USC CSCI 467 Intro to Machine Learning	**Midterm Exam** March 9, 2023, 2:00-3:20pm	Spring 2023 Instructor: Robin Jia

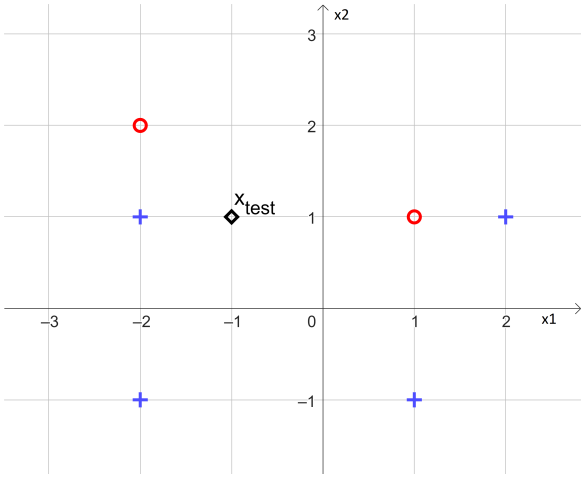
Name: _____

USC e-mail: _____@usc.edu

Answer the questions in the spaces provided. **If you write solutions on the back of the pages, indicate this on the front of the pages so we know to look there, but please try to avoid this if possible.** You may use the backs of pages for scratch work. This exam has 4 questions, for a total of 100 points.

Question 1: Classification Options (23 points)

Emily has the following binary classification dataset:

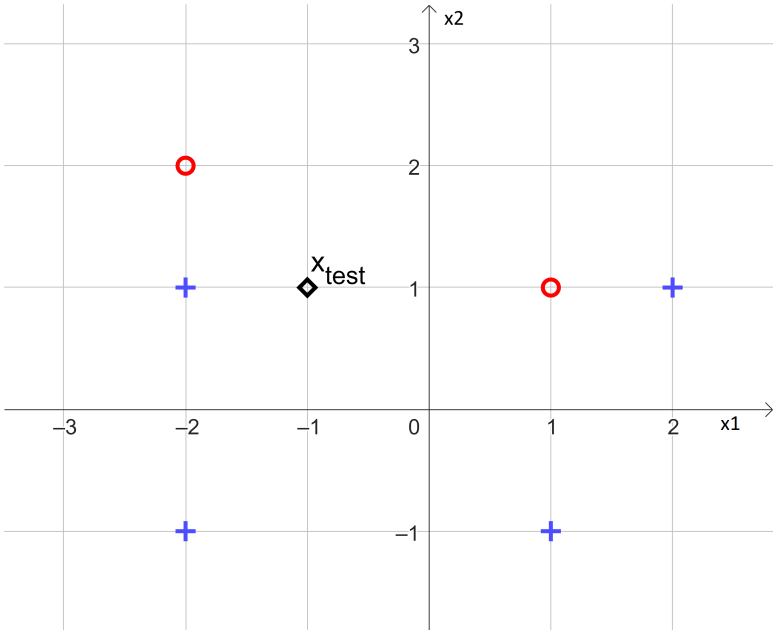


Pluses denote positive training examples ($y = 1$) and circles denote negative training examples ($y = -1$). There is also a test example labeled x_{test}. Emily is trying different classifiers on this data.

(a) (4 points) Emily first tries k-Nearest Neighbors with $k = 3$ and the Euclidean norm as the distance metric. She then makes a prediction on the point $x_{test} = (-1, 1)$. What prediction will the model make on x_{test}, and why?

Solution: The 3 nearest neighbors are the points at $(-2, 1)$, $(-2, 2)$, and $(1, 1)$. Two out of these three are negative examples, so x_{test} will be classified as negative.

(b) (3 points) Next, Emily tries logistic regression, using the given features and a bias term. In the copy of the figure below, draw a decision boundary that Emily is likely to learn if she uses logistic regression. Explain your reasoning. (You don't need to compute the exact parameters.)

