Cairo University
Faculty of Engineering

Department of Computer Engineering

Assignment 1
Report
Solutions for Handwritten Problems & Plots/Comments from the code

Student Info

Name	Abdelrahman Hamdy Ahmed Mekhaimar
Section	1
BN	38
ID	9202833

Date: 30/03/2024

Submitted to: Eng. Mohammed Shawky

1.6 We have a sample of 10 marbles drawn independently from a bin containing red & green marbles

$P(\text{red}) = \mu \quad P(\text{green}) = 1 - \mu$

v: Number of red marbles drawn

a) One sample is drawn $\rightarrow P(v=0) = ?$

For $\mu = 0.05$ $\quad P(v=0) = (1-0.05)^{10} = \boxed{0.59874}$

For $\mu = 0.5$ $\quad P(v=0) = (1-0.5)^{10} = \boxed{9.7656 \times 10^{-4}}$

For $\mu = 0.8$ $\quad P(v=0) = (1-0.8)^{10} = \boxed{1.024 \times 10^{-7}}$

b) 1000 independent samples

Probability that atleast one of the samples has $v=0$

$= 1 -$ Probability that none has $v=0 = 1-P$

To calculate $P \rightarrow$ Complement of what we did in (a)

For $\mu = 0.05$ $\quad P = (1 - 0.59874)^{1000} = 0$

$\hookrightarrow 1 - 0 = \boxed{1}$

For $\mu = 0.5$ $\quad P = (1 - (9.7656) \times 10^{-4})^{1000} = 0.3764$

$\hookrightarrow 1 - P = 1 - 0.3764 = \boxed{0.6236}$

For $\mu = 0.8$ $\quad P = (1 - (1.024) \times 10^{-7})^{1000} = 0.9998976$

$\hookrightarrow 1 - P = \boxed{1.02395 \times 10^{-4}}$

1.6 c) Repeat (b) but with 1000000 independent samples

$\mu = 0.05$ $P = (1 - 0.59874)^{1000000} = 0$

$\hookrightarrow 1 - P = 1 - 0 = \boxed{1}$

$\mu = 0.5$ $P = (1 - (9.7656 \times 10^{-4}))^{1000000} = 0$

$\hookrightarrow 1 - P = 1 - 0 = \boxed{1}$

$\mu = 0.8$ $P = (1 - (1.024 \times 10^{-7}))^{1000000} = 0.90267$

$\hookrightarrow 1 - P = 1 - 0.90267 = \boxed{0.09733}$

2.5 Prove by induction that

$$\sum_{i=0}^{D} (N\ i) \leq N^D + 1, \text{ hence } m_H(N) \leq N^{d_{vc}} + 1$$

To prove by mathematical induction, we need to show that the statement holds first for the base case

Then, assume that the statement is true for some value N & prove that it's also true for N+1.

- $N = 1, D = 0 \rightarrow \sum_{i=0}^{D} NC_i \leq N^D + 1$

  $1C0 \leq 1^0 + 1 \rightarrow 1 \leq 2$ ✓ holds

- $N = 1, D = 1 \rightarrow 1C0 + 1C1 \leq 1^1 + 1 \rightarrow 2 \leq 2$ ✓ holds

  } base case validated

- $N = 2\ D = 1 \rightarrow 2C0 + 2C1 \leq 2^1 + 1 \rightarrow 3 \leq 3$ ✓

Now, assume that $\sum_{i=0}^{D} NC_i \leq N^D + 1$

Prove $\sum_{i=0}^{D} (N+1)C_i \leq (N+1)^D + 1$ $\longrightarrow$ Need to get back to the (N) Form

We know that $N+1\,C_i = NC_i + NC_{i-1}$ $\longleftarrow$

$$\sum_{i=0}^{D} N+1\,C_i = \sum_{i=0}^{D} NC_i + NC_{i-1}$$

$$= \sum_{i=0}^{D} NC_i + \sum_{i=0}^{D} NC_{i-1}$$

$$= \sum_{i=0}^{D} NC_i + \sum_{k=0}^{D-1} NC_k \leq N^D + 1 + N^{D-1} + 1$$

$$\leq \underbrace{N^D + N^{D-1} + 1}\; + 1$$

we also know that $(N+1)^D$ is greater than this so we can replace it since we are on the right side of the $\leq$ inequality

which means $\leq (N+1)^D + 1$

There we go, we proved $\sum_{i=0}^{D} N+1\,C_i \leq (N+1)^D + 1$

Since $m_H(N) \leq \sum_{i=0}^{dvc} NC_i$ & $\sum_{i=0}^{dvc} NC_i \leq N^{dvc} + 1$ [proven]

