p(1|x) \leq p(-1|x) \\ 1 & p(-1|x) \leq p(1|x) \end{cases}$$

Problem 1.2

Fig. 1.1 shows all the optimal functions plotted with $P(1|x)$ as the x-axis. There are two important observations; first that optimal classifiers for exponential and logistic loss are smoother, differentiable and wouldbe easier to use in practice. Second is these loss functions provide us an understanding of how sure our algorithm is in making a particular classification, giving a higher absolute value when the prediction is more likely to be correct. The hinge loss and misclassificaion losses on the other hand lead to binary classifier functions which don't distinguish between the probability of one class being slightly more likely than the other and that class being much more likely than the other.

Problem 2

Setup of the problem I will be using the data matrix X to be

$$X = [X_1 X_2 X_3 ... X_n] \in \mathbb{R}^{d \times n} \tag{1.6}$$

where, X_i is the i^{th} d-dimensional data point.

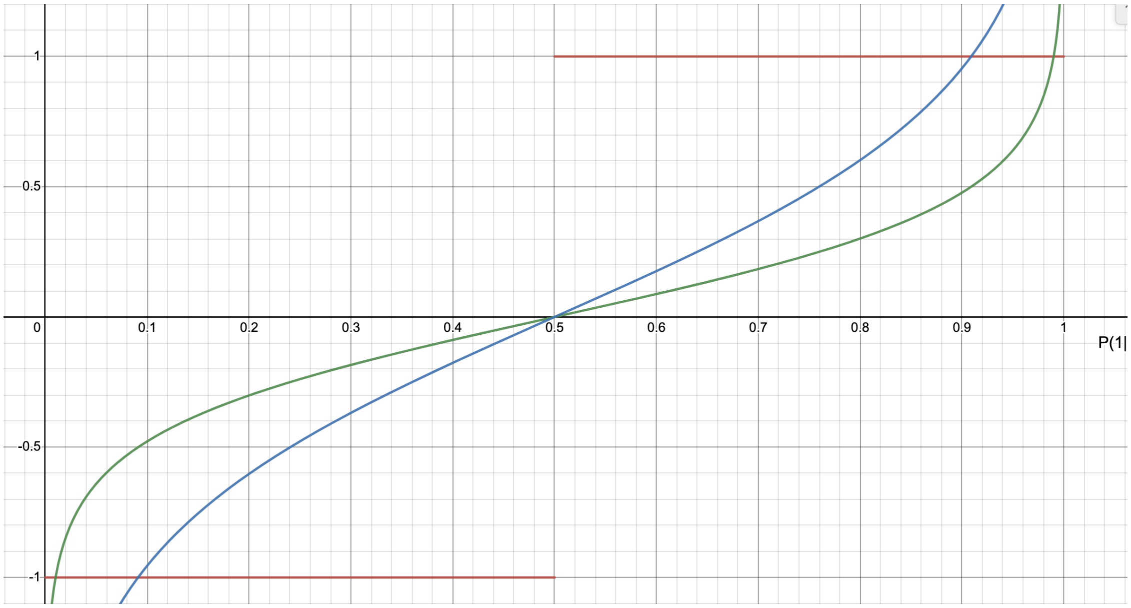

Figure 1.1. Red: Hinge loss and Misclassification loss — Green: Exponential loss — Blue: Logistic loss

Problem 2.1

To show that the solution to the mentioned minimization problem is of the form $\hat{w} = \sum_{i=1}^{n} x_i c_i$ I will consider the following 3 all encompassing cases and prove the result for all of them

- $d \leq n$ and $rank(X) = d$
- $d \leq n$ and $rank(X) = r < d$
- $d > n$ and $rank(X) = r < d$

Case 1: In this case, the matrix X is full rank and the columns of this matrix span the entire $\mathbb{R}^d$ space. This means that any $w \in \mathbb{R}^d$ can be written as a linear combination of the columns of the data matrix. This implies that even the optimal w^* can be written as a linear combination of the columns of X and the result hold trivially for this case.