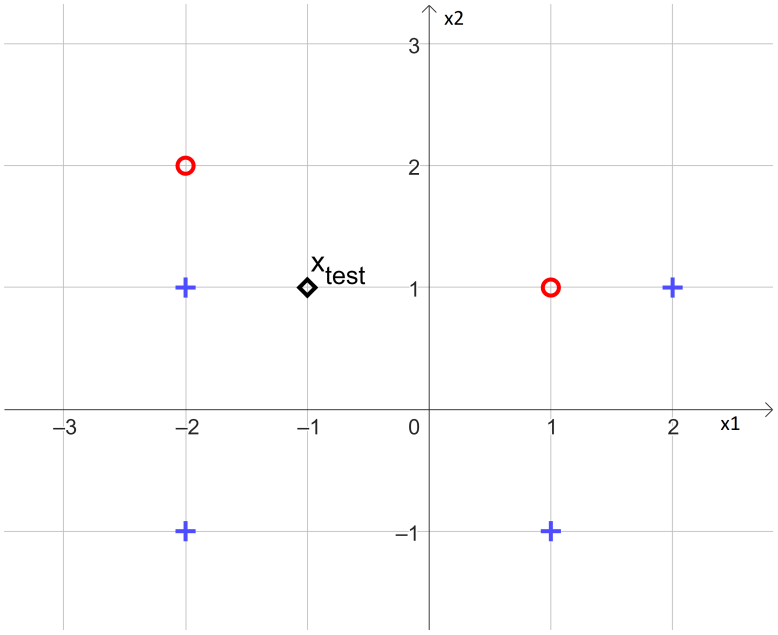


Solution: The decision boundary will be a straight line. It's impossible to have 0 errors, but there are multiple ways to have 1 error. Any of those is acceptable as a solution.

(c) (6 points) Next, Emily tries adding more features. She defines the following feature function:

$$\phi(x) = \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ x_1^2 \end{pmatrix}$$

In the copy of the figure below, draw a decision boundary that Emily is likely to learn if she runs logistic regression using this feature function. Explain your reasoning. (You don't need to compute the exact parameters.)

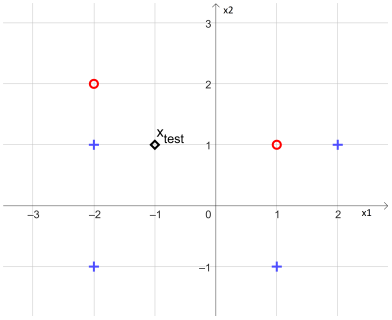


Solution: The general form of the decision boundary is

$$w_1 + w_2 x_1 + w_3 x_2 + w_4 x_1^2 = 0$$
$$x_2 = \frac{-1}{w_3} \cdot \left(w_4 x_1^2 + w_2 x_2 + w_1 \right).$$

This is the equation for a parabola. The boundary will be a parabola curved upwards such that the two negative examples are above the curve, and the other points are below the curve.

(d) (10 points) Finally, Emily decides to try Naive Bayes on this dataset (copied below).



She isn't sure how to run Naive Bayes with real-valued features,[1] so she decides to discretize her inputs by first applying the following feature function:

$$\phi(x) = \begin{pmatrix} \mathbb{I}[x_1 > 0] \\ \mathbb{I}[x_2 > 0] \end{pmatrix}.$$

Recall that $\mathbb{I}[expr]$ takes in a boolean expression and returns 1 if it is true, and 0 if it is False. After applying this feature function, Emily runs Naive Bayes on the dataset

[1] Actually it is possible to use Naive Bayes with real-valued features, but that is beyond the scope of this question.

with Laplace Smoothing $\lambda = 1$. For the test point $x_{test} = (-1, 1)$, compute the model's probability that x_{test} has the label $y = 1$. Show your work.