$$\therefore\; m_H(N) \leq N^{dvc} + 1$$

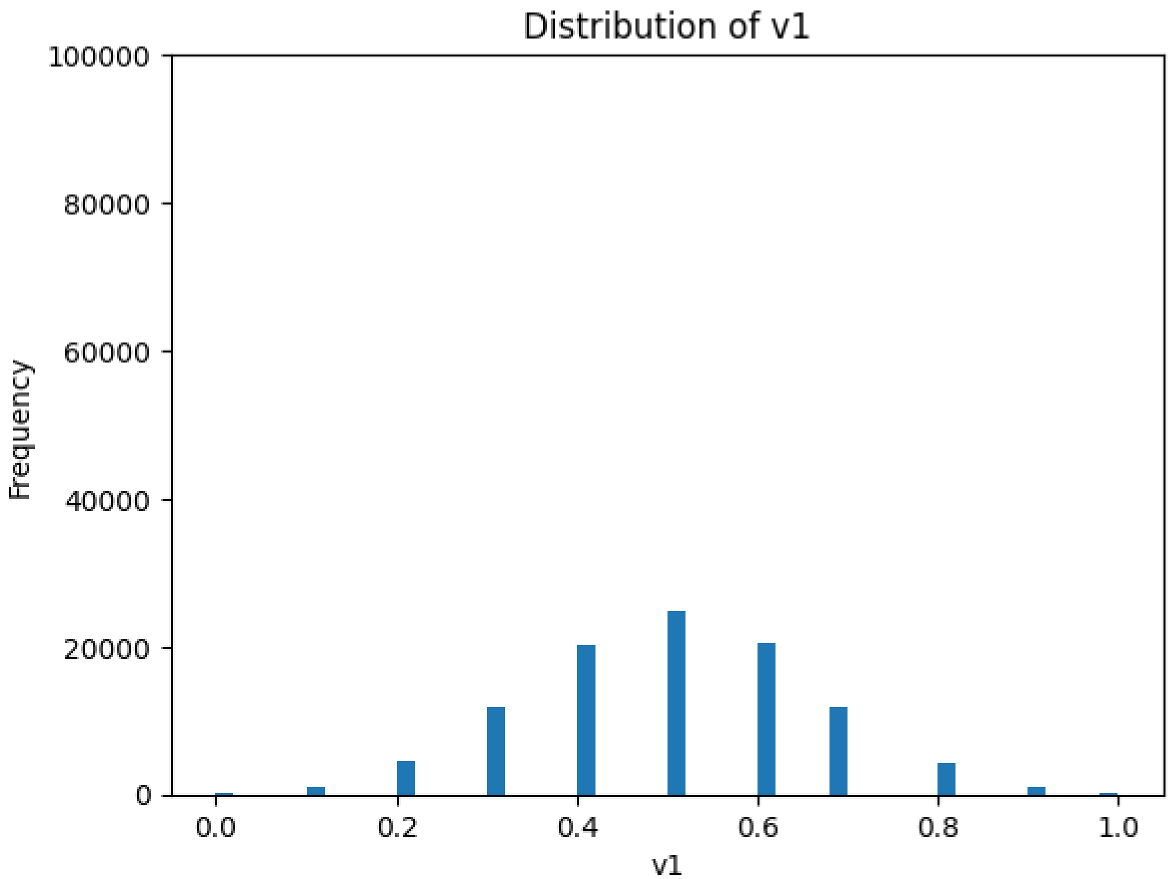
from the lecture

it was given a theorem that if H has a breakpoint then $m_H(N)$ is bounded by a polynomial in N

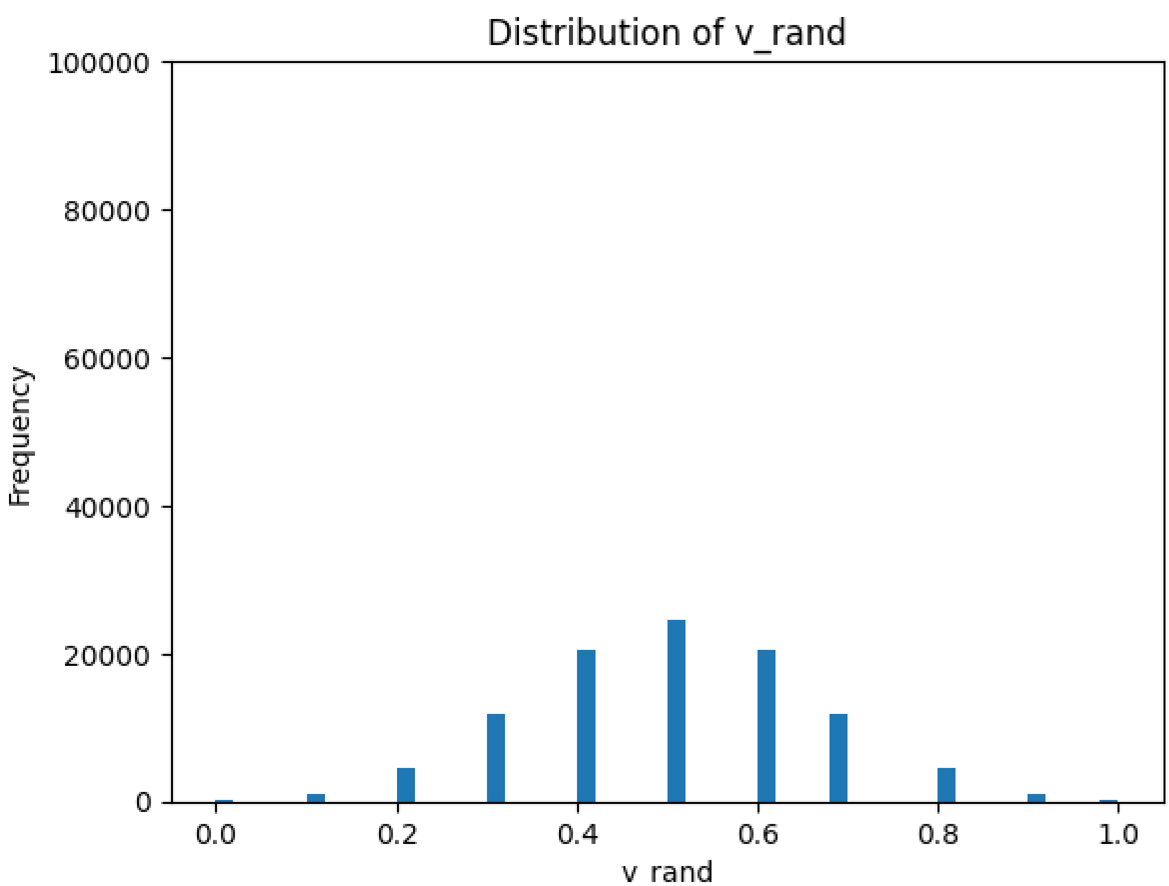
$$m_H(N) \leq \sum_{i=0}^{dvc(H)} \binom{N}{i} \;\cancel{=}$$

