Midterm 3

Printed Name:

Due date:

1. The exam is due Monday 5 Dec 8AM.
2. You may submit it either on sakai electronically or by putting a physical copy under my door.

Rules:

1. The exam is untimed. You do not have to complete the exam in a single sitting. You may pause and restart whenever you'd like.
2. You may use any non-human resources that you like, including notes, books, internet articles, and computers.
3. You are not allowed to discuss the exam in any way with any human until after the due date. This includes:
   (a) obviously bad behavior like copying answers,
   (b) more banal behavior such as:
      i. telling your friend ”Problem 6 was really hard” or
      ii. asking your friend ”Have you completed the exam yet?”

   Even after you finish the exam, you may not discuss it.
4. If you do have questions about the exam, you should email me the questions rather than posting to github.

Grading:

1. For the True/False/Open questions: Each correct answer will be awarded +1 point, each incorrect answer will result in a -1 point penalty, and each blank answer will result in 0 points.
2. All other problems are worth 1 point, with no penalty for incorrect answers.
3. There are 16 points possible on the exam. Your final grade entered into sakai will be

$$\min\{15, \text{the number of points earned}\}.$$

4. If you find a substantive error on the exam, then I will award you +1 bonus point.

Good luck :)

Problem 1. For each statement below, circle `True` if the statement is known to be true, `False` if the statement is known to be false, and `Open` if the statement is not known to be either true or false. Ensure that you pay careful attention to formal definitions in your responses.

1. [✓ `True`] `False` `Open` If $|E_{\text{in}} - E_{\text{val}}|$ is large and E_{in} is small, then VC theory predicts that you should decrease the size of your VC dimension.

2. `True` [✓ `False`] `Open` If you are training a logistic regression model with the polynomial kernel that is overfitting, then VC theory predicts that increasing the degree of the kernel is more likely to improve performance than decreasing the degree of the kernel.

3. `True` [✓ `False`] `Open` You have trained a logistic regression model with L2 regularization and the polynomial kernel of degree 3. If you increase the degree of the polynomial kernel to 10, then the optimal soft order constraint regularization hyperparameter C will also increase.

4. [✓ `True`] `False` `Open` When training a neural network with the ReLU activation function and one hidden layer, increasing the width of the hidden layer will increase the generalization error.

5. [✓ `True`] `False` `Open` You have a dataset with the number of features $d = 10^6$. You have trained a boosted SVM and used a validation set to determine that the optimal number of base classifiers T is 1000. If instead of using an SVM as the base model you use a decision stump, then VC theory predicts that you will need to increase the number of base classifiers T in order to achieve the same generalization error.

6. [✓ `True`] `False` `Open` The VC dimension of neural networks with the ReLU activation function is $\Omega(d)$, where d is the number of input feature dimensions.

7. [✓ `True`] `False` `Open` Assume you are training an SVM with the polynomial kernel on a dataset with $N = 10^6$ and $d = 10^6$. You are not using any regularization, and you run the optimization long enough so that optimization error is 0. Then in the limit as the degree of the polynomial approaches infinity, the training error is guaranteed to approach 0 for all possible datasets.

8. `True` [✓ `False`] `Open` In vowpal wabbit, increasing the `--l1` hyperparameter tends to increase the generalization error.

9. `True` [✓ `False`] `Open` You have trained a scikit-learn `sklearn.linear_model.LogisticRegression` model with default hyperparameters. It has a high approximation error, low estimation error, and zero optimization error. VC theory predicts that changing the `solver` hyperparameter from the default of `'lbfgs'` to `'saga'` will improve performance.

10. `True` [✓ `False`] `Open` You are using transfer learning to train the final layer of a deep neural network for your specific task. The `ResNet18` model has 18 hidden layers and the `ResNet50` model has 50 hidden layers, and in both cases all layers have the same width. VC theory predicts that if you use the `ResNet50` model you will have a higher generalization error than if you use the `ResNet18` model.

Problem 2. Provide an example of a hypothesis class implemented by scikit learn with an infinite VC dimension. Also, describe a learning problem where it makes sense to use this hypothesis class.

Solution: The 1-nearest neighbor classifier. A nearest neighbor hypothesis h satisfies

$$E_{\text{out}}(h) \leq 2E_{\text{out}}(f) + 4\sqrt{d}N^{-\frac{1}{d+1}} \tag{1}$$

and so the model will only work will on problem with both a very low input feature dimension d and a low Bayes error $E_{\text{out}}(f)$.