Solution: First, note that $\phi(x_{test}) = [0, 1]$. We want to compute

$$P(Y = 1 \mid X = \phi(x_{test}))$$

Since we are using Naive Bayes, we make the Naive Bayes assumption, namely that the first and second feature are independent conditioned on the label. Thus we have:

$$\begin{aligned} P(y = 1) &= \frac{2}{3} \\ P(\phi(X)_1 = 0 \mid Y = 1) &= \frac{2+1}{4+2} = \frac{1}{2} \\ P(\phi(X)_1 = 0 \mid Y = 0) &= \frac{1+1}{2+2} = \frac{1}{2} \\ P(\phi(X)_2 = 1 \mid Y = 1) &= \frac{2+1}{4+2} = \frac{1}{2} \\ P(\phi(X)_2 = 1 \mid Y = 0) &= \frac{2+1}{2+2} = \frac{3}{4} \end{aligned}$$

Thus, putting this all together, we have:

$$\begin{aligned} P(X = \phi(x_{test}), Y = 1) &= \frac{2}{3} \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{6} \\ P(X = \phi(x_{test}), Y = 0) &= \frac{1}{3} \cdot \frac{1}{2} \cdot \frac{3}{4} = \frac{1}{8} \end{aligned}$$

So, when we normalize things out, we get

$$P(Y = 1 \mid X = \phi(x_{test}) = \frac{1/6}{1/6 + 1/8} = \boxed{\frac{4}{7}}.$$

Question 2: Analyzing hinge loss (28 points)

In class, we briefly discussed hinge loss, which is the loss function employed by support vector machines (SVMs). In this problem, we will dive deeper into hinge loss.

For a binary classification dataset $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$, the SVM loss is defined as:

$$L(w) = \frac{1}{n} \left(\sum_{i=1}^{n} [1 - y^{(i)} w^\top x^{(i)}]_+ \right) + \lambda \|w\|_2^2$$

where each $x^{(i)}$ is a vector $\in \mathbb{R}^d$ representing an input, $y^{(i)}$ is either $+1$ or -1, $w \in \mathbb{R}^d$ is the weight vector, and $[z]_+$ denotes the "positive part" of a number z, defined as:

$$[z]_+ = \begin{cases} z & \text{if } z > 0 \\ 0 & \text{if } z \leq 0 \end{cases}$$

(a) (3 points) $L(w)$ depends on the quantity $y^{(i)} w^\top x^{(i)}$. What is the name for this quantity? Is it better for it to be large or small? Explain your reasoning.

Solution: This is called the margin. A positive margin corresponds to classifying the example correctly, so larger is better.

(b) (10 points) Compute the gradient of this loss function, $\nabla_w L(w)$. Do not worry about what happens at the non-differentiable part of the "positive part" function. (Hint: The indicator function $\mathbb{I}[expr]$ may be useful in your answer)

Solution: We apply chain rule, noting that the derivative of the "positive part" function is 1 if the argument is positive and 0 if the argument is non-negative, which is succinctly described by the indicator function.

$$\nabla_w L(w) = \frac{1}{n}\left(\sum_{i=1}^{n} \mathbb{I}[1 - y^{(i)} w^\top x^{(i)} > 0] \cdot (-y^{(i)} x^{(i)})\right) + 2\lambda \cdot w.$$

(c) (4 points) Write the gradient descent update rule to update w, using a learning rate η.

Solution:

$$w \leftarrow w - \eta \cdot \left(\frac{1}{n}\left(\sum_{i=1}^{n} \mathbb{I}[1 - y^{(i)} w^\top x^{(i)} > 0] \cdot (-y^{(i)} x^{(i)})\right) + 2\lambda \cdot w\right).$$

We also accepted answers that just wrote $\nabla_w L(w)$ instead of substituting in the expression from the previous part. You got this part correct even if the previous part was wrong, as long as the expression here correctly substitutes in what you computed as the gradient.

(d) (5 points) Suppose you keep making λ larger and larger. What will happen to the Euclidean norm of the optimal value of w (that is, $\|w\|$) as you do this? Explain your reasoning.

Solution: Increasing λ corresponds to increasing the penalty for w having large norm. As we make λ larger and larger, $\|w\|$ will decrease towards 0.

