CS689: Machine Learning

Fall 2021 - Midterm Exam

Name: ____________________________________

Instructions: Write your name on the exam sheet. The duration of this exam is **two hours**. No electronic devices may be used during the exam. You may consult either your paper notes and/or a print copy of texts during the exam. Sharing of notes/texts during the exam is strictly prohibited. Show your work for all derivation questions. Provide answers that are as detailed as possible for explanation questions. Attempt all problems. Partial credit may be given for incorrect or incomplete answers. If you need extra space for answers, write on the back of the **preceding** page. If you have questions at any time, please raise your hand.

Problem	Topic	Page	Points	Score
1	Loss Functions	1	10	
2	Gradient Descent	3	10	
3	Regularized Risk Minimization	5	10	
4	Basis Expansion	7	10	
5	Bias	9	10	
6	Regularization Dynamics	11	10	
7	Sub-Differentials	13	10	
8	Lagrangian Functions	15	10	
9	Lagrange Duals	17	10	
10	Modeling	19	10	
Total:			100	

1. (*10 points*) **Loss Functions**. Consider the function $f(z)$ given below when answering the following questions.

$$f(z) = \begin{cases} 1 - 2z & z < 0 \\ (z-1)^2 & 0 \leq z < 1 \\ 0 & 1 \leq z \end{cases}$$

a. (*5 pts*) What is the derivative of $f(z)$? Show your work.

Example Solution:

$$\frac{\partial}{\partial z} f(z) = \begin{cases} \frac{\partial}{\partial z}(1 - 2z) & z < 0 \\ \frac{\partial}{\partial z}\,(z-1)^2 & 0 \leq z < 1 \\ \frac{\partial}{\partial z}\,0 & 1 \leq z \end{cases} = \begin{cases} -2 & z < 0 \\ 2(z-1) & 0 \leq z < 1 \\ 0 & 1 \leq z \end{cases}$$

b. (*5 pts*) Let the function $h(z) = [z \leq 0]$ be equal to 1 when z is less than or equal to 0 and let it be 0 otherwise. Explain the significance of the fact that $f(z) \geq h(z)$ when considering using $l(y, g_\theta(\mathbf{x})) = f(y \cdot g_\theta(\mathbf{x}))$ as a classification loss function.

Example Solution: If $f(z) \geq h(z)$ then $l(y, g_\theta(\mathbf{x})) \geq h(y \cdot g_\theta(\mathbf{x})) = [y \neq \text{sign}(g_\theta(\mathbf{x}))]$. In other words, the loss induced by $l(y, g_\theta(\mathbf{x}))$ upper bounds the classification error rate. Thus, if we select $\hat{\theta}$ by minimizing $l(y, g_\theta(\mathbf{x}))$, the classification error on the training set will be no larger than $l(y, g_{\hat{\theta}}(\mathbf{x}))$.

2. (*10 points*) **Gradient Descent**. Consider the learning problem given below for the linear regression model $f_\theta(\mathbf{x}) = \mathbf{x}\mathbf{w} + b$ with $\theta = [\mathbf{w}, b]$ when answering the following questions.

$$\hat{\theta} = \arg\min_{\theta} \sum_{n=1}^{N} (y_n - f_\theta(\mathbf{x}_n))^2$$

a. (*5 pts*) Give pseudo code for a gradient descent algorithm for finding $\hat{\theta}$, including the computation of the gradient.

Example Solution:

```
0: Inputs: $\mathcal{D}$, $\alpha$, $T$
1: Initialize $\theta = 0$
2: for $i$ from 1 to $T$:
    2.1: $\mathbf{g} \leftarrow -2\sum_{n=1}^{N}(y_n - \mathbf{x}_n\theta)\mathbf{x}_n^T$
    2.2: $\theta \leftarrow \theta - \alpha\mathbf{g}$
```

b. (*5 pts*) Give two reasons why we might prefer to use a gradient descent algorithm to solve the OLS learning problem instead of the closed-form estimator $\hat{\theta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$.

Example Solution: First, if D is larger than N, then the OLS estimator requires $O(D^3)$ time. The gradient-based solution can be significantly more computationally efficient in this case. Second, if $\mathbf{X}^T\mathbf{X}$ is not invertible, then the OLS estimator can not be computed, but the gradient-based solution will still yield the global minimum.

3. (*10 points*) **Regularized Risk Minimization**. Consider the linear regression model $f_\theta(\mathbf{x}) = \mathbf{x}\mathbf{w} + b$ with $\theta = [\mathbf{w}, b]$.

a. (*5 pts*) Give the regularized risk minimization problem when fitting the model $f_\theta(\mathbf{x})$ using the absolute loss and squared two norm regularizer. Explain your answer.