Cases 2 and 3: In the case where $rank(X) = r < d$ we have to do a little more analysis. In this case, the columns of X span some r-dimensional subspace in $\mathbb{R}^d$. Let the orthogonal basis vectors of this r-dimensional subspace be $[b_1 b_2 ... b_r]$, where $b_i \in \mathbb{R}^d$. Using this we can say that any linear combination of the columns of the data matrix can be written as a linear combination of the basis vectors ie.

$$\sum_{i=1}^{n} c_i x_i = \sum_{j=1}^{r} k_j b_j$$

Now use basis extension we can say that there exist $d-r$ orthogonal vectors that together with the b_is span the entire $\mathbb{R}^d$ space. Let the set of vectors $\{b_1 b_2 ... b_r b^c_{r+1} b^c_{r+2} ... b^c_d\}$ be an orthogonal bases of $\mathbb{R}^d$, where the first r vectors are the basis of the r-dimensional subspace spanned by the columns of the data matrix. Using this we can write any $w \in \mathbb{R}^d$ as

$$w = \sum_{i=1}^{r} k_i b_i + \sum_{i=r+1}^{d} k_i b_i^c$$

Let us now denote the $w_n = \sum_{i=1}^{r} b_i$ and $w_n^{\perp} = \sum_{i=r+1}^{d} b_i^c$. Clearly, given the construction, $w_n^T w_n^{\perp} = 0$. Also note that given the construction of the bases vectors $[b_1 b_2 ... b_r]$, we can write any data point $x_i \in \mathbb{R}^d$ as a linear combination of these bases , ie.

$$x_i = \sum_{j=1}^{r} p_j^{(i)} b_j$$

where $p_j^{(i)}$ are some scalars.

Now, note that for any $w = w_n + w_n^{\perp}$ and any datapoint x_i we have $w^T x_i = w_n^T x_i$ because $(w_n^{\perp})^T x_i = 0$ by construction. From here, I will prove the result in the question by contradiction. Suppose that the minimiser of the given problem is $w^* = w_n^* + w_n^{\perp *}$. However, given our previous observation , we can say that that $w^{*T} x_i = w_n^{*T} x_i$. Now, we can see that

$$\begin{aligned} L(w^*) &= \frac{1}{n} \sum_{i=1}^{n} (y_i - w^{*T} x_i)^2 + \lambda ||w^*||^2 \\ &= \frac{1}{n} \sum_{i=1}^{n} (y_i - w_n^{*T} x_i)^2 + \lambda (||w_n^*||^2 + ||w_n^{\perp *}||^2) \\ &\geq \frac{1}{n} \sum_{i=1}^{n} (y_i - w_n^{*T} x_i)^2 + \lambda ||w_n^*||^2 \\ &= L(w_n^*) \end{aligned}$$

which shows, that w^* cannot be the minimizer because its corresponding w_n^* will always have a smaller value for the objective function. This shows that the optimal weight will have only the w_n component ie.

$$w^* = \sum_{i=1}^{r} k_i b_i = \sum_{i=1}^{n} c_i x_i$$

as b_is are the orthogonal basis of the subspace spanned by the columns of the data matrix. QED

Problem 2.2

The proof for this question is almost identical as that of the previous part. The only key part to note is, that given l is convex, and the regularizer is strongly convex, the objective function is strongly convex. This means that there exists a unique minimiser w^* for this objective function. From here, we can proceed in the same fashion as above, using the observation that $w^{*T} x_i = w_n^{*T} x_i$.

