CS 491 Final Exam Name ________________________

Short Problems (20 points)

1. True/False: When training a neural network, you should check to see when the error is at a minimum on the test data, and stop training at that point. Explain. (5 points)

2. Give one similarity and one difference between feature selection and PCA. (5 points)

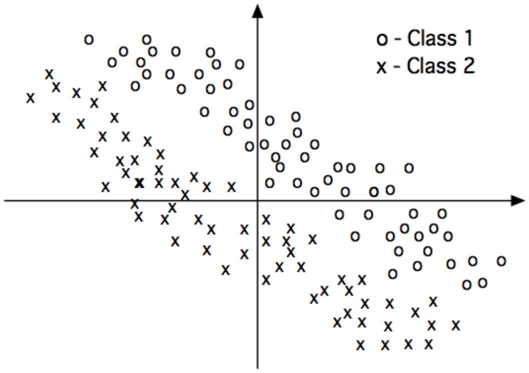
3. True/False: The maximum margin decision boundaries that support vector machines construct have the lowest generalization error among all linear classifiers. Explain. (5 points)

4. Which of the following statements are true? No need to explain. (5 points)
   (a) Training a k-nearest neighbors classifier takes more computational time than applying it.
   (b) The more training examples, the more accurate the prediction of a k-nearest neighbors classifier.
   (c) k-nearest neighbors cannot be used for regression (to predict a real-value).
   (d) A k-nearest neighbors is sensitive to outliers.
   (e) K=1 in K-NN always give 100% accuracy on the training data.

491

PCA (10 points)

5. Explain how you could combine PCA with a supervised method we have covered this semester to classify the following data. Clearly state the steps you would take to combine the two methods and identify the particular supervised method you would use.



491

Neural Networks (20 points)

6. Consider two types of Neural Network activations:

   - linear: $h = w \cdot x + b$
   - hard threshold: $h = 1$ if $w \cdot x + b \geq 0$, and $h = 0$ otherwise.

   Which of the following functions can be exactly represented by a neural network with one hidden layer which uses linear and/or hard threshold activation functions? For each case, justify your answer.

   (a) Polynomials of degree one (5 points)

   (b) Hinge loss: $h(x) = max(1 - x, 0)$ (5 points)

   (c) Polynomials of degree two (5 points)

   (d) Piece-wise constant functions (5 points)

491

Decision Boundaries (20 points)

7. For each of the following situations, **indicate whether the classifier will produce a linear decision boundary always, sometimes or never. Give a short explanation of your answer.** In every case, assume that the input has two features x_1 and x_2, which have continuous values ranging from -1 to 1. For example, the value of x_1 is any real number between -1 and 1. And suppose the label, y has two possible values, 1 or -1. In all cases, assume there is plenty of training data (at least 100 examples for each class).

   (a) We build a decision tree with two levels. (3 points)

   (b) We build a decision tree with one level. (3 points)

   (c) We train a Perceptron. (3 points)

   (d) We use K-nearest neighbor, with K = 1. (3 points)

   (e) We train an SVM using a quadratic kernel. (4 points)

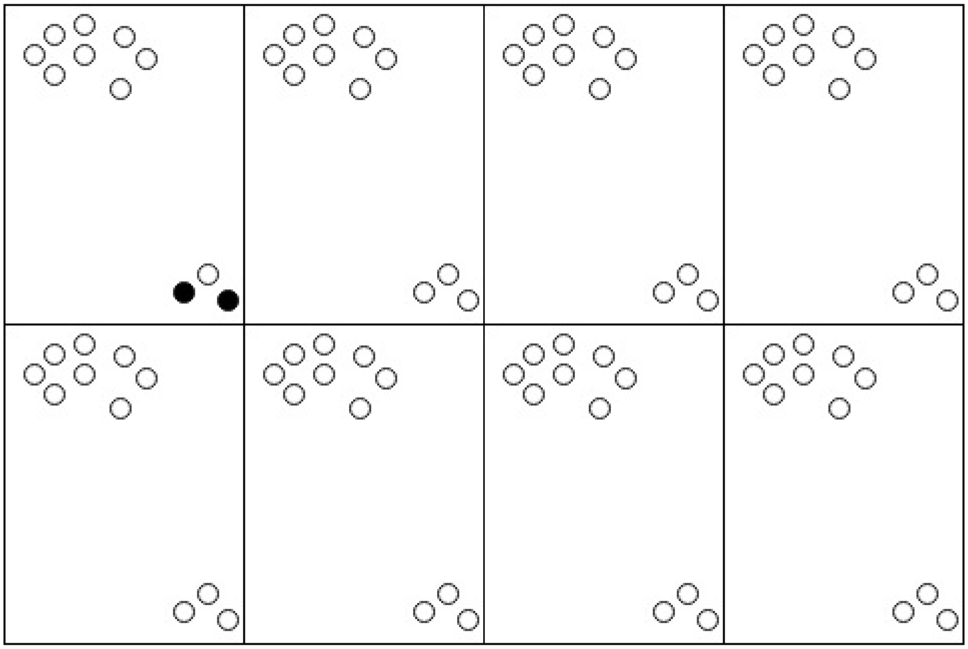
   (f) We use a neural network. The network has two input units, two hidden units and one output unit. We use ReLU in the hidden units (remember ReLU(x) = max(0,x)) and a sigmoid after the output unit. The input is classified as belonging to class 1 if the output is greater than $\frac{1}{2}$, and as class -1 if the output is less than $\frac{1}{2}$. (4 points)

