Practice Questions for the ECE 661 Spring 2025 Midterm Exam

Introduction to Machine Learning for Engineers
Prof. Gauri Joshi and Prof. Carlee Joe-Wong

The number of practice questions included in this document may not be representative of the length of the exam. These are solely intended for you to gain more experience in answering questions similar to what you will see on the exam.

1 True or False

Problem 1: The computational complexity of KNN scales linearly with the dimension of the data D but quadratically with the number of samples N

- ◯ True
- ◯ False

Solution: False, the computational complexity of KNN is $O(ND)$, so it scales linearly with both

Problem 2: Both KNN and Logistic Regression (without applying non-linear transformations to the input features) will result in a linear decision boundary

- ◯ True
- ◯ False

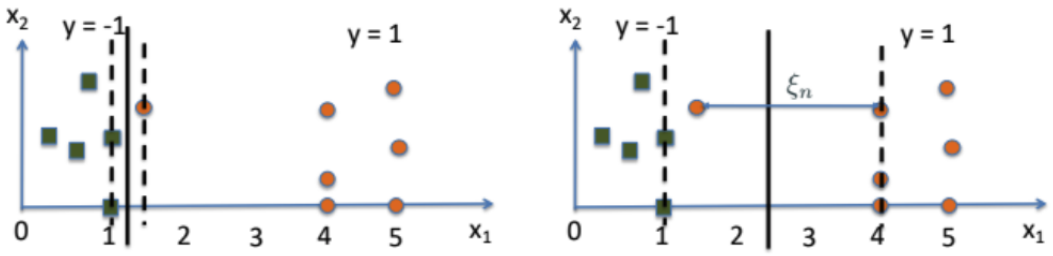
Solution: False, KNN does not necessarily result in a linear decision boundary.

Problem 3: In multi-class classification, taking the One-versus-One approach results in high variance, whereas taking the One-versus-All approach results in high bias.

- ◯ True
- ◯ False

Solution: True

Problem 4: Consider the following plot. The graph on the left demonstrates the classifier that would be learned by using soft-margin SVM, while the graph on the right demonstrates the classifier that would be learned using hard-margin SVM.



- ◯ True
- ◯ False

Solution: False, the graph on the left would be hard-margin SVM while the graph on the right would be soft-margin SVM. Soft-margin will allow for the misclassification of outliers

Problem 5: In multinomial logistic regression, our goal is to classify points $(\boldsymbol{x}_n, y_n)$ for $n = 1, \ldots, N$, with each point belonging to one of K classes. We learn K **independent** parameter vectors $\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_K$ such that the probability of the n-th point belonging to the k-th class is equal to

$$\Pr(y = k) = \frac{e^{\boldsymbol{w}_k^T \boldsymbol{x}_n}}{\sum_{j=1}^{K} e^{\boldsymbol{w}_j^T \boldsymbol{x}_n}}.$$

- ◯ True
- ◯ False

Solution: False. The K parameter vectors are not independent since all the probabilities need to sum up to 1. Therefore we only have $K - 1$ parameter vectors to learn.

Problem 6: By using kernel functions to implicitly map input data into a higher-dimensional space, linear algorithms (e.g., linear SVMs) can effectively learn non-linear decision boundaries in the original input space.

- ◯ True
- ◯ False

Solution: True.

2 Multiple Choice

Problem 7: In Naïve Bayes with Laplacian smoothing, if we increase the parameter α (the number of times we add the count of each unique item's occurrences for each label), the bias and the variance of the estimator change as follows:

- ◯ bias increases, variance decreases
- ◯ bias decreases, variance increases
- ◯ bias decreases, variance decreases
- ◯ bias increases, variance increases

