Georgia State University
Computer Science

CSC 4850 / 6850– Practice Final
Machine Learning

Instructor: Prof. Dong Hye Ye

2023/4/20

Name: _____
Student Number: _____

This exam contains 7 pages (including this cover page) and 4 questions. Total of points is 100. Good luck!

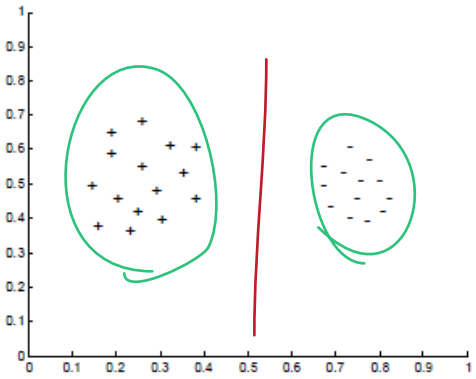
Distribution of Marks

Question	Points	Score
1	20	
2	20	
3	30	
4	30	
Total:	100	

1. (20 points) **Gaussian Naive Bayes and Logistic Regression**

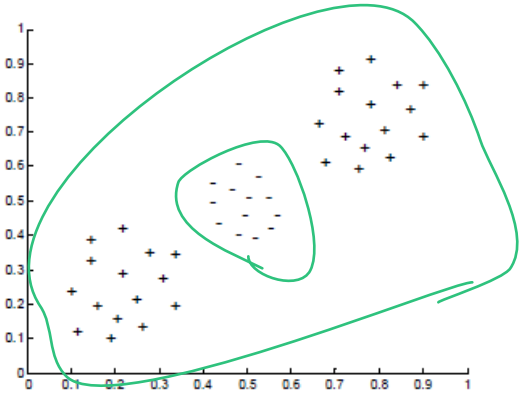
In this question you will compare the decision boundaries of Gaussian Naive Bayes (GNB) and Logistic Regression (LR). For each of the datasets circle true (T) or false (F), based on whether the method can separate the two classes.

- If the answer for Logistic Regression is true, draw the decision boundary.
- If the answer for Gaussian Naive Bayes is true, draw the shape of the two Gaussian bumps (center and the variance).
- If the answer for any of Gaussian Naive Bayes or Logistic Regression is false, please give a one sentence explanation.



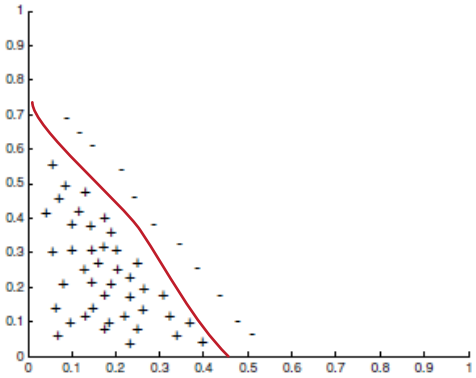
GNB can separate: T F

LR can separate: T F



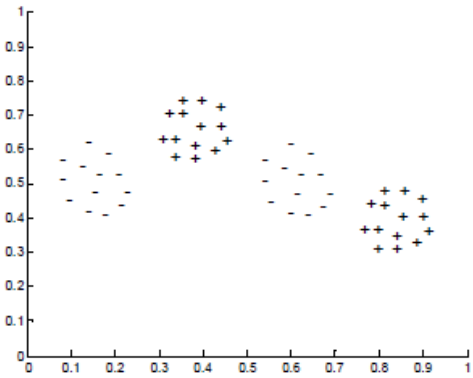
GNB can separate: T F

LR can separate: T F



GNB can separate: T F

LR can separate: T F



GNB can separate : T F

LR can separate: T F

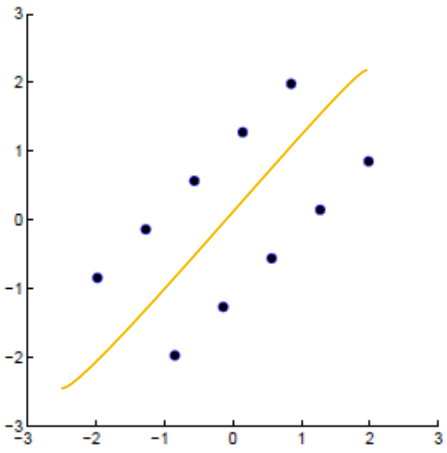
2. Dimensionality reduction - PCA [20 points]

Note that PCA should be used with caution for classification problems, because it does not take information about classes into account. In this problem you will show that, depending on the dataset, the results may be very different. Suppose that the classification algorithm is 1-nearest-neighbor, the source data is 2-dimensional and PCA is used to reduce the dimensionality of data to 1 dimension. There are 2 classes (+ and −). The datapoints (without class labels) is pictured on plots below (the two plots are identical).