(e) Recall from Homework 1 that we studied what happens to logistic regression when you have a linearly separable dataset. Let's do some similar analysis for support vector machines. Suppose you have a weight vector w such that $w^\top x^{(i)} = 0.5$ whenever $y^{(i)} = 1$ and $w^\top x^{(i)} = -0.5$ whenever $y^{(i)} = -1$. Let's also set $\lambda = 0$.

i. (3 points) What is the loss $L(w)$ of the weight vector above? Explain your reasoning. (Hint: Your answer will be independent of n, the number of training examples)

Solution: The hinge loss is 0.5 on every example, and the objective is an average over the examples, so it is just 0.5.

ii. (3 points) Suppose we multiply w by 10. What will be the new value of $L(w)$? Explain your reasoning.

Solution: Now the margin is 5 for every example, so the loss will just be 0.

Question 3: Classifying Cats (21 points)

James likes big cats. He has a collection of color images, each of which has been accurately labeled as being a leopard, jaguar, cheetah, or lion. He wishes to design a four-way classifier that can classify new images as one of these four species. Your task is to assist James in creating a 4-way classifier for this dataset.

(a) (3 points) It's best to try simple solutions first, so you tell James to try a linear model. Consider (1) linear regression, (2) logistic regression, and (3) softmax regression. Which one would you recommend? Why is it more suitable than the other two?

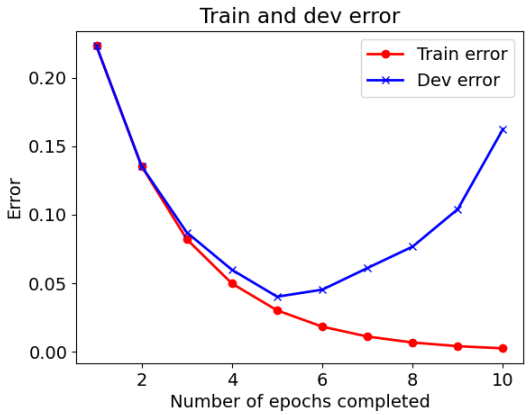
Solution: Softmax regression is appropriate because it is the only algorithm out of the three that is designed for multi-class classification.

(b) (4 points) James is not satisfied with the accuracy of the model, so you suggest training a convolutional neural network with stochastic gradient descent. The CNN uses convolutional layers, ReLU, max pooling, and a fully-connected layer at the end. You advise James to try different values of hyperparameters, such as the number of training epochs, to see which ones lead to the best development set accuracy. Name four other hyperparameters that James should try to tune.

Solution: Possible answers include:

- Kernel width
- Kernel number of output channels
- Max pooling width
- Number of layers
- Learning rate
- Batch size
- Amount of dropout

(c) Below is a plot of the training and development error (i.e., $1 - accuracy$) James observes at each epoch during one training run.



i. (2 points) Which epoch's model should James select as the best model? Explain your reasoning.

Solution: Epoch 5, since it has the lowest dev error.

ii. (2 points) Do you see evidence of overfitting on this plot? If so, where?

Solution: Yes, the right side of the plot shows the training error getting smaller while the dev error gets much larger. This is a sign of overfitting.

(d) (4 points) James decides to increase the resolution of his images from 100×100 to 200×200. (There are also 3 color channels in both cases). He wants to know how this will change the number of parameters in the first layer of the network, which is a convolutional layer. By what factor will the number of parameters in this layer change? (e.g., will it have 10 times more parameters?) Assume everything besides the size of the input images remains the same. You may ignore bias terms.

Solution: The number of parameters will not change, since you just use the same size convolutional kernel and have the same number of input and output channels.

(e) (4 points) James has also heard that he could use a fully-connected layer as the first layer of his neural network, instead of a convolutional layer. He is curious about the same question: when he increases the resolution from 100×100 to 200×200, by what factor does the number of parameters in the first layer change? Again, assume everything besides the size of the input images remains the same and ignore bias terms.

Solution: A fully connected layer has numInputs $\times$ numOutputs total parameters. Increasing the resolution increases the number of inputs by a factor of 4, so the number of parameters also increases by a factor of 4.