Solution: A. bias increases, variance decreases

Problem 8: Answer the following question about the sigmoid function defined as $\sigma(x) = \frac{1}{1+e^{-x}}$

a. Which of the following statements is true -

- ◯ $\sigma(x) = \sigma(-x)$
- ◯ $\sigma(x) = 1 - \sigma(x)$
- ◯ $\sigma(x) = 1 - \sigma(-x)$
- ◯ $\sigma(x) = 1 + \sigma(-x)$

b. Which of the following statements is true -

- ◯ $\frac{d\sigma(x)}{dx} = \sigma(-x)$
- ◯ $\frac{d\sigma(x)}{dx} = (\sigma(x))^2$
- ◯ $\frac{d\sigma(x)}{dx} = (\sigma(-x))^2$
- ◯ $\frac{d\sigma(x)}{dx} = \sigma(x) \cdot \sigma(-x)$

Solution: C, D

Problem 9: Figure 1 shows the plots of loss vs. number of iterations for Batch Gradient Descent (GD), Stochastic Gradient Descent (SGD) and Mini-Batch Stochastic Gradient Descent (mini-batch SGD), which are run on a linear regression dataset with n data points to obtain the optimal parameters. Which of the following statements correctly identifies the plots and gives a correct comparison between the methods?

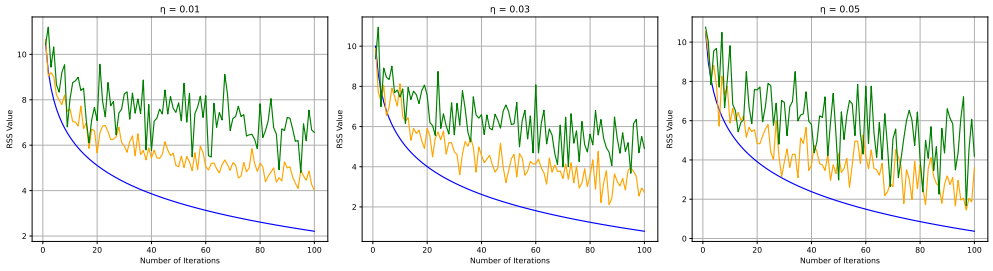


Figure 1: GD, SGD and Mini-batch SGD plots for different η (learning rate)

- ◯ blue: GD, yellow: mini-batch SGD, green: SGD.
  Running K iterations of SGD returns the same parameter vector as 1 iteration of GD.

◯ blue: GD, yellow: SGD, green: mini-batch SGD.
GD requires more computation per iteration than SGD and mini-batch SGD.

◯ blue: GD, yellow: mini-batch SGD, green: SGD.
Mini-batch SGD offers a compromise between GD and SGD as it has faster convergence than SGD but more computation and storage requirements than SGD.

◯ blue: mini-batch SGD, yellow: SGD, green: GD.
SGD requires less storage than mini-batch SGD.

Solution: C.
A is false since n iterations of SGD is not equivalent to 1 iteration of GD due to randomness in selection of data points in each iteration in SGD. B is false since yellow is mini-batch SGD and green is SGD. C is true since the plots are correctly identified and the tradeoff explained for mini-batch SGD is correct. D is false since plots are wrongly identified.

Problem 10: Consider a Naive Bayes classifier f and a ham/spam dataset $\mathcal{D}$ consisting of both spam and ham documents, where each document is a sequence of words. We will use the bag-of-words method to calculate the required parameters for the Naive Bayes Classifier and we will not use any smoothing. Which of the below changes to the dataset would also change these Naive Bayes parameters?

◯ Duplicating each document in the dataset exactly one time.

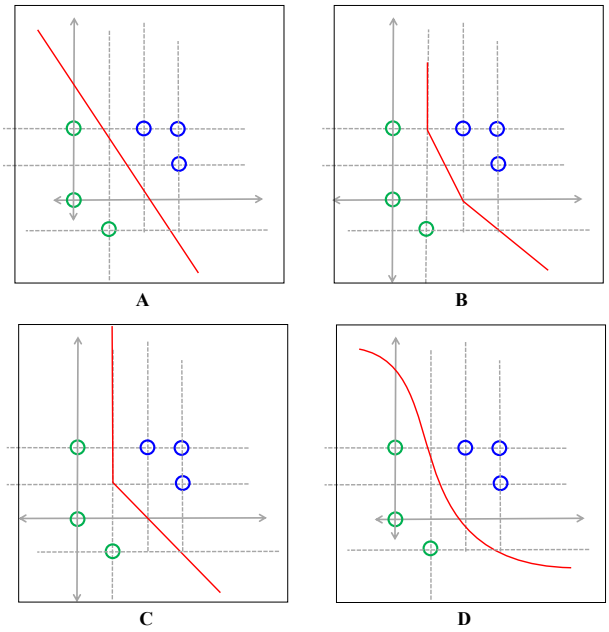
◯ Changing the order of the words in each document.