491

K-Means (10 points)

8. Consider the following questions regarding K-Means.

   (a) Run K-Means manually for the following dataset. Circles are data points and the filled-in circles in the first panel are the initial cluster centers. Draw the cluster centers and the decision boundaries that define each cluster. Use a different panel for each iteration. Use as many panels as you need until convergence. (6 points)

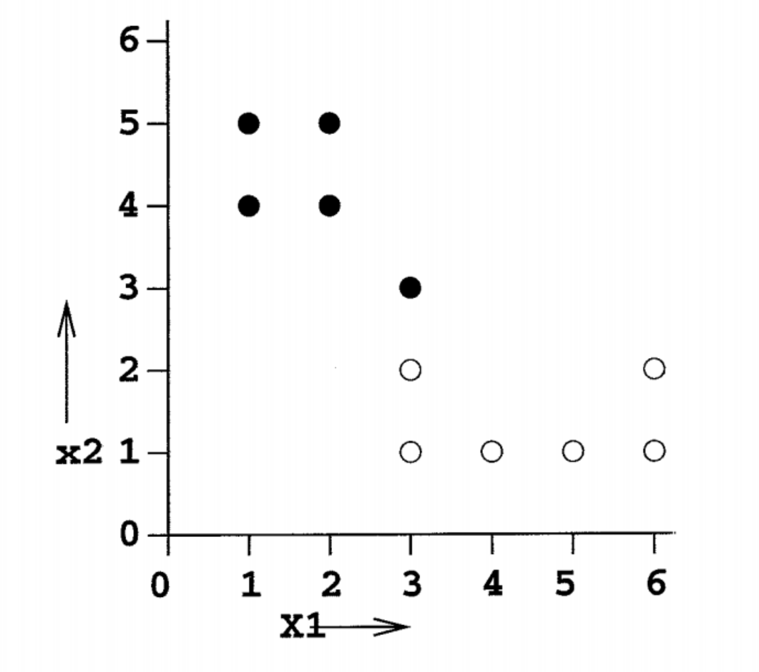


   (b) True/False: It is possible that K-Means (K=2) gives the same decision boundary as 1-NN for a binary classification problem. If true, give an example. False, briefly explain. (4 points)

491

SVM (10 points)

9. The following figure shows a dataset with two real-valued features (x_1 and x_2) and a binary output class. The positive points are shown as solid dots and the negative points are small circles.



(a) Suppose you are using a linear hard-margin SVM. Draw three lines on the above diagram, showing the decision boundary and the two sides of the margin. Circle the support vector(s). (3 points)

(b) Calculate the values of w and b learned for part (a). (7 points)

491

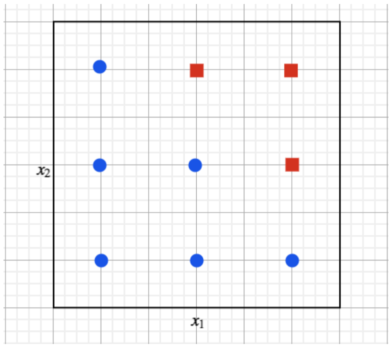
Ensembles (10 points)