Common mistakes:

1. (-0.5) Correctly identified a hypothesis class, but did not describe a situation in which that hypothesis class would do well and why.
2. (-0.25) Mention only low bayes error or only low dimensions
3. (-1) Saying that the SVM has an infinite VC dimension is false. SVM has a finite VC dimension of $\Theta(d)$. The SVM *with a gaussian kernel and appropriate σ value* can have an infinite VC dimension, but it is the gaussian kernel that causes the infinite VC dimension and not the SVM.

Problem 3. You work at an online advertisement company that uses vowpal wabbit to determine which ads to display to which users. You have trained a binary classification problem with hyperparameters

```
--bit_precision=28
--ngram=2
--passes=20
--learning_rate=0.01
```

Describe the likely effect on in-sample and generalization errors if you change the value of `--bit_precision` to `22` and `--ngram` to `3` (while keeping all other hyperparameters constant).

Solution: The VC dimension of the model is $2^{\text{bit precision}}$. Since the bit precision is decreasing, the VC dimension will also decrease, resulting in decreased generalization error. The ngram increasing has no effect on the VC dimension because the ngram feature map happens before feature hashing.

The training error will likely increase (although this is not guaranteed). There are two effects that will push the training error up. First, when the bit precision decreases, the number of hash collisions increases. Second, when ngrams increases, the total number of features (before hashing) increases, which will also increase the number of hash collisions. There is one effect that will push the training error down. Specifically, when ngrams increases, the number of features also increases, and it is possible that one of the new features is a particularly effective feature. Since more effects are pushing the training error up and one down, the training error is most likely to increase. It is possible, however, to construct datasets where the training error will decrease, and there is no way to know for sure without actually running the models and testing.

For the purposes of grading, your answer on the generalization error needed to agree with mine exactly, but your answer on the training error did not as long as what you wrote was reasonable.

Common mistakes:

1. (-0.5) If you stated that ngrams affects the VC dimension
2. (-0.5 or -0.25) A weak discussion of E_{in}, for example by not discussing hash collisions

Problem 4. You have a dataset with $N = 10^3$ training data points and $d = 10^2$ feature dimensions. You are training a logistic regression model using second order stochastic gradient descent, and you are not computationally limited so you run the optimization long enough for the optimization error to be negligible. You have evaluated the model on a separate validation set in order to determine that the PCA kernel with output dimension $d' = 10$ and elastic net regularization with $\lambda = 10^{-3}$ and $\alpha = 0.1$ provide good performance. Your boss wants you to simplify the training procedure by removing the PCA kernel feature map. Assuming you keep α constant, how does VC theory predict that you should change λ? That is, do you expect it to increase, decrease, or stay the same? Why?

Solution: Removing the PCA kernel will result in increasing the VC dimension, which corresponds to an increase in model complexity. In order to keep generalization error constant, we need a corresponding decrease in model complexity from regularization. Increasing lambda will add regularization and decrease model complexity, so the optimal λ will likely be larger for the new model.

Common mistakes:

1. (-0.5) Getting the right answer but not explaining why

Problem 5. You have successfully trained a `sklearn.neural_network.MLPClassifier` model and used a validation set to determine that the optimal hyperparameters are `hidden_layer_sizes=[1000]` and `alpha=0.0001`. (All other hyperparameters are the default values.) If you change the value of `hidden_layer_sizes` to `[10000]`, how would you expect to change the `alpha` hyperparameter in order to achieve the same generalization error? That is, do you expect it to increase, decrease, or stay the same? Why?

Solution: Increasing the size of the hidden layer increases the VC dimension and model complexity. Therefore, we will need to adjust `alpha` so that it adds regularization and decreases model complexity. The MLPClassifer uses the augmented loss regularization methodology, and `alpha` is equivalent to λ in the textbook. Increasing λ reduces model complexity, so we should expect to increase α.

Common mistakes:

1. (-0.5) If you assumed that `alpha` behaved like the soft order constraint regularization parameter C, but your reasoning was otherwise correct.
2. (-0.25) Not enough detail
3. (no penalty) No discussion about the difference between the soft order constraint and augmented error regularization methods.

