Homework 5 – CLT & Ensemble Methods

Exercise 1: For each hypothesis class below, you are asked to:

- Determine the VC-dimension.
- Compute the sample complexity required to guarantee that, with probability at least 1–δ, the true error of a hypothesis is at most ϵ (i.e., to be PAC-learnable), using the standard sample complexity bound:

$$m \geq \frac{1}{\epsilon}\left(4\log_2\left(\frac{2}{\delta}\right) + 8\,\mathrm{VC}\cdot\log_2\left(\frac{13}{\epsilon}\right)\right)$$

(a) One-level decision trees over real-valued vectors in $\mathbb{R}^2$: A one-level decision tree (also called a decision stump) splits based on a single threshold on one of the two features. Use δ=0.05 and ϵ=0.05.

- What is the VC-dimension of this hypothesis class?
- Compute the sample complexity.

(b) Linear separators through the origin in $\mathbb{R}^2$: This class includes all linear classifiers in two dimensions that pass through the origin. Use δ=0.01 and ϵ=0.01.

- What is the VC-dimension?
- Compute the sample complexity.

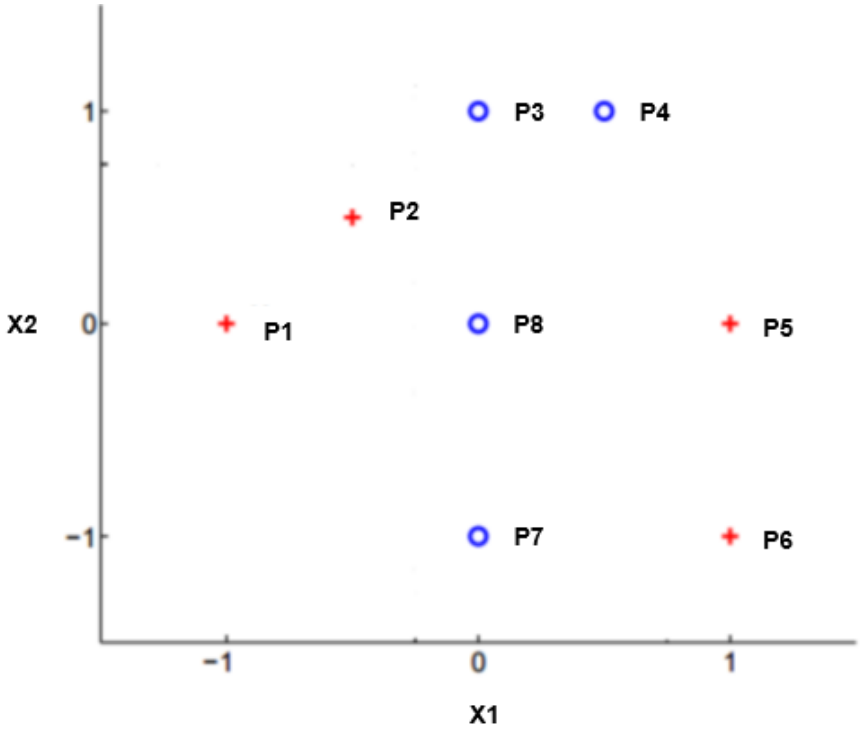
(c) Axis-aligned triangles in $\mathbb{R}^2$: An axis-aligned triangle is defined as a triangle where each side is parallel to one of the coordinate axes. Use δ=0.05 and ϵ=0.01.

- What is the VC-dimension of this hypothesis class?
- Compute the sample complexity.

Exercise 2: Consider the agnostic learner L of the form $(a \leq x1 \leq b) \wedge (c \leq x2 \leq d) \wedge (e \leq x3 \leq f)$. Let a, b be integers in the range [0,199] and c, d, e, f be integers in the range [0,99]. Compute the sample complexity for learner L, by giving the minimum number of training examples sufficient to ensure (with probability 99%) that every hypothesis in H will have a true error of at most 5%.

Exercise 3: Consider the following training dataset

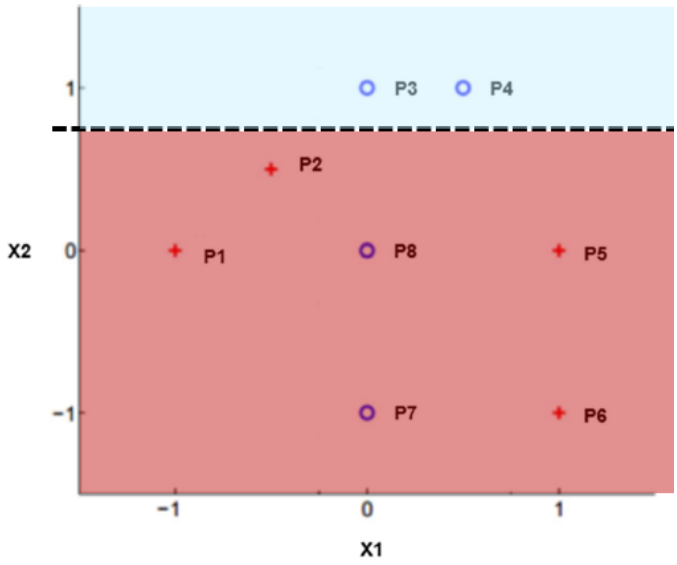
Data Points	X1	X2	Class
P1	-1	0	1
P2	-0.5	0.5	1
P3	0	1	-1
P4	0.5	1	-1
P5	1	0	1
P6	1	-1	1
P7	0	-1	-1
P8	0	0	-1



You are asked to run 3 iterations of AdaBoost with decision stumps given by the weak learners above.