Plots

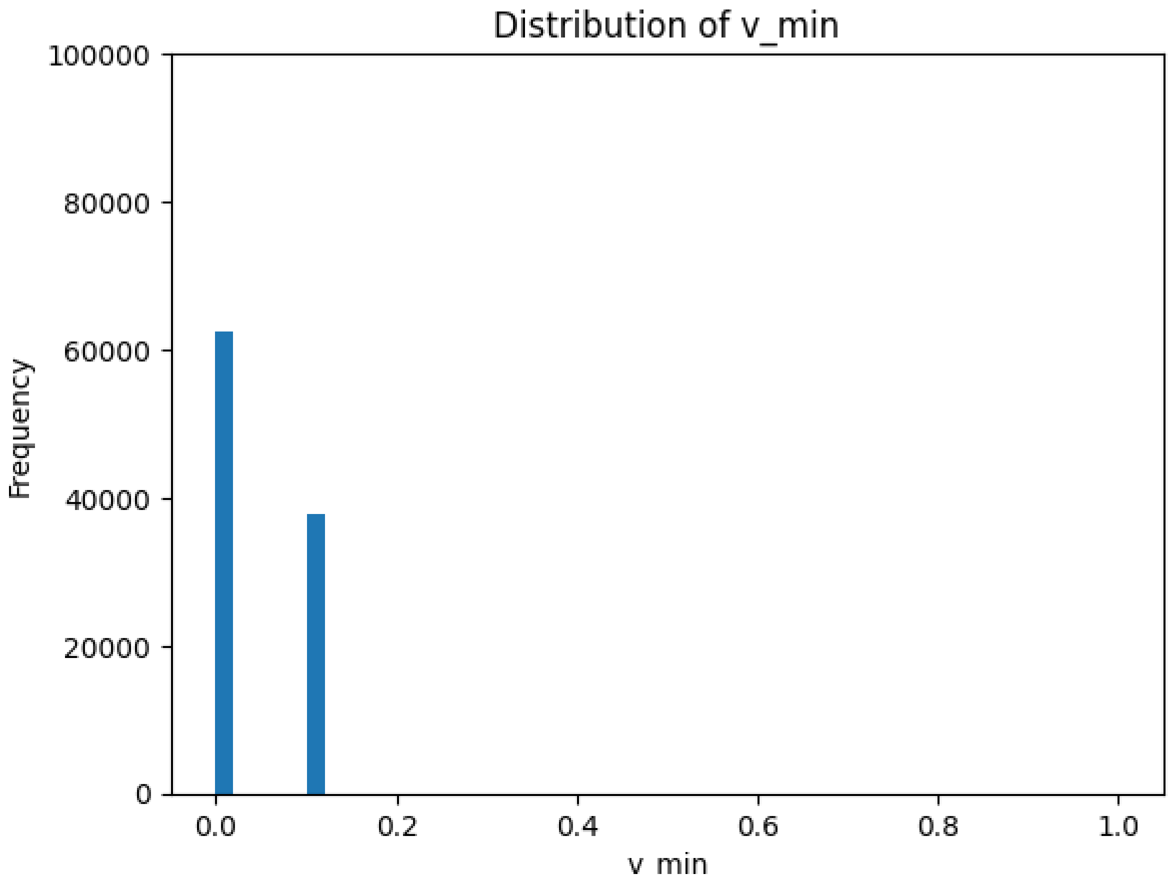
1.10 (b)