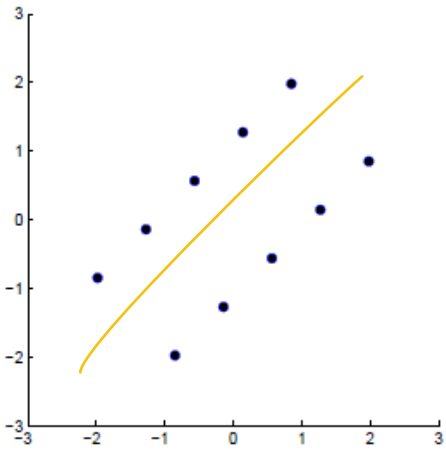
(a) (4 points) On one of the plots draw a line that PCA will project the datapoints to.

(b) (16 points) For each of the plots, label the source datapoints so that 1-NN will have the following leave-one-out cross-validation error:

2D data: 100% error
1D data from PCA: 0% error



2D data 0% error
1D data from PCA: 100% error

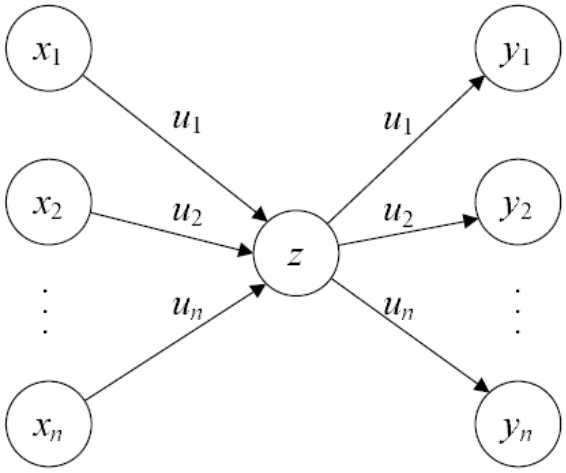


3. Neural networks and PCA [30 points]

It turns out that neural networks can be used to perform dimensionality reduction. We are given a dataset $x^1, \cdots, x^m$, where each point is a n-dimensional vector: $x^i = (x_1^i, \cdots, x_n^i)$. Suppose that the dataset is centered:

$$\bar{x} = \sum_{i=1}^{m} x^i = 0.$$

Consider the following network with 1 node in the hidden layer:



In this network, the hidden and the output layer share the weight parameters, i.e., the edge $x_j \to z$ uses the same weight u_j as the edge $z \to y_j$. The nodes have linear response functions:

$$z = \sum_{j=1}^{n} u_j x_j, \; y_j = u_j z.$$

Typically, in neural networks, we have training examples of the form (x^i, t^i). Here, the network is trained as follows: for each point x^i of our dataset, we construct a training example (x^i, x^i) that assigns the point to both the inputs and the outputs, i.e., $t^i = x^i$.

(a) (6 points) Suppose that we minimize a square loss function,

$$l(\mathbf{w}; \mathbf{x}) = \sum_i \sum_j (t_j^i - \text{out}_j(x^i; \mathbf{w}))^2,$$

where $\text{out}(\mathbf{x}; \mathbf{w})$ is the prediction $\mathbf{y}$ of the network on input $\mathbf{x}$, given weights $\mathbf{w}$. Write down the loss function in terms of the network parameters $\mathbf{u}$ and the way we are training this neural network.

(b) (9 points) Recall that, in principal component analysis with m components, we approximate a data point x^i, by projecting it onto a set of basis vectors $(u^1, \cdots, u^k)$:

$$\hat{x}^i = \bar{x} + \sum_{j=1}^{k} z_j^i u_j,$$

where $z_j^i = x^i \cdot u_j$, in order to minimize the squared reconstruction error:

$$\sum_{i=1}^{m} ||x^i - \hat{x}^i||_2^2,$$

where $||v||_2^2 = \sum_{j=1}^{n} (v_j)^2$. Relate the optimization problem from part 1 to the problem optimized in PCA.

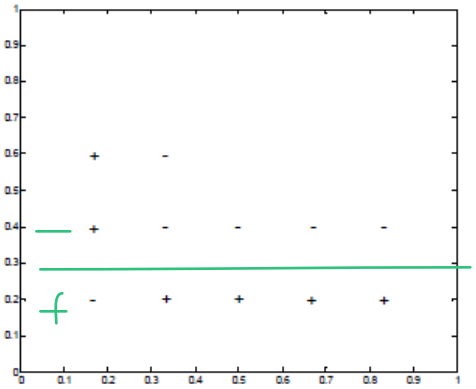
(c) (15 points) Construct a neural network that will result in the same reconstruction of data as PCA with k first principal components. Write down both the structure of the network and the node response functions.