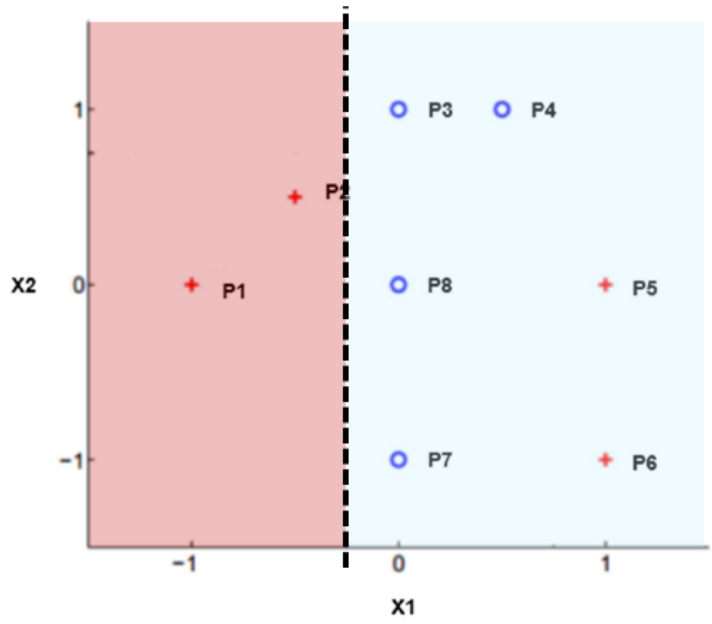
Hypothesis (decision stump) for iteration t=1:

$$h1(x) = \begin{cases} positive, & if\ x2 \leq 0.75 \\ negative, & otherwise \end{cases}$$



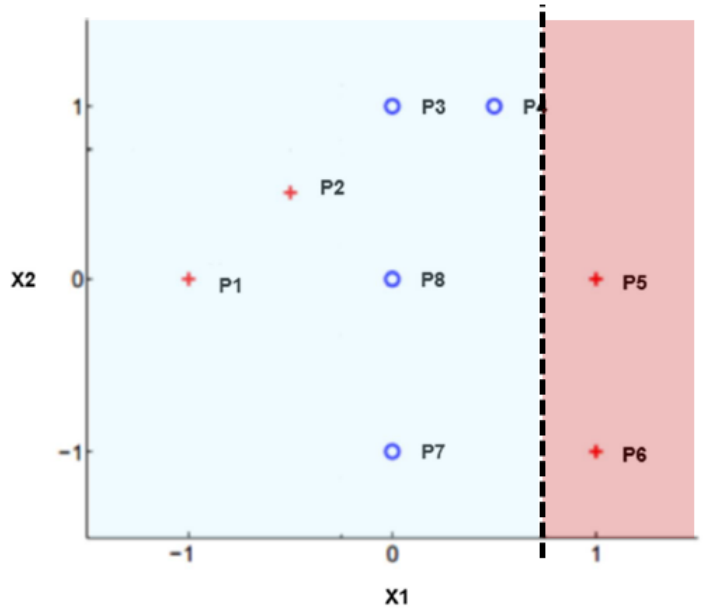
Hypothesis (decision stump) for iteration t=2:

$$h2(x) = \begin{cases} positive, & if\ x1 \leq -0.25 \\ negative, & otherwise \end{cases}$$



Hypothesis (decision stump) for iteration t=3:

$$h3(x) = \begin{cases} positive, & if\ x1 \geq 0.75 \\ negative, & otherwise \end{cases}$$



For each iteration t = 1, 2, 3, compute ε_t, α_t, and the weights $D_t(i)$ (i = 1, 2…8). For each iteration t you will use weak learner h_t.

t	ε_t	α_t	D_t(P1)	D_t(P2)	D_t(P3)	D_t(P4)	D_t(P5)	D_t(P6)	D_t(P7)	D_t(P8)
1										
2										
3										

Finally, after obtaining the model, what is the classification for the unseen data point (-0.25, 1)? You can assume $H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$

Exercise 4 – Coding Exercise (Random Forest): In this assignment, you train a Random Forest model using sklearn in Python and perform sensitivity analysis on key hyperparameters to understand their impact on model performance.

Instructions:

(1) **Dataset**: You can use any dataset of your choice from the `sklearn.datasets` module, or import any external dataset (it can be a dataset from your project), as long as it is a dataset used for the purpose of classification task.

Example:

```
from sklearn.datasets import load_wine
data = load_wine()
X, y = data.data, data.target
```

(2) **Data Preparation:** Split the data into a training set (80%) and a testing set (20%) using train_test_split.

(3) **Model Training:** Train a Random Forest classifier using the default configuration (not changing/setting any hyperparameter).

Evaluate the model on the test set using accuracy as the performance metric.

(4) **Hyperparameter Sensitivity Analysis:** Perform a sensitivity analysis on the following hyperparameters, by testing at least 10 values for each hype-parameter, in the ranges specified below:

a. Number of Trees (n_estimators): [10 to 1000].
b. Maximum Depth of Trees (max_depth): [5 to 100].
c. Minimum Samples per Leaf (min_samples_leaf): [1 to 200].

For each hyperparameter, vary its values while keeping the others fixed at the base configuration. Create a plot of the test accuracy for each hyperparameter value to illustrate how the model's performance responds to variations in the hyperparameter being analyzed.