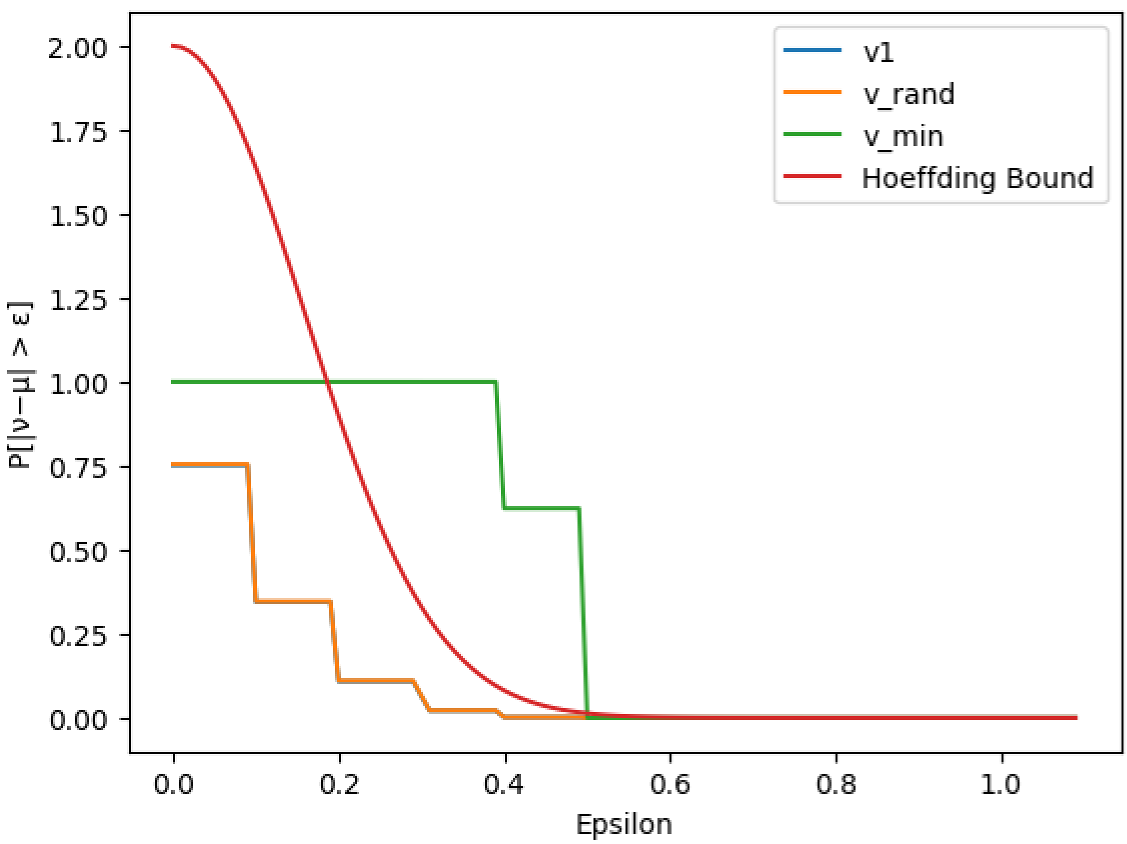
Distribution of v1
Frequency
v1
100000
80000
60000
40000
20000
0
0.0
0.2
0.4
0.6
0.8
1.0



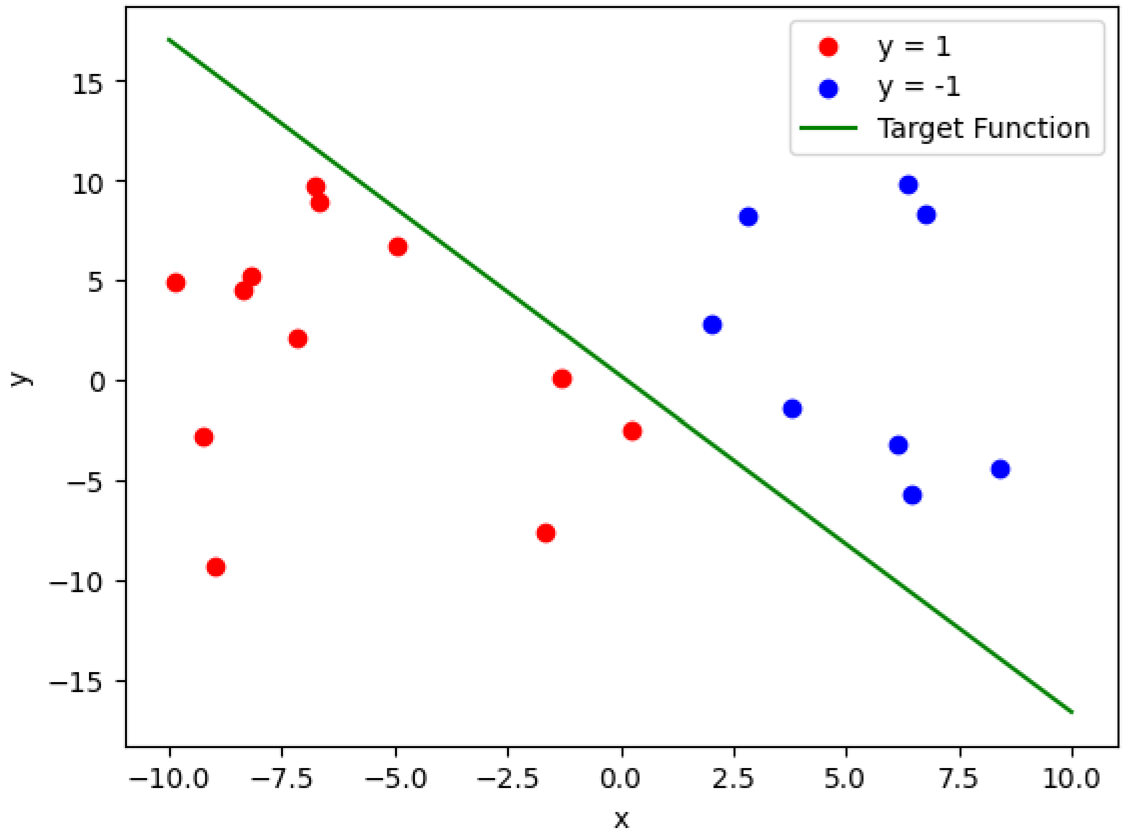
Distribution of v_rand
Frequency
v_rand
100000
80000
60000
40000
20000
0
0.0
0.2
0.4
0.6
0.8
1.0


