DS-GA-1003: Machine Learning (Spring 2023)

Midterm Exam (4:55pm–6:35pm, March 7)

Answer the questions in the spaces provided. If you run out of room for an answer, use the blank page at the end of the test.

Name: _____

NYU NetID: _____

Question	Points	Score
Generalization	15	
Optimization	15	
Regularization	10	
SVM	15	
Kernels	15	
Total:	70	

1. **Generalization and risk decomposition.** Sara is a data scientist who works for a hospital, and she is tasked with building a model to predict which patients are likely to develop diabetes. She has a dataset that contains information about the patient's age, BMI, blood pressure, glucose level, and other relevant factors.

   To begin her work, Sara must decide which machine learning algorithm to use. She knows that there are two main types of machine learning: supervised and unsupervised learning.

   (a) (3 points) What are the differences between supervised and unsupervised learning, and in what situations would each type be appropriate?

   Solution: Supervised learning involves training a model on labeled data, where the input data is accompanied by corresponding output labels. The goal is for the model to learn a mapping between the input and output variables so that it can predict the output for new, unseen input data.

   On the other hand, unsupervised learning involves training a model on unlabeled data where there are no output labels. The goal is typically to find patterns or structures in the data, such as clustering similar data points together or reducing the dimensionality of the data.

   Supervised learning is generally better when we have a clear idea of the input-output mapping we want to learn and have labeled data to train on. This is often the case in tasks like image classification, speech recognition, or sentiment analysis.

   Unsupervised learning is better when we have a large amount of unlabeled data and want to extract meaningful patterns or structure from it. This can be useful for tasks like anomaly detection, customer segmentation, or data compression.

After some research, Sara decided to use a supervised learning algorithm. Before training her model, Sara splits her dataset into training, validation, and test sets.

(b) (4 points) What is the purpose of splitting a dataset into training, validation, and test sets, and how does this affect the estimation of the generalization error? **Briefly** explain using the definition of generalization error.

Solution: Splitting a dataset into training, validation, and test sets is done to evaluate the model's performance on new data. The training set is used to train the model, while the validation set is used to tune hyperparameters and prevent overfitting. A separate test set is used for unbiased evaluation of the model's performance on unseen data. Overall, this process helps to ensure that the model generalizes well and provides an accurate estimate of its performance on new data

Next, Sara wants to explore her model and its error.

(c) (4 points) Explain the concept of a hypothesis class, how it relates to approximation error in machine learning, and how it relates to the model's ability to fit the true underlying function.

Solution: A hypothesis class is a set of functions that a model can choose from to approximate the true underlying function. The choice of hypothesis class affects the approximation error and the model's ability to fit the true function. If the hypothesis class is too simple, the model has high bias; if it is too complex, it has high variance. To find the right balance, we must choose an appropriate hypothesis class by evaluating the model's performance on a validation set.

With all of these considerations in mind, Sara is ready to train her model and predict which patients are likely to develop diabetes.

(d) (4 points) What is estimation error, and how is it influenced by the complexity of the model and the amount of available training data?

Solution: The estimation error is the difference between the maximum inside the hypothesis class and the function the model chooses. It is influenced by model complexity, available training data, and hypothesis class. A more complex model can reduce the estimation error but may overfit, resulting in a higher error on new data. A simpler model may have a higher error due to underfitting. The amount of available data also affects the error. We need to balance the model complexity and available data to achieve the best results.

2. **Optimization.**

(a) (3 points) What is the difference between batch gradient descent and stochastic gradient descent, and when should one be used over the other?

Solution: Batch gradient descent updates the model using gradients of the entire dataset, making it computationally expensive but less noisy. Stochastic gradient descent uses a single training example or a small subset, making it faster but more noisy. Mini-batch gradient descent is a compromise between the two and is most commonly used in deep learning. The choice between them depends on the problem at hand. For small datasets and convex problems, batch gradient descent may be better. For large datasets and non-convex problems, stochastic gradient descent may be more appropriate. A combination of these methods is often used in practice

(b) (3 points) How does the learning rate affect the convergence of the gradient descent algorithm, and what are the advantages and disadvantages of using a higher learning rate?

Solution:

The learning rate controls the step size taken during gradient descent optimization. A higher learning rate can lead to faster convergence, but it can also cause the algorithm to overshoot the optimal solution or diverge. A lower learning rate may converge too slowly, but it is less likely to overshoot or diverge. The optimal learning rate depends on the problem at hand and may require tuning.

(c) (3 points) What is the tradeoff between using a larger or smaller mini-batch size in stochastic gradient descent, in terms of gradient estimation quality and optimization speed?

Solution: The choice of mini-batch size in SGD impacts both gradient estimation quality and optimization speed. Larger mini-batches provide a more accurate estimate of the gradient, but can lead to slower convergence due to increased computational cost. Smaller mini-batches converge faster, but with a noisier estimate of the gradient. The optimal mini-batch size depends on the problem and available computational resources.