(f) (2 points) You look at the dataset and notice that the dataset is highly imbalanced—80% of the images are leopards. Explain why accuracy may not be a good metric for evaluation, and suggest an alternative evaluation metric.

Solution: Accuracy is not good when there is label imbalance because a model can do very well by just outputting the majority class. F1 macro-averaged over all classes is a better choice. Another good answer is to balance the test set and report accuracy on a balanced test set, or to compute the accuracy on data from each class and average those 4 numbers.

Question 4: Short Answer (28 points)

In the following questions, circle the correct answer(s).

(a) (2 points) **True** or **False**: The function $f(x) = -|x|$ from $\mathbb{R} \to \mathbb{R}$ is a convex function.

Solution: False. If you draw a line connecting two points, it is below the function, not above.

(b) (2 points) **True** or **False**: Second-order optimization methods like Newton-Raphson are computationally difficult to apply to models that have a very large number of parameters.

Solution: True, since it is quadratic in the number of parameters.

(c) (2 points) **True** or **False**: For logistic regression, the amount of L2 regularization λ should be chosen to maximize accuracy on the test set.

Solution: False, it should be chosen on the dev set.

(d) (2 points) **True** or **False**: In a multi-layer perceptron, the function $f(z) = 2z$ is a reasonable choice for an activation function.

Solution: False, this is a linear function. Activation functions should be non-linear.

(e) (2 points) **True** or **False**: In a Transformer, information flows between different words in the feedforward layers.

Solution: False. Information flows between words in the multi-headed attention layers. The feedofrward layers process each word independently.

(f) (2 points) Which of the following is **not** true about kernels?

A. With kernels, you can learn a classifier that is equivalent to doing normal learning in a very high-dimensional feature space.

B. Kernels add regularization and prevent overfitting.

C. Kernels allow you to learn a non-linear function of your original features.

D. Kernels can encode the intuition that similar examples should have similar labels.

Solution: (B) Kernels do not prevent overfitting. The other statements are true.

(g) (4 points) Which of the following neural network layers/components have **no** learnable parameters? Circle all that apply.

A. Fully connected layer
B. ReLU activation
C. Convolutional layer
D. Max Pooling layer
E. Word vector layer
F. Dropout layer
G. Attention layer (the version used in RNN encoder-decoder models)
H. Multi-headed Attention layer

Solution: B, D, F, and G.

(h) You have an RNN model and use it to encode two different sentences: one with 10 words and one with 50 words.

i. (2 points) **True** or **False**: The number of parameters is larger when encoding the second sentence.

ii. (2 points) **True** or **False**: The number of hidden states is larger when encoding the second sentence.

iii. (2 points) **True** or **False**: The time it takes to run the RNN is larger when encoding the second sentence.

Solution: False, True, True.
RNNs have a fixed set of parameters, but they can be applied to sequences of any length. However, this requires more time and generates more hidden states (one per word).

(i) For each question below, circle all models that fit the description.

i. (3 points) When predicting on the test set, it is necessary to keep the training data stored in memory.

A. Linear Regression
B. Naive Bayes
C. k-Nearest Neighbors
D. Kernel Logistic Regression
E. Multi-layer Perceptron

Solution: C and D.
For k-NN and kernel logistic regression, the prediction on the test set depends on similarity of the test example to the different training examples.
For linear regression and MLP, you only need to fit parameters on the training data; then you no longer need the training data at test time.
For Naive Bayes, you just need to obtain counts of different features on the training data; after that, the training data is not needed.

ii. (3 points) Training and testing can be done in time that is linear in the size of the training dataset.

A. Linear Regression
B. Naive Bayes

C. k-Nearest Neighbors

D. Kernel Logistic Regression

E. Multi-layer Perceptron

Solution: A, B, C, and E.
The only one for which this is not true is kernel logistic regression, which requires quadratic time in the size of the dataset. Note that for an MLP, a training epoch takes linear time in the size of the dataset.