Problem 2.3

Re-writing the objective in the question in matrix/vector form for convenience.

$$\min_{c \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^{n} l(y_i, c^T(X^T x_i)) + \lambda ||Xc||_2^2 \tag{1.7}$$

The logistic loss and its gradient wrt c are given by

$$l_{logistic}(i) = log(1 + exp(-y_i(c^T X^T x_i)))$$

$$\nabla_c l_{logistic}(i) = \frac{-y_i(X^T x_i)}{1 + exp(y_i(c^T X^T x_i))} \in \mathbf{R}^n$$

$$\nabla_c(||X_c||_2^2) = 2\lambda X^T X c$$

With this, we can write the gradient descent algorithm for c as the following

$$c_{t+1} = c_t - \eta_c(-\frac{1}{n} \sum_{i=1}^{n} \frac{y_i X^T x_i}{1 + exp(y_i c^T x_i)} + 2\lambda X^T X c)$$

The time complexity of this is $O(n^2 d + n^2)$ the first term is for computing $X^T X$ and the second is for computing $c^T X^T X$. The first term of the complexity can be dropped however, if we assume that $X^T X$ is pre-computed and stored to be used for all iterations. The space complexity is $O(n^2 + nd)$ which is for storing the matrices $X^T X$ and X. However, the second term can be dropped with the observation that X appears in the gradient descent only through $X^T X$.

For the gradient descent algorithm derived in class over w we have the time complexity $O(nd)$ due to the computation of $w^T X$. The space complexity is also $O(nd)$ which is the space required to store the data.

The first approach using **c** would be useful when the dimension of the data d is very large or made larger using features. The space complexity in this case grows only with the number of training points used.

Problem 3

For the setup of this problem, we know that $\Phi(x), \Phi(x') \in \mathcal{F}$. Given that it is a feature space induced by a kernel, we know the following :

- Inner product $< \Phi(x), \Phi(x') >_{\mathcal{F}} = K(x, x')$ is defined
- norm $||\Phi(x)||^2_{\mathcal{F}} = K(x, x)$ is defined
- Distance between 2 points is defined by $d_K(x, x') = ||\Phi(x) - \Phi(x')||_{\mathcal{F}}$

Problem 3.1

Using the above, we can try to simplify the term for $d_K(x, x')$

$$\begin{aligned}
d_K^2(x, x') &= ||\Phi(x) - \Phi(x')||^2_{\mathcal{F}} \\
&=< \Phi(x) - \Phi(x'), \Phi(x) - \Phi(x') > \\
&=< \Phi(x), \Phi(x) > + < \Phi(x'), \Phi(x') > -2 < \Phi(x), \Phi(x') > \\
&= K_d(x, x) + K_d(x', x') - 2K_d(x, x') \\
d_K(x, x') &= \sqrt{K_d(x, x) + K_d(x', x') - 2K_d(x, x')}
\end{aligned}$$

Problem 3.2

Aim is to derive a classification rule based on distances from the means in feature space of the positive and negative examples.

Let S_+ be the set of indices for which the data points lie in the positive class and S_- be the indices for which the datapoints lie in the negative class. Using this notation, we can define the means in feature space as :

$$\mu_+^{\mathcal{F}} = \frac{1}{n_+} \sum_{i \in S_+} \Phi(x_i)$$
$$\mu_-^{\mathcal{F}} = \frac{1}{n_-} \sum_{i \in S_-} \Phi(x_i)$$

Now, given a a new datapoint x we want to be able to be able to classify it either to the positive or the negative class. Define distances to the positive and negative class means in feature space respectively.

$$d_{\mu_+^{\mathcal{F}}} = ||\Phi(x) - \mu_+^{\mathcal{F}}||_{\mathcal{F}}$$
$$d_{\mu_-^{\mathcal{F}}} = ||\Phi(x) - \mu_-^{\mathcal{F}}||_{\mathcal{F}}$$

The decision rule would be to assign the new data point to the class for which the distance to the mean defined above is smaller. I will now simplify the above distance equations to give an explicit form of the classifier.

$$\begin{aligned} d^2_{\mu_+^{\mathcal{F}}} &=< \Phi(x) - \mu_+^{\mathcal{F}}, \Phi(x) - \mu_+^{\mathcal{F}} > \\ &= K(x,x) + \frac{1}{n_+^2} \sum_{i,j \in S_+} K(x_i, x_j) - \frac{2}{n_+} \sum_{i \in S_+} K(x, x_i) \end{aligned}$$