Problem 6. You are training a boosted decision stump model on a dataset with $N = 10^6$ and $d = 10^6$, and have found that the optimal number of decision stumps in the model is $T = 10^3$. If you increase the amount of the data to $N = 10^9$ and the number of features to $d = 10^9$, but keep the number of base models constant at $T = 10^3$, does VC theory predict that the generalization error would increase, decrease, or stay the same?

To justify your answer, you should compute a tight upper bound on the VC dimension of the boosted decision stump hypothesis class and apply the fundamental theorem of statistical learning.

Solution: Boosted models have a VC dimension of $O(Td_{\text{VC}}(B)\log(Td_{\text{VC}}(B)))$ where $d_{\text{VC}}(B)$ is the VC dimension of the base classifier. Decision stumps have a VC dimension of $O(\log d)$. Substituting, we have that our model's VC dimension is

$$d_{\text{VC}} = O(T\log(d)\log(T\log(d)))). \tag{2}$$

VC theory bounds the generalization error with the formula

$$|E_{\text{in}} - E_{\text{out}}| = O\left(\sqrt{\frac{d_{\text{VC}}\log N}{N}}\right). \tag{3}$$

When we increase the number of dimensions d and N at the same rate, the numerator will increase by $\log d \log N$, but the denominator will increase at the much faster rate of N. Therefore, we expect the generalization error to decrease.

Common mistakes:

1. (-1) Incorrect formula (or no formula) for bounding the d_{VC} resulting in an incorrect discussion
2. (no penalty) Stating the correct formula and then stating $d_{\text{VC}} = O(T\log d)$ because the log factor "doesn't matter". The intuition is correct, but the formalism is incorrect. There is another version of asymptotic notation $\tilde{O}$ that ignores these log factors, and it is correct to say that $d_{\text{VC}} = \tilde{O}(T\log d)$.

Problem 7. In this problem you will derive a closed form solution to the ridge regression problem, which is closely related to the OLS regression problem. Recall that OLS uses the linear hypothesis class

$$\mathcal{H} = \left\{ \mathbf{x} \mapsto \mathbf{w}^T \mathbf{x} : \mathbf{w} \in \mathbb{R}^d \right\} \tag{4}$$

and the squared loss

$$\ell(\hat{y}, y) = (\hat{y} - y)^2 \tag{5}$$

so that the in-sample error is defined as

$$E_{\text{in}}(h) = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}_i^T \mathbf{w} - y_i)^2 = \|X\mathbf{w} - \mathbf{y}\|_2^2, \tag{6}$$

where X is a $N \times d$ matrix with ith row equal to $\mathbf{x}_i$ and $\mathbf{y}$ is the d dimensional vector with ith position equal to y_i. Finally, we computed the parameters for the OLS model by solving the equation

$$\hat{\mathbf{w}}^{\text{OLS}} = \operatorname*{arg\,min}_{\mathbf{w} \in \mathbb{R}^d} \|X\mathbf{w} - \mathbf{y}\|_2^2. \tag{7}$$

The ridge regression model modifies Equation (7) above by adding L2 regularization to get

$$\hat{\mathbf{w}}^{\text{ridge}} = \operatorname*{arg\,min}_{\mathbf{w} \in \mathbb{R}^d} \|X\mathbf{w} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{w}\|_2^2. \tag{8}$$

Your task is to derive a closed form solution for $\mathbf{w}^{\text{ridge}}$.

Solution: To solve for $\hat{\mathbf{w}}^{\text{ridge}}$ we take the derivative of the expression inside the arg min, set it equal to zero, and solve for $\mathbf{w}$:

$$\begin{aligned}
0 &= \frac{d}{d\mathbf{w}} \left(\|X\mathbf{w} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{w}\|_2^2 \right) && (9) \\
&= 2X^T(X\mathbf{w} - \mathbf{y}) + 2\lambda \mathbf{w} && (10) \\
&= 2X^T X\mathbf{w} - 2X^T \mathbf{y} + 2\lambda \mathbf{w} && (11) \\
&= X^T X\mathbf{w} - X^T \mathbf{y} + \lambda \mathbf{w} && (12) \\
X^T \mathbf{y} &= X^T X \mathbf{w} + \lambda \mathbf{w} && (13) \\
&= (X^T X + \lambda I)\mathbf{w} && (14) \\
(X^T X + \lambda I)^{-1} X^T \mathbf{y} &= \mathbf{w} && (15)
\end{aligned}$$

Common mistakes:

1. (-0.5) Incorrect use of left/right matrix multiplication or ordinary scalar division