Distribution of v_min
Frequency
100000
80000
60000
40000
20000
0
0.0
0.2
0.4
0.6
0.8
1.0
v_min

1.10 (c)

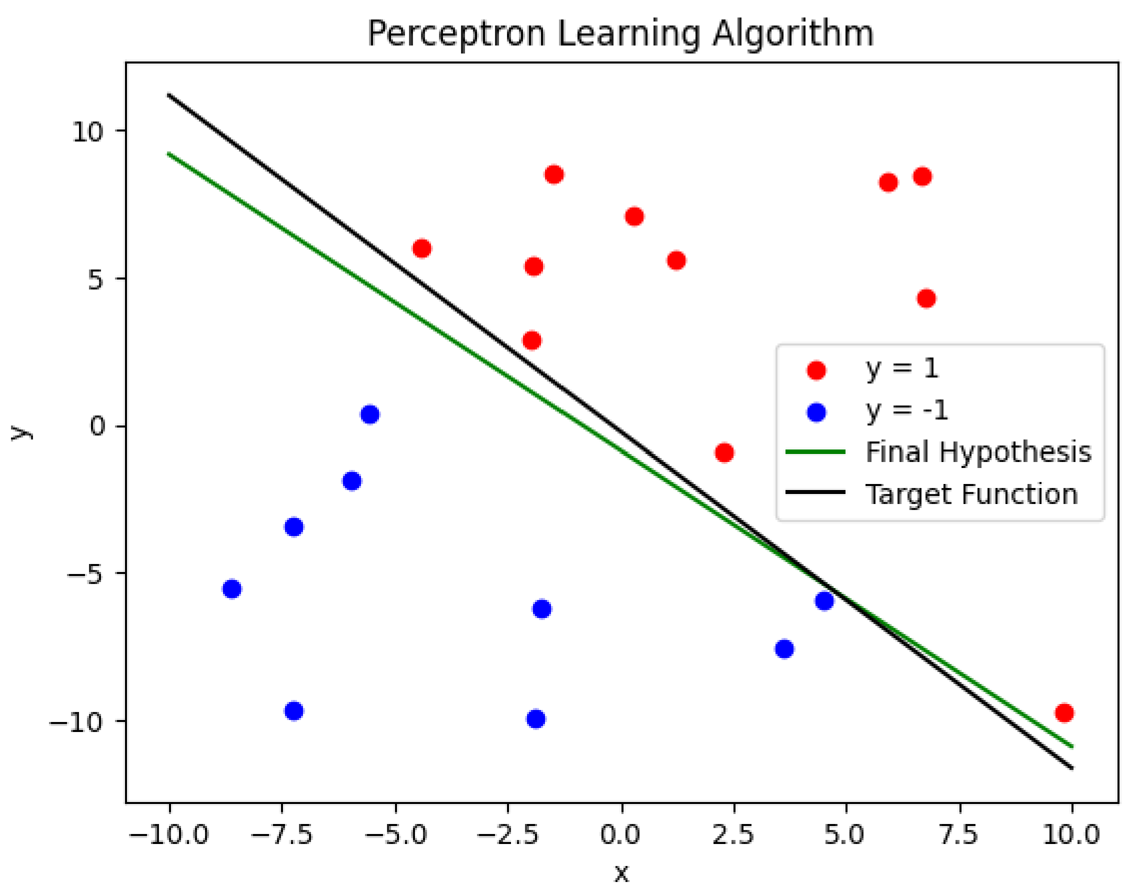

v1
v_rand
v_min
Hoeffding Bound
P[|ν−μ| > ε]
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0.0
0.2
0.4
0.6
0.8
1.0
Epsilon

1.4 (a)