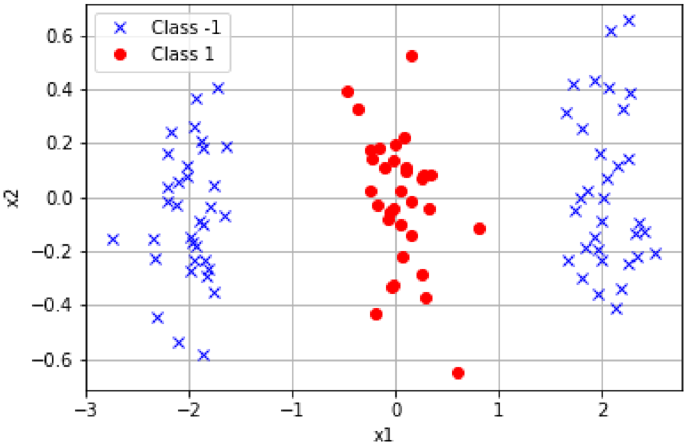
Example Solution: The absolute loss function is $|y_n - f_\theta(\mathbf{x}_n)|$. The squared two norm regularizer is $\|\theta\|_2^2$ or $\|\mathbf{w}\|_2^2$ when not regularizing the bias. In the solution shown below we do not regularize the bias, but either answer is acceptable. The regularized risk minimization problem requires combining the sum of the loss over the data set and the regularizer using a weight on either the loss or regularization term. We weight the regularization term below, but either is acceptable. The minimization problem is to find the value of θ that minimizes this function.

$$\theta = \arg\min_\theta \sum_{n=1}^{N} |y_n - f_\theta(\mathbf{x}_n)| + \lambda \|\mathbf{w}\|_2^2$$

b. (*5 pts*) Explain why we should not use regular gradient descent to solve this regularized risk minimization problem.

Example Solution: Since the loss term is not differentiable, standard gradient descent may not converge.

4. (*10 points*) **Basis Expansion**. Consider the data set shown below where $\mathbf{x} \in \mathbb{R}^2$. Give a minimal basis expansion (a basis expansion with the fewest components) that will perfectly classify these data when using a linear classifier in the basis expanded space. Explain your answer.



Example Solution: Consider the basis expansion $\phi(\mathbf{x}) = x_1^2$. Since all of the data in class 1 satisfy $x_{1n}^2 \leq 1$ and all of the data in class -1 satisfy $x_{1n}^2 > 1$, the discriminant function $g_\theta(\phi(\mathbf{x})) = \phi(\mathbf{x})w + b$ will perfectly classify these data with $w = -1$ and $b = 1$. This must be the smallest basis expansion since it consists of only one component.

5. (*10 points*) **Bias**. Consider a regression problem where $\mathbf{X}$ is the matrix of feature vectors and $\mathbf{Y}$ is the vector of targets. Recall that the OLS estimator for the linear regression model given these data is $\hat{\theta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$ (assume $\mathbf{X}^T\mathbf{X}$ is invertible).

Suppose we re-scale $\mathbf{Y}$ forming new targets $\mathbf{Z} = s\mathbf{Y}$ for $s > 0$. Consider the estimator $\hat{\phi} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Z}$. Under the assumption that $E_{p(\mathbf{Y}|\mathbf{X})}[\mathbf{Y}] = \mathbf{X}\theta_*$, prove that $\hat{\phi}/s$ is an unbiased estimator for θ_*.
Example Solution: For $\hat{\phi}/s$ to be an unbiased estimator for θ_* we must have that $\mathbb{E}_{p(\mathcal{D})}[\hat{\phi}/s] = \theta_*$. We proceed as follows:

$$\begin{aligned}
\mathbb{E}_{p(\mathcal{D})}[\hat{\phi}/s] &= \frac{1}{s}\mathbb{E}_{p(\mathcal{D})}[\hat{\phi}] \\
&= \frac{1}{s}\mathbb{E}_{p(\mathcal{D})}[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Z}] \\
&= \frac{1}{s}\mathbb{E}_{p(\mathcal{D})}[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T(s\mathbf{Y})] \\
&= \frac{s}{s}\mathbb{E}_{p(\mathcal{D})}[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{Y})] \\
&= \mathbb{E}_{p(\mathbf{X})}[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbb{E}_{p(\mathbf{Y}|\mathbf{X})}[(\mathbf{Y})]] \\
&= \mathbb{E}_{p(\mathbf{X})}[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{X}\theta_*] \\
&= \mathbb{E}_{p(\mathbf{X})}[I\theta]_* \\
&= \theta_*
\end{aligned}$$