Similarly, we get

$$d^2_{\mu_-^{\mathcal{F}}} = K(x,x) + \frac{1}{n_+^2} \sum_{i,j \in S_-} K(x_i, x_j) - \frac{2}{n_+} \sum_{i \in S_-} K(x, x_i)$$

The classifier can be written as

$$\frac{1}{n_+^2} \sum_{i,j \in S_+} K(x_i, x_j) - \frac{2}{n_+} \sum_{i \in S_+} K(x, x_i) \lessgtr^{x \in S_+}_{x \in S_-} \frac{1}{n_+^2} \sum_{i,j \in S_-} K(x_i, x_j) - \frac{2}{n_+} \sum_{i \in S_-} K(x, x_i)$$

Problem 4

Problem 4.1

To prove the first part of this question, it is sufficient to show that the the equations that we get by setting $w = T(w)$ and by setting $\nabla_w \frac{1}{2n} ||Y - Xw||_2^2 = 0$ are identical.

$$\begin{aligned} w &= T(w^*) \\ w &= w - \frac{1}{n} X^T (Xw - Y) \\ X^T(XW - Y) &= 0 \\ \nabla_w \frac{1}{2n} ||Y - Xw||_2^2 &= 0 \\ \frac{1}{n} X^T (Y - Xw) &= 0 \end{aligned}$$

Clearly, the two equations are the same, which means w being a fixed point and being the empirical risk minimizer imply each other. Now, to prove that T is a contraction. For this, we need to show that $\exists L < 1$ s.t. $||T(w) - T(w')||_2 \leq L||w - w'||$. Start with

$$\begin{aligned}||T(w)-T(w')||_2 &= ||(I-\frac{X^TX}{n})(w-w')||_2\\ &\leq ||I-\frac{X^TX}{n}||_2||w-w'||_2\end{aligned}$$

The above is true by the definition of the operator norm ; $||A||_2 = \max_{||X||_2=1}||Ax||_2$. $w-w'$ does not necessarily having norm equal to 1. So, we normalize it by its norm, to make the resulting vector unit norm. So, we get

$$\begin{aligned}||(I-\frac{X^TX}{n})\frac{(w-w')}{||(w-w')||_2}||_2||(w-w')||_2 &\leq (\max_w ||(I-\frac{X^TX}{n})\frac{(w-w')}{||(w-w')||_2}||_2)||(w-w')||_2\\ &= ||I-\frac{X^TX}{n}||_2||w-w'||_2\end{aligned}$$

by the definition of the operator norm. So, the map T is a contraction, where the contraction is defined by L which is the maximum singular value of the matrix $I-\frac{X^TX}{n}$. Now, all that is left to be shown is that the maximum singular value of $I-\frac{X^TX}{n}$ is smaller than one, where the rows of the matrix X are given by the input data vectors, which have a norm $||x|| \leq 1$. The following are some useful linear algebra facts that I will use to bound the above equation:

$$\begin{aligned}\sum_{i=n}^{n}\lambda_i &= tr(\frac{X^TX}{n})\\ &= tr(\frac{XX^T}{n})\\ &= \frac{\sum_{i=1}^{n}||x_i||^2}{n}\\ &\leq \frac{n}{n}\\ &= 1\end{aligned}$$

where, λ_is are the eigenvalues of $\frac{X^TX}{n}$. We also know that X^TX by its form is a symmetric positive-definite matrix. This implies all its eigenvalues are real and positive. This combined with the previous inequality showing the sum of eigenvalues to be less than equal to 1 implies that all the eigenvalues of this matrix are smaller than 1 (with the exception of the largest eigenvalue which may be equal to 1 .But, to find the constant L , we want to find the operator norm for $I-\frac{X^TX}{n}$. The eigenvalues $\bar{\lambda_i}$ of this matrix are given by:-

$$\begin{aligned}\bar{\lambda_i} &= 1 - \lambda_i \\ ||I - \frac{X^TX}{n}||_2^2 &= \bar{\lambda}_{max} \\ &= 1 - \lambda_{min} \\ &< 1\end{aligned}$$