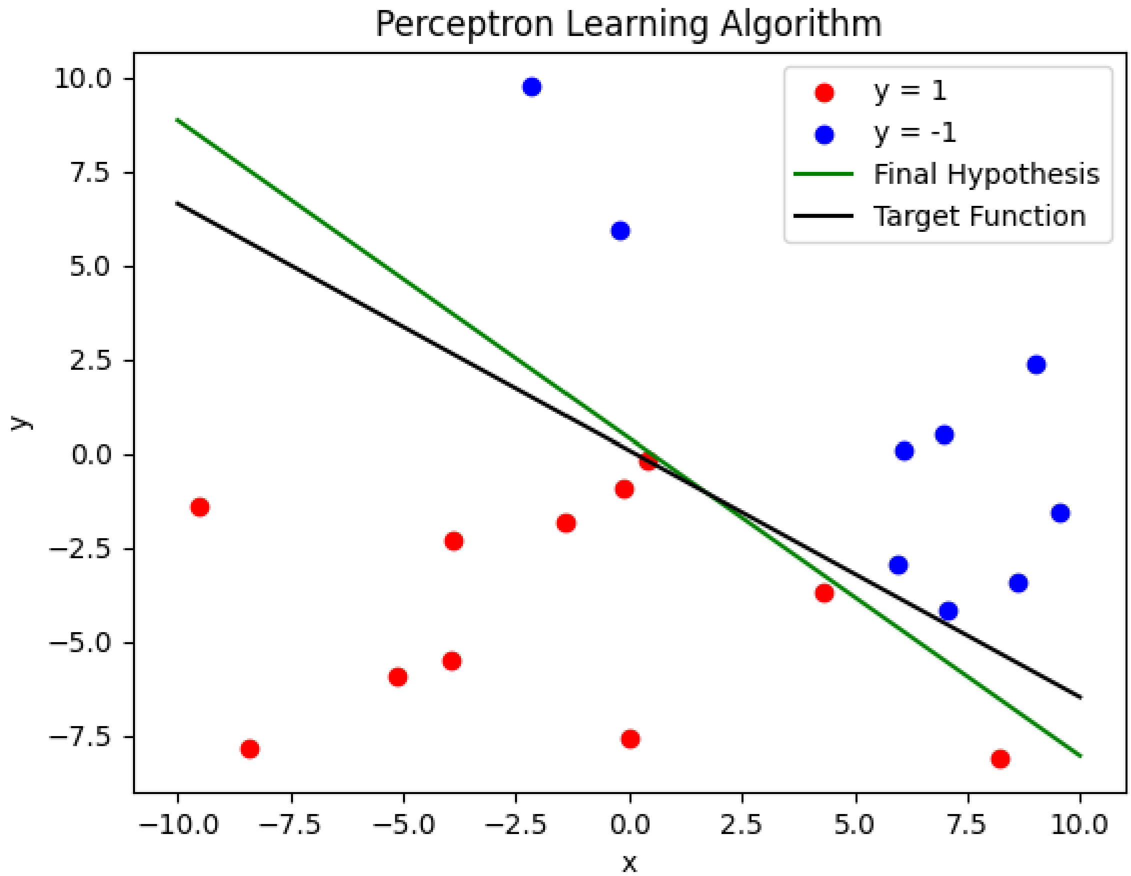
y = 1
y = -1
Target Function
15
10
5
0
−5
−10
−15
y
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
x

1.4 (b)