4. Boosting [30 points]

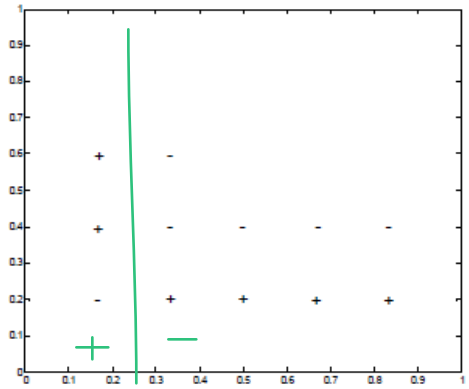
Given a small dataset, we want to learn a classifier that will separate pluses from minuses, using the AdaBoost algorithm. Each datapoint x_i has a class $y_i \in \{+1, -1\}$.

We will be using weak learners where the weak learner's decision boundary is always parallel to one of the axis, i.e., the separating plane is either vertical or horizontal. You can think of weak learners as decision stumps where we split either on X or Y coordinate.

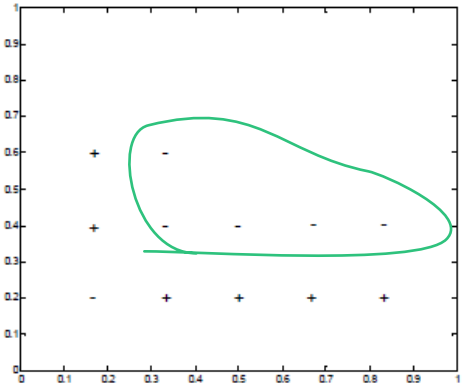
When training the new weak learner $h_t(x)$, we will choose the one that maximizes the weighted classification accuracy with respect to the current weights D_t, i.e., choose h_t that maximizes $\sum_i D_t(i) \cdot \delta(h_t(x_i) = y_i)$. Note that $h_t(x)$ only takes values in $\{+1, -1\}$, depending on whether it classifies x as positive or negative.



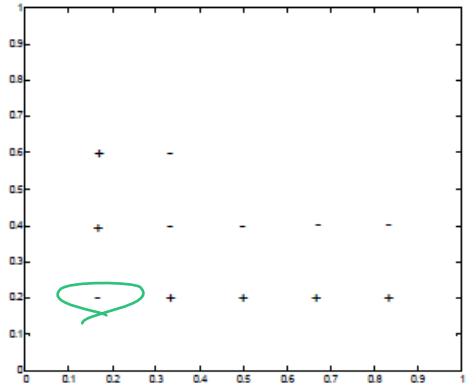
(a) Decision boundary after 1st iteration



(b) Decision boundary after 2nd iteration



(c) Example with *lowest* weight



(d) Example with *highest* weight

(a) (4 points) Draw the decision boundary of boosting after the first iteration (after the first weak learner is chosen) on Figure (a). Don't forget to mark which parts of the plane get classified as + and which as −.

(b) (4 points) Now we do second iteration of boosting. Draw decision boundaries of both weak learners in Figure (b). Boosting combines the decisions of both weak learners. Mark which parts of the plane boosting with 2 weak learners classifies as + and which as −. Use Figure (b).

(c) (6 points) In AdaBoost, we choose α_t as the weight of the t-th weak learner, where $\alpha_t = \frac{1}{2}ln(\frac{1-\epsilon_t}{\epsilon_t})$. Hereby, $\epsilon_t = P_{x\sim D_t}[h_t(x) \neq y]$, i.e. the weighted fraction of examples misclassified by the t-th weak learner. Which one is larger, α_1 or α_2? Give a one sentence explanation of your answer. error rate for fig(a) =3/12

(d) (4 points) After two iterations of boosting, how many training examples are misclassified?

(e) (4 points) Mark which training example(s) have the lowest weight (D_t) after the two iterations of boosting, $t = 2$? Use Figure (c).

(f) (4 points) Mark which training example(s) have the highest weight (D_t) after the second iteration of boosting, $t = 2$? Use Figure (d).

(g) (4 points) Using the dataset from figure and our weak learners, will Boosting ever achieve zero training error? Give a one sentence explanation of your answer.

No, it can not be perfectly split perfectly even if you run infinite times.