6. (*10 points*) **Regularization Dynamics**. Recall the the linear classifier is given by $f_\theta(\mathbf{x}) = \text{sign}(g_\theta(\mathbf{x}))$, $g_\theta(\mathbf{x}) = \mathbf{x}\mathbf{w} + b$, $\theta = [\mathbf{w}, b]$. Consider the regularized logistic regression learning problem for the linear classifier shown below when answering the following questions.

$$\hat{\theta} = \arg\min_{\theta} \sum_{n=1}^{N} \log(1 + \exp(-y_n g_\theta(\mathbf{x}_n)) + \lambda \|\mathbf{w}\|_2^2$$

a. (*5 pts*) Suppose we fit the model with $\lambda = 1$ and find that the train error is much lower than the test error. Should we increase or decrease λ to try to improve the test error? Explain your answer.

Example Solution: If the training error is much lower than the test error, then the most likely cause is that the model is overfit. To reduce overfitting we need to increase the amount of regularization. For a problem formulation where the regularization term has the weight on it, we need to increase this weight to increase the amount of regularization. Thus, we should increase the value of λ.

b. (*5 pts*) Suppose that we run a grid search using a validation set split from the training set. However, at the optimal value of λ the error on the test set is still much higher than the error on the training set. Assuming that all implementations are correct, explain how this might happen.

Example Solution: The fundamental reason that this could happen is that the distribution of the training/validation data and the distribution of the test set are very different. This can happen when all data sets are sampled from p_*, but at least one of them is small so that they are not representative of p_* by chance. The other potential cause is that the test data are sampled from a different distribution than the training and validation data. In this case, no matter how much data we have, selecting hyper-parameters and learning models on the train and validation sets can lead to arbitrarily poor performance on the test set.

7. (*10 points*) **Sub-Differentials**. Consider the function $f(z) = \max(0, 1 - z)$. Derive the sub-differential of $f(z)$. Show your work and/or explain your answer.

Example Solution: This is the hinge loss function. It has one point of non-differentiability at $z = 1$. For $z > 1$, the function is constant at $f(z) = 0$ and the derivative is $d/dz\,(0) = 0$, so the sub-differential is the singleton set $\partial f(z) = \{0\}$. When $z < 1$ we have $f(z) = 1 - z$. The derivative is thus $d/dz\,(1 - z) = -1$ and the sub-differential is the singleton set $\partial f(z) = \{-1\}$. At 1, the sub-differential is the closed interval $\partial f(1) = [-1, 0]$ since the maximum derivative of the function on the left branch at $z = 1$ is -1, and the derivative of the function on the right branch at $z = 1$ is 0. The complete sub-differential is given below.

$$\partial f(z) = \begin{cases} \{-1\} & z < 1 \\ [-1, 0] & z = 1 \\ \{0\} & z > 1 \end{cases}$$

8. (*10 points*) **Lagrangian Functions**. Recall that the linear classifier is given by $f_\theta(\mathbf{x}) = \text{sign}(g_\theta(\mathbf{x}))$, $g_\theta(\mathbf{x}) = \mathbf{x}\mathbf{w} + b$, $\theta = [\mathbf{w}, b]$. Consider the SVC learning problem for the linear classifier shown below.

$$\hat{\theta} = \arg\min_\theta \; C\sum_{n=1}^{N} \max(0, 1 - y_n g_\theta(\mathbf{x}_n)) + \|\mathbf{w}\|_2^2$$

Suppose we need to solve this problem with the additional simplex constraints $\forall\, d\; w_d \geq 0$ and $\sum_{d=1}^{D} w_d = 1$. Give a Lagrangian function for this problem and specify any constraints on the Lagrange multipliers. Show your work and/or explain your answer.

Example Solution: To form the Lagrangian, we need to introduce but the constraints in canonical form, introduce a Lagrange multiplier for each constraint function, then subtract off the sum of the Lagrange multipliers times the constraint functions from the primal objective functions. We let $c_0(\theta) = (\sum_{d=1}^{D} w_d - 1)$ be the first constraint function (after changing to the given equality constraint to canonical form). We let $c_d(\theta) = w_d$ be constraint functions derived from the inequality constraints for $1 \leq d \leq D$. We let λ_i be the Lagrange multipliers. We need to include the constraints $\lambda_d \geq 0$ for the inequality constraints (e.g., $1 \leq d \leq D$). This gives us the following Lagrangian function and constraints:

$$\mathcal{L}(\mathbf{w}, b, \lambda) = C\sum_{n=1}^{N} \max(0, 1 - y_n g_\theta(\mathbf{x}_n)) + \|\mathbf{w}\|_2^2 - \lambda_0(\sum_{d=1}^{D} w_d - 1) - \sum_{d=1}^{D} \lambda_d w_d$$

$$\text{s.t. } \lambda_d \geq 0 \text{ ... for } 1 \leq d \leq D$$

9. (*10 points*) **Lagrange Duals**. Consider the Lagrangian function given below. x and y are the primal parameters and λ is the Lagrange multiplier. Derive the Lagrange dual function and provide the Lagrange dual optimization problem. Show your work and explain your answer.

$$\mathcal{L}(x, y, \lambda) = x^2 + y^2 - \lambda(x + y - 2)$$

Example Solution: We need to take derivatives with respect to the primal variables, set them equal to zero, and then attempt to eliminate the primal parameters from the Lagrangian. We have:

$$\frac{\partial}{\partial x}\mathcal{L}(x, y, \lambda) = 2x - \lambda = 0$$
$$\frac{\partial}{\partial y}\mathcal{L}(x, y, \lambda) = 2y - \lambda = 0$$

These equations imply that $x = \lambda/2$ and $y = \lambda/2$. We can substitute these equations into the Lagrangian to form the dual $q(\lambda)$. We have:

$$\begin{aligned} q(\lambda) &= \left(\frac{\lambda}{2}\right)^2 + \left(\frac{\lambda}{2}\right)^2 - \lambda\left(\left(\frac{\lambda}{2}\right) + \left(\frac{\lambda}{2}\right) - 2\right) \\ &= \frac{\lambda^2}{4} + \frac{\lambda^2}{4} - \lambda^2 + 2\lambda \\ &= -\frac{\lambda^2}{2} + 2\lambda \end{aligned}$$

10. (*10 points*) **Modeling**. In some real-world supervised learning problems, the targets are semi-continuous. This means that for some fraction p of the data cases, the target values y_n are exactly 0. For the remainder of the cases, the target values y_n are strictly positive real numbers. Which cases have targets equal to 0 and which do not typically depends on the values of the feature variables $\mathbf{x}_n$. The target values not exactly 0 are also dependent on the values of the features. Describe how we could use the models we have seen to date to provide a prediction function for this problem. Briefly explain how you would learn the parameters of the prediction function.

Example Solution: We can use a two-stage approach to this problem. We can first use a classification model $f_\theta(\mathbf{x})$ to predict whether a given input $\mathbf{x}$ will have a target value of exactly 0 or not. We will define an auxiliary classification variable z_n to be -1 if $y_n = 0$. We will let $z_n = 1$ if $y_n > 0$. We will learn the classification model parameters using the data set $\mathcal{D}_C = \{(\mathbf{x}_n, z_n) | 1 \leq n \leq N\}$, obtaining estimated model parameters $\hat{\theta}$. Once the model is trained, we will use it to make prediction $\hat{z} = f_{\hat{\theta}}(\mathbf{x})$. We could use a logistic regression classifier or an SVC model for this component.

In the second stage, we will learn a regression model $h_\phi(\mathbf{x}_n)$ to predict the target values $y_n > 0$. To train this model, we will form the data set $\mathcal{D}_R = \{(\mathbf{x}_n, y_n) | 1 \leq n \leq N, y_n > 0\}$ consisting of data cases with $y_n > 0$. We will learn the model obtaining the parameters $\hat{\phi}$. We will make predictions using $\hat{y} = h_{\hat{\phi}}(\mathbf{x})$. We could use a linear regression model for this task trained using squared loss, absolute loss or epsilon insensitive loss.

Our complete prediction function for semi-continuous targets is the model shown below.

$$sc_{\theta,\phi}(\mathbf{x}) = \begin{cases} 0 & \text{... if } f_\theta(\mathbf{x}) = -1 \\ h_{\hat{\phi}}(\mathbf{x}) & \text{... if } f_\theta(\mathbf{x}) = 1 \end{cases}$$

If the relationship between z and $\mathbf{x}$ or y and $\mathbf{x}$ is non-linear, we could use a basis expansion or kernel instead of a model that is linear in the original feature space. We would train the model using regularized risk minimization using the squared two norm regularizer. We will select the regularization parameters for both models using a validation set held out from the training set.