◯ Adding the word 'start' to the beginning of every document when 'start' already existed in the vocabulary.

◯ Duplicating every word in each document exactly two times.

Solution: C

Problem 11: Which of the following is the 1-nearest neighbors decision boundary for the dataset that is shown in the figure? We use the Euclidean distance metric to measure the distance between any two points.



◯ A

◯ B

◯ C

◯ D

Solution: C

Problem 12: Which of the following correctly compares the concept of support vectors in hard-margin vs. soft-margin SVM?

◯ In a hard-margin SVM, only points on the margin can be support vectors; in a soft-margin SVM, any point (inside or beyond the margin) can also become a support vector.

◯ In both hard-margin and soft-margin SVMs, every training point is a support vector.

◯ In a hard-margin SVM, points on the correct side but more than one margin-width away from the decision boundary are also considered support vectors.

◯ In a soft-margin SVM, no points that are misclassified can be support vectors.

Solution: A

Problem 13: Consider the Maximum-likelihood estimator (MLE) and the Maximum a posteriori (MAP) estimator in an arbitrary parameter estimation problem. Select the correct option from the following.

◯ It is always possible to find a closed-form expression for the MLE.

◯ If the MAP estimate is unique, then the MLE is unique.

◯ MLE is a special case of MAP estimation assuming a uniform prior distribution.

◯ MLE maximizes the probability of a set of parameters given the observed data, while MAP maximizes the joint probability of observed data conditioned on a set of parameters.

Solution: C

Problem 14: Consider a multi-class classification problem with $K = 5$ classes. How many binary classifiers you need to train respectively using either "one-versus-one" or "one-versus-rest" approaches?

◯ 20 and 5

◯ 10 and 5

◯ 5 and 10

◯ 10 and 10

Solution: B

3 Descriptive

Problem 15: For a course assignment on spam classification using logistic regression, Alice and Bob are given a dataset of emails containing two categories "spam" and "ham". Alice argues that since the main motive is to figure out "spam" email, one should label the spam emails as $y = 1$ and the "ham" as $y = 0$. Bob argues that class labels are not important in logistic regression and labelled the spam emails as $y = 0$ and the "ham" as $y = 1$.
Recall that the cross-entropy loss function is:

$$\mathcal{E}(\boldsymbol{w}) = -\sum_{n=1}^{N}\{y_n \log \sigma(\boldsymbol{w}^\top \boldsymbol{x}_n) + (1-y_n)\log[1-\sigma(\boldsymbol{w}^\top \boldsymbol{x}_n)]\} \tag{1}$$

where $\mathbf{w}$ denote the weight vector (including the bias) and $\boldsymbol{x}$ is the augmented feature vector (including a 1 corresponding to the bias). The sigmoid function is $\sigma(a) = 1/(1+e^{-a})$. The loss function can be simplified to the following expression:

$$\mathcal{E}(\boldsymbol{w}) = \sum_{n=1}^{N}\{(1-y_n)\boldsymbol{w}^\top \boldsymbol{x}_n + \log(1+e^{-\boldsymbol{w}^\top \boldsymbol{x}_n}).\} \tag{2}$$

a. Using the simplified version of the cross-entropy function shown in (2), express Bob's cross-entropy loss function $\mathcal{E}_{Bob}(\boldsymbol{w})$ in terms of Alice's cross-entropy loss fucntion $\mathcal{E}_{Alice}(\boldsymbol{w})$ and show that