The strict inequality in the last step is due to the fact that the minimum eigenvalue λ_{min} is strictly less than one and has to be strictly greater than 0 because XX^T is positive definite. This completes the proof that T is a contraction.

$$\begin{aligned}||T(w) - T(w')||_2 &= ||(I - \frac{X^TX}{n})(w - w')||_2 \\ &\leq ||I - \frac{X^TX}{n}||_2||w - w'||_2 \\ &\leq \sqrt{(1 - \lambda_{min})}||w - w'||_2\end{aligned}$$

Problem 4.2

In this problem, we need to show that the gradient is Lipschitz continuous and in the process, find out what the Lipschitz constant is. This can be done by upper bounding the quantity $||\nabla_w(\mathbb{L}(w) - \nabla_w\mathbb{L}(w')||$ using some matrix inequalities.

$$\begin{aligned}||\nabla_w(\mathbb{L}(w) - \nabla_w\mathbb{L}(w')||_2 &= ||\frac{X^TXw}{n} - \frac{X^TY}{n} - \frac{X^TXw'}{n} + \frac{X^TY}{n}||_2 \\ &= ||\frac{X^TX}{n}(w - w')||_2 \\ &\leq \frac{||X^TX||_2}{n}||w - w'||_2 \\ &\leq \sqrt{\lambda_{max}}||w - w'||_2 \\ &\leq ||w - w'||_2\end{aligned}$$

where, λ_{max} is the maximum eigenvalue of $\frac{X^TX}{n}$ and the last inequality of it being smaller than one and positive is proven in the previous part. Clearly, the Lipschitz constant $L = 1$ and we can choose the step-size γ to be

$$\gamma = \frac{2}{L} = 1$$

Problem 5

For this question, first re-write the objective in matrix form for convenience. Define a matrix Γ , such that it is a diagonal matrix with entries $\Gamma_{ii} = \gamma_i$.

$$\Gamma = \begin{bmatrix} \gamma_1 & & \\ & \ddots & \\ & & \gamma_n \end{bmatrix}$$

Problem 5.1

Using this and the fact that X is the data matrix with rows corresponding to individual input vectors x_i, we can rewrite the objective as:

$$\min_{w \in \mathcal{R}^d} (Xw - Y)^T \Gamma (Xw - Y) + \lambda w^T w$$

Now, to derive the explicit form of the minimizer, all we need to do is to set the gradient with respect to w of the objective to be zero.

$$\begin{aligned} \nabla_w \mathbb{L}(w) &= X^T \Gamma (Xw - Y) + \lambda w \\ &= 0 \\ w^* &= (X^T \Gamma X + \lambda I)^{-1} X^T \Gamma Y \end{aligned}$$

Problem 5.2

For this problem also, we take the same approach as above. For convenience, let $e_n = [1, 1....1]^T \in \mathcal{R}^n$,. Using this, we re-write the objective as

$$\min_{w \in \mathcal{R}^d, b \in \mathcal{R}} (Xw + be_n - Y)^T \Gamma (Xw + be_n - Y) + \lambda w^T w$$

The explicit minimizers for this can be found by setting the gradients wrt w and b to be equal to zero.

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial b} &= 0 \\ e_n^T \Gamma (Xw^* + b^* e_n - Y) &= 0 \\ \nabla_w \mathbb{L}(w) &= 0 \\ X^T \Gamma (Xw^* + b^* e_n - Y) + \lambda w^* &= 0 \end{aligned}$$

Solving these equations simultaneously for w^* and b^* , we get the following explicit forms for the minimizers.

$$
\begin{aligned}
w^* &= (X^T \Gamma X + \lambda I - \frac{X^T \Gamma e_n^T e_n \Gamma X}{n})^{-1} (X^T - \frac{X^T \Gamma e_n^T e_n \Gamma Y}{n}) \\
b^* &= \frac{1}{n} (e_n^T \Gamma (Y - X w^*))
\end{aligned}
$$