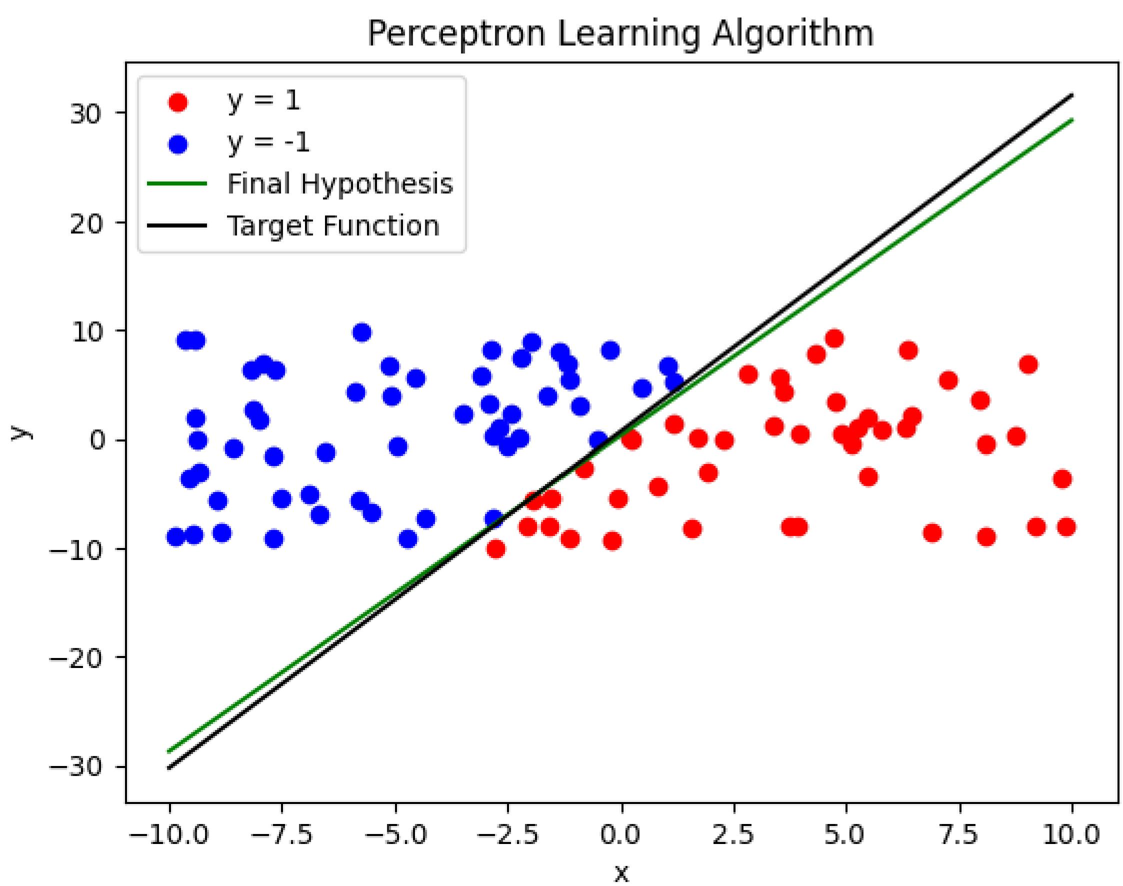


Perceptron Learning Algorithm
y = 1
y = -1
Final Hypothesis
Target Function
10
5
0
−5
−10
y
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
x

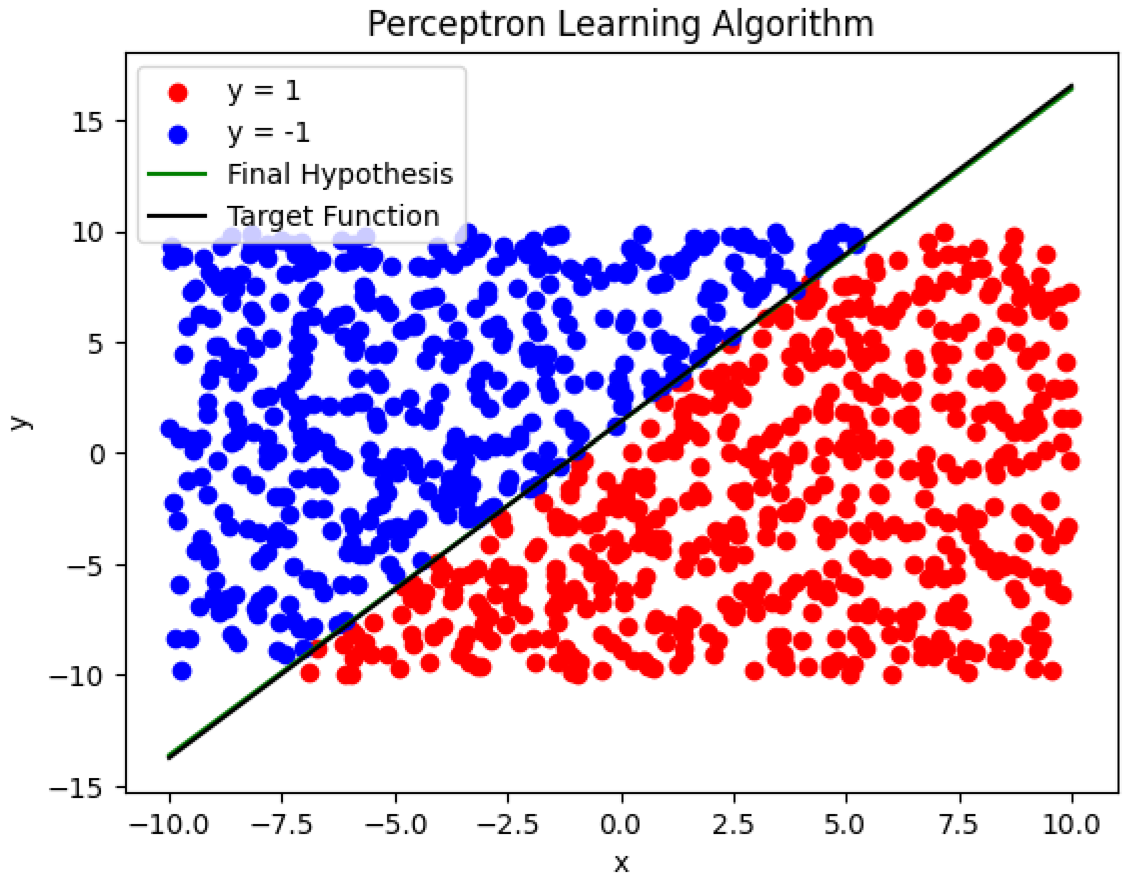
1.4 (c)


Perceptron Learning Algorithm
y = 1
y = -1
Final Hypothesis
Target Function
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
x
y