$$\mathcal{E}_{Bob}(\boldsymbol{w}) = \mathcal{E}_{Alice}(-\boldsymbol{w}) \tag{3}$$

b. Suppose that Alice and Bob start with the same initialization $\boldsymbol{w}_{Alice}^{(0)} = \boldsymbol{w}_{Bob}^{(0)} = \mathbf{0}$, and run t iterations of gradient descent on their own loss functions using the same fixed learning rate η. Prove that for any $t \in \mathbb{N}$,

$$\boldsymbol{w}_{Alice}^{(t)} + \boldsymbol{w}_{Bob}^{(t)} = \mathbf{0}$$

Hint: Use your answer from part (a) and prove by induction.

c. Compare the equations of the decision boundaries learnt by Alice and Bob and comment on their similarity and/or differences.

Solution:

a. The cross-entropy function can be simplified as follows:

Let $\mathcal{E}_{Alice}(\mathbf{w})$ be the loss function calculated by Alice. It is equal to the simplified cross-entropy error function:

$$\mathcal{E}_{Alice}(\boldsymbol{w}) = \sum_{n=1}^{N}\{(1-y_n)(\boldsymbol{w}^\top \boldsymbol{x}_n) + \log\left(1+e^{-\boldsymbol{w}^\top \boldsymbol{x}_n}\right)\} \tag{4}$$

Bob's loss function will replace y_n by $1-y_n$ everywhere to get

$$\mathcal{E}_{Bob}(\boldsymbol{w}) = \sum_{n=1}^{N}\{y_n(\boldsymbol{w}^\top \boldsymbol{x}_n) + \log\left(1+e^{-\boldsymbol{w}^\top \boldsymbol{x}_n}\right)\} \tag{5}$$

$$= \sum_{n=1}^{N}\{y_n(\boldsymbol{w}^\top \boldsymbol{x}_n) + \log\left(1+e^{\boldsymbol{w}^\top \boldsymbol{x}_n}\right) - \boldsymbol{w}^\top \boldsymbol{x}_n\} \tag{6}$$

$$= \sum_{i=1}^{n}\left((1-y_i)\left(-\boldsymbol{w}^\top \boldsymbol{x}_i\right) + \log\left(1+e^{\boldsymbol{w}^\top \boldsymbol{x}_n}\right)\right) \tag{7}$$

$$= \mathcal{E}_{Alice}(-\boldsymbol{w}) \tag{8}$$

Hence,

$$\nabla_{\mathbf{w}} \mathcal{E}_{Bob}(\boldsymbol{w}) = -\nabla_{\boldsymbol{w}} \mathcal{E}_{Alice}(-\boldsymbol{w}) \tag{9}$$

b. **Proof by induction**: Clearly the results holds for $k = 0$ as all the weights are zero. Assume it holds for the k-th iteration, i.e., $\boldsymbol{w}_{Alice}^{(k)} + \boldsymbol{w}_{Bob}^{(k)} = \mathbf{0}$. The next weight vector after a gradient descent step is given by

$$\boldsymbol{w}_{Alice}^{(k+1)} = \boldsymbol{w}_{Alice}^{(k)} - \eta \nabla_{\mathbf{w}} \mathcal{E}_{Alice}(\boldsymbol{w}_{Alice}^{(k)}). \tag{10}$$

Similarly,

$$\begin{aligned}
\boldsymbol{w}_{Bob}^{(k+1)} &= \boldsymbol{w}_{Bob}^{(k)} - \eta \nabla_{\mathbf{w}} \mathcal{E}_{Bob}(\boldsymbol{w}_{Bob}^{(k)}) && (11)\\
&= \boldsymbol{w}_{Bob}^{(k)} + \eta \nabla_{\boldsymbol{w}} \mathcal{E}_{Alice}(-\boldsymbol{w}_{Bob}^{(k)}) && (12)\\
&= -\boldsymbol{w}_{Alice}^{(k)} + \eta \nabla_{\mathbf{w}} \mathcal{E}_{Alice}(\boldsymbol{w}_{Alice}^{(k)}) && (13)\\
&= -\boldsymbol{w}_{Alice}^{(k+1)} && (14)
\end{aligned}$$

This completes the proof.

c. The Decision boundary is given by the equation $\mathbf{w}^\top x = 0$ and it is invariant of sign. Hence both Alice and Bob will end up with the same decison boundary.