(d) (3 points) How does the differentiability of a loss function affect the optimization process, and what methods can be used to handle non-differentiable loss functions?

Solution: The differentiability of a loss function plays a crucial role in the optimization process, especially in gradient-based optimization algorithms like stochastic gradient descent (SGD). When a loss function is differentiable, its gradient can be computed, and optimization algorithms can use it to update the parameters of a model to minimize the loss. However, if the loss function is non-differentiable, it can pose a challenge to the optimization process. Some methods to handle non-differentiable loss functions include subgradient methods and methods that approximate the loss with different differentiable one.

(e) (3 points) How do outliers affect the selection of a loss function, and how can this issue be addressed?

Solution:
Outliers can distort the objective of a model, so choosing a loss function that is less sensitive to them is important. One can use robust loss functions, data preprocessing, weighted loss functions, and ensemble methods to address this.

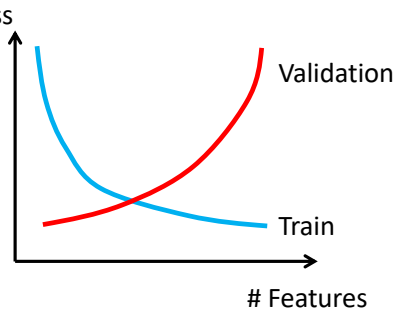
3. **Regularization.**

(a) (3 points) Suppose you are trying to choose a good subset of features for a least-squares linear regression model. Let Algorithm A be forward stepwise selection, where we start with zero features and at each step we add the new feature that most decreases validation error, stopping only when validation error starts increasing. Let Algorithm B be similar, but at each step we include the new feature that most decreases training error (measured by the usual cost function, mean squared error), stopping only when training error starts increasing. What is the relationship between the number of features that the two algorithms will end up selecting?

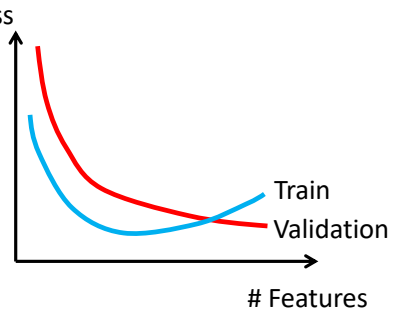
Solution: B selects more features than A.

(b) (2 points) You are selecting a subset of features for a machine learning program. Which of the following will you likely observe in terms of training and validation loss?

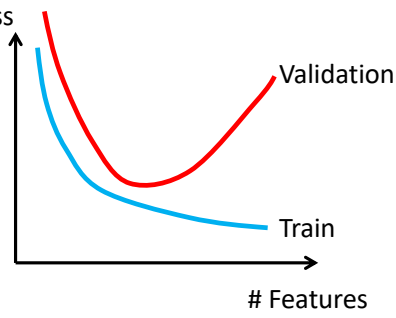
A)



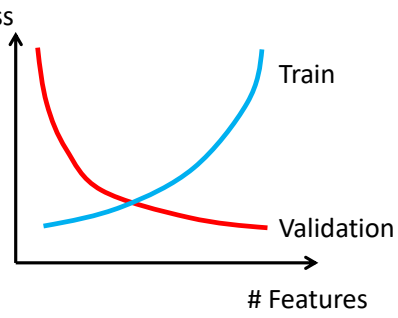
B)



C)



D)



Solution: C.

(c) (2 points) Select all true statements below.

1. Ridge regression has an analytical solution.
2. Lasso regression has an analytical solution.
3. Lasso regression tends to produce sparse solution.
4. Both L1 and L2 regularizers encourage weights to be close to zero.

Solution: 1, 3, and 4.

(d) (3 points) Suppose that you are building a binary classification for logistic regression. Recall that logistic regression loss is $L = -y_i \log(p_i) - (1 - y_i) \log(1 - p_i)$, where $p_i = \frac{1}{1+\exp(-\mathbf{w}^\top \mathbf{x}_i + b)}$.
Suppose that the dataset is linearly separable, explain why you may need to apply L2 regularization.

Solution: w would go to infinity if no L2 is applied.

4. **Support Vector Machines.** Recall the (soft-margin) SVM primal problem:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}||w||^2 + \frac{c}{n}\sum_{i=1}^{n}\xi_i \\ \text{subject to} \quad & -\xi_i \leq 0 \quad \text{for } i = 1, \ldots, n \\ & \left(1 - y_i\left[w^T x_i + b\right]\right) - \xi_i \leq 0 \quad \text{for } i = 1, \ldots, n \end{aligned}$$

and its dual problem:

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j x_j^T x_i \\ \text{subject to} \quad & \sum_{i=1}^{n}\alpha_i y_i = 0 \\ & \alpha_i \in \left[0, \frac{c}{n}\right] \quad \text{for } i = 1, \ldots, n \end{aligned}$$