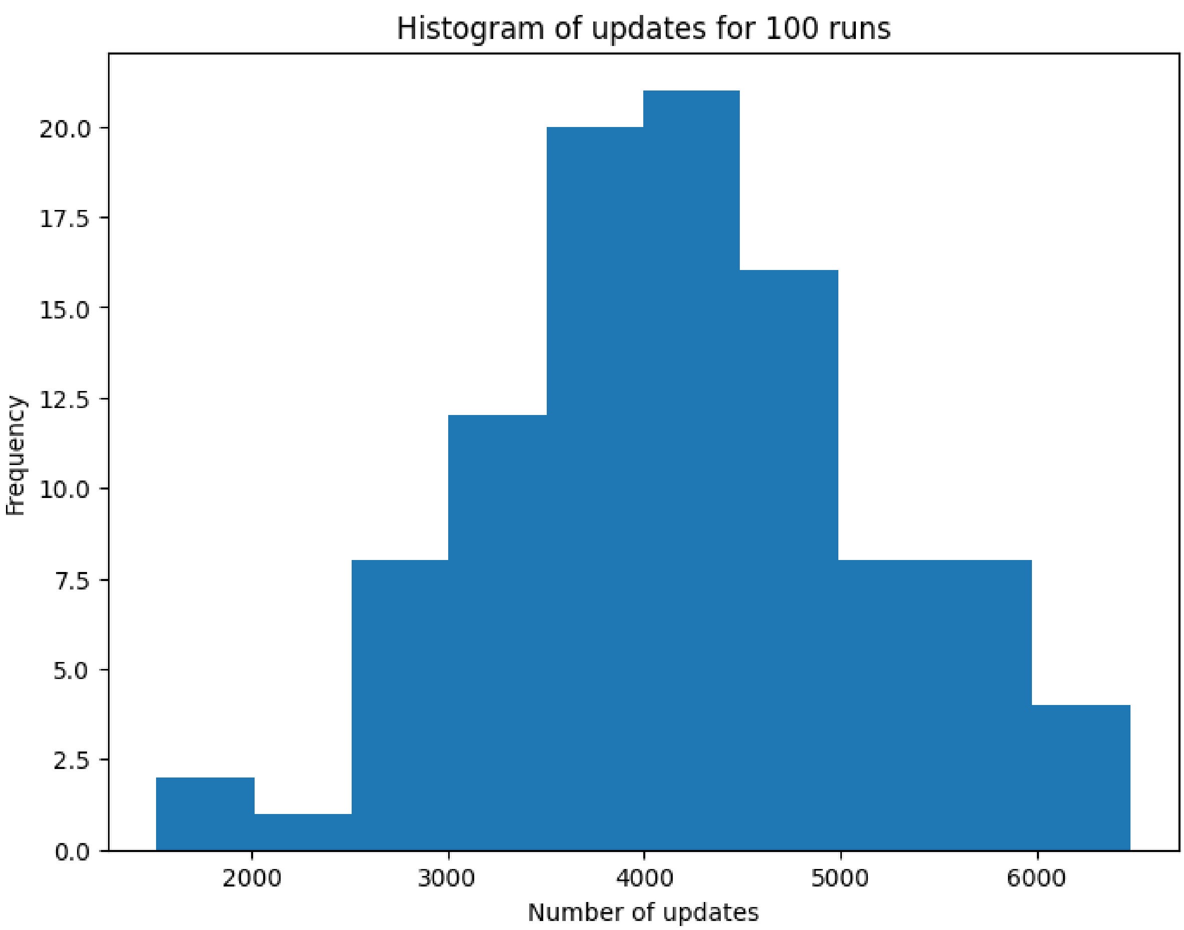
1.4 (d)


Perceptron Learning Algorithm
y = 1
y = -1
Final Hypothesis
Target Function
30
20
10
0
−10
−20
−30
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
x
y

1.4 (e)


Perceptron Learning Algorithm
y = 1
y = -1
Final Hypothesis
Target Function
y
x
15
10
5
0
−5
−10
−15
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

1.4 (g)


Histogram of updates for 100 runs
Frequency
Number of updates
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
2000
3000
4000
5000
6000

Comments

1.10 (d)

The v1 and v_rand coins obey the Hoeffding bound, while the v_min coin does not. The Hoeffding bound is based on the probability of the observed frequency of heads differing from the true frequency of heads by more than a certain amount and it assumes that all samples are random and independent. The v1 and v_rand coins are independent of the number of coins flipped and are completely random, so they obey the Hoeffding bound. The v_min coin, however, is dependent on the number of coins flipped, so it does not obey the Hoeffding bound and for it to do so, it needs a looser bound that can satisfy it.

1.10 (e)

The multiple bins in Figure 1 represent the different coins that were flipped. This means that after conducting the experiments, the strategy selects the bin with the lowest error. This is similar to the v_min strategy, which selects the coin with the lowest fraction of heads. Since the bin with the lowest fraction of heads is not random and independent, it does not obey the Hoeffding bound.

1.4 (b)

g is very close to f since the data is linearly separable. The perceptron learning algorithm is guaranteed to converge (Scientifically proven