Problem 16: Suppose you are given a training dataset $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)\}$ where each $(\boldsymbol{x}_i, y_i)$ pair consists of a d-dimensional feature vector $\boldsymbol{x}_i$ and label y_i, $i = 1, 2, \ldots, n$.

a. Suppose that $d = 2$. For each training data point i, the second feature's value is equal to the label y_i: thus, $\boldsymbol{x}_i = \begin{bmatrix} \boldsymbol{x}_{i,1} \\ y_i \end{bmatrix}$. You now solve for the linear regression coefficients $\boldsymbol{w}^*$ that minimize the squared-error $\sum_{i=1}^{n} \left(y_i - \boldsymbol{w}^T \boldsymbol{x}_i\right)^2$.

Is the optimal $\boldsymbol{w}^*$ unique? If so, find a closed-form expression for $\boldsymbol{w}^*$. If not, explain why not and suggest a way to modify the linear regression formulation so that it has a unique solution $\boldsymbol{w}^*$.

Solution: By inspection, the optimal $\boldsymbol{w}^* = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$. Note that we can also use the linear regression formula $\boldsymbol{w}^* = \left(\begin{bmatrix} \boldsymbol{x}^T \\ \boldsymbol{y}^T \end{bmatrix} \begin{bmatrix} \boldsymbol{x} & \boldsymbol{y} \end{bmatrix} \right)^{-1} \begin{bmatrix} \boldsymbol{x}^T \\ \boldsymbol{y}^T \end{bmatrix} \boldsymbol{y} = \begin{bmatrix} \boldsymbol{x}^T\boldsymbol{x} & \boldsymbol{x}^T\boldsymbol{y} \\ \boldsymbol{y}^T\boldsymbol{x} & \boldsymbol{y}^T\boldsymbol{y} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{x}^T\boldsymbol{y} \\ \boldsymbol{y}^T\boldsymbol{y} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$. This solution is unique.

b. Now suppose you add another new feature to each input vector from part (b), so that $d = 3$. This new feature is equal to the sum of the first input feature and the label, so that each feature vector $\boldsymbol{x}_i = \begin{bmatrix} \boldsymbol{x}_{i,1} \\ y_i \\ \boldsymbol{x}_{i,1} + y_i \end{bmatrix}$.

Is the optimal $\boldsymbol{w}^*$ that minimizes $\sum_{i=1}^{n} \left(y_i - \boldsymbol{w}^T \boldsymbol{x}_i\right)^2$ unique? If so, find a closed-form expression for $\boldsymbol{w}^*$. If not, explain why not and suggest a way to modify the linear regression formulation so that it has a unique solution $\boldsymbol{w}^*$.

Solution: The features are no longer linear independent, so $\boldsymbol{w}^*$ is no longer unique. We can use ridge regression to make the solution unique.

c. Now suppose you apply ridge regression to the training dataset from part (c), i.e., you solve for $\boldsymbol{w}^*_{ridge}$ that minimizes $\sum_{i=1}^{n} \left(y_i - \boldsymbol{w}^T \boldsymbol{x}_i\right)^2 + \lambda \|\boldsymbol{w}\|_2^2$ for some $\lambda > 0$. Is the optimal $\boldsymbol{w}^*_{ridge}$ unique? Explain why or why not in one or two sentences.

Solution: Yes, $\boldsymbol{w}^*_{ridge}$ is unique. The ridge regression solution is always unique, since it is equal to $\left(2\boldsymbol{X}^T\boldsymbol{X} + \lambda I\right)^{-1} \boldsymbol{X}^T\boldsymbol{y}$, which is a unique solution.