10. Recall that Adaboost learns a classifier f using a weighted sum of weak learners f_t as follows:

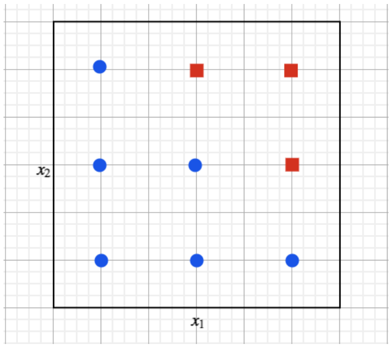
$$f(x) = sign(\sum_k \alpha_k f_k(x))$$

In this question, we will use decision trees as our weak learners, which classify a point as $\{1, -1\}$ based on a sequence of threshold splits on its features (x_1 and x_2). Red squares are negative points and blue circles are positive points. **In the questions below, be sure to mark which regions are classified positive/negative.** Assume that our weak learners are decision trees of maximum depth 2, which minimize the weighted training error. That is, the set of classifiers f_k are all depth-1 or depth-2 decision trees with minimum training set error.

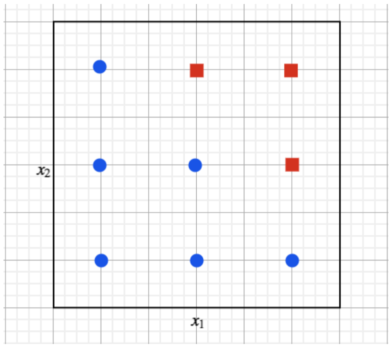
(a) Using the dataset below, draw the decision boundary learned by f_1 (the first decision tree). There are many possible answers for this.



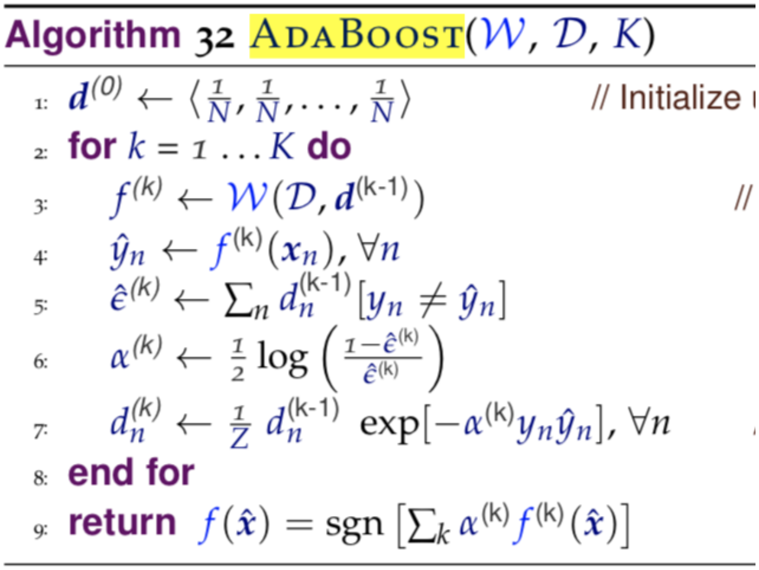
(b) On the dataset below, circle the point(s) with the highest weights on the second iteration, and draw the decision boundary learned by f_2 (the second decision tree).



(c) On the dataset below, draw the decision boundary $f = sign(\alpha_1 f_1 + \alpha_2 f_2)$. (Hint: you do not need to explicitly compute the α's).



491



Algorithm 32 AdaBoost($\mathcal{W}$, $\mathcal{D}$, K)

```
$\boldsymbol{d}^{(0)} \leftarrow \langle \frac{1}{N}, \frac{1}{N}, \ldots, \frac{1}{N} \rangle$          // Initialize
for $k = 1 \ldots K$ do
    $f^{(k)} \leftarrow \mathcal{W}(\mathcal{D}, \boldsymbol{d}^{(k-1)})$          //
    $\hat{y}_n \leftarrow f^{(k)}(\boldsymbol{x}_n), \forall n$
    $\hat{\epsilon}^{(k)} \leftarrow \sum_n d_n^{(k-1)} [y_n \neq \hat{y}_n]$
    $\alpha^{(k)} \leftarrow \frac{1}{2} \log \left( \frac{1 - \hat{\epsilon}^{(k)}}{\hat{\epsilon}^{(k)}} \right)$
    $d_n^{(k)} \leftarrow \frac{1}{Z} d_n^{(k-1)} \exp[-\alpha^{(k)} y_n \hat{y}_n], \forall n$
end for
return $f(\hat{\boldsymbol{x}}) = \mathrm{sgn} \left[ \sum_k \alpha^{(k)} f^{(k)}(\hat{\boldsymbol{x}}) \right]$
```