(a) (1 point) The primal objective is

A. convex

B. concave

C. neither convex nor concave

(b) (1 point) The dual objective is

A. convex

B. concave

C. neither convex nor concave

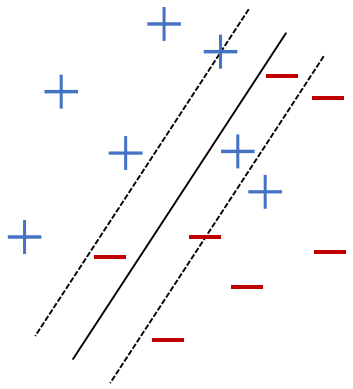
(c) (2 points) If you have high dimensional features but relatively less data points, it is more advantageous to optimize

A. the primal objective

B. the dual objective

C. either objective

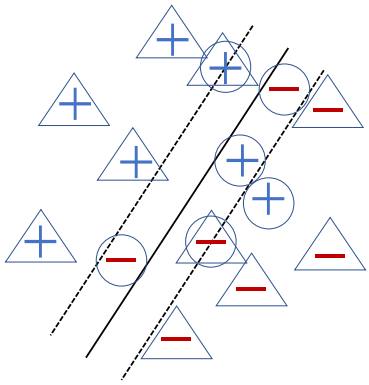
(d) Recall that given the dual solution α_i^*'s, the primal solution is given by $w^* = \sum_{i=1}^{n}\alpha_i^* y_i x_i$, and the support vectors are defined to be x_i's where $\alpha_i > 0$. The figure below shows a toy dataset and the SVM decision boundary (the solid line) with the corresponding margin (indicated by the two dotted lines).



i. (2 points) Draw a triangle around all points that have the slack variable likely to be zero ($\xi_i = 0$) (no partial credits).

ii. (2 points) Draw a circle around all points that are likely support vectors (no partial credits).

Solution:



(e) Assuming the data is not linearly separable, increasing c is likely to result in (circle the right answer)

i. (2 points) smaller / larger geometric margin

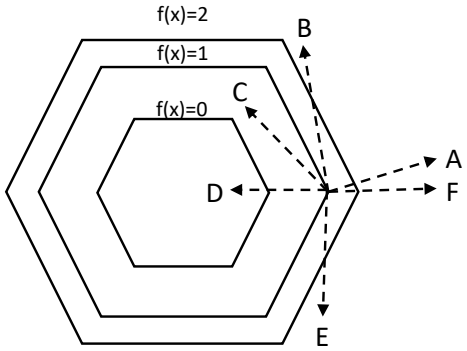
Solution: smaller

ii. (2 points) fewer / more support vectors

Solution: fewer

(f) (3 points) Circle all vectors in the following figure that is in the subdifferential.

Solution: A, F.



5. **Kernels.**

   (a) (3 points) The Gaussian kernel is a popular choice for kernel regression and is defined as:

$$K(x_i, x_j) = \exp\left(-\frac{(x_i - x_j)^2}{2h^2}\right)$$

   where x_i and x_j are two input values, and h is the kernel bandwidth or smoothing parameter.

   Discuss the bias-variance tradeoff in kernel regression. How does the choice of the kernel bandwidth affect the tradeoff? Provide an example of a situation where increasing the kernel bandwidth might improve the performance of the kernel regression model.

   Solution: Increasing the bandwidth may improve model performance when the underlying function being estimated is very smooth, preventing overfitting.

(b) (2 points) Why is it important to know that a solution is in the "span of the data"?

Solution: Knowing that a solution is in the "span of the data" means that the solution can be expressed as a linear combination of the input data. One reason why it is important to know that a solution is in the span of the data is that it allows us to represent the solution in a more efficient and interpretable way. For example, in linear regression, if the solution is in the span of the data, we can represent it as a linear combination of the input features, which makeswhich makes it easier to understand the relationship between the features and the response variable.It will be more efficient to know that a solution is in the span of the data when we are dealing with large datasets or complex models

(c) (4 points) How does the representer theorem affect the tradeoff between model complexity and generalization performance? Provide an example of a situation where the representer theorem can be used to extract useful information from a linear model.

Solution: he representer theorem allows expressing the solution of a regularized linear model as a linear combination of input data, reducing model complexity without sacrificing performance. In kernel regression, it can help identify important features or patterns in high-dimensional data relevant to the response variable, aiding understanding of the data-generating process and improving predictions.

(d) (2 points) Why we should use a feature extractor?

Solution: Feature extraction can also help to mitigate the effect of noise or irrelevant information in the data, which can improve the accuracy and generalization of the model. Additionally, feature extraction can help to improve the interpretability of the model by identifying the most important features that contribute to the model's predictions. Overall, feature extraction is a critical step in machine learning that can help to optimize the performance and interpretability of the model.

(e) (4 points) What is the difference between linear and nonlinear models, and how do kernels help in modeling nonlinear relationships?

Solution: Linear models assume a linear relationship between variables, while nonlinear models allow for more complex relationships. Kernels transform data to a higher-dimensional space, making it easier to model nonlinear relationships with linear models, such as support vector machines.

Congratulations! You have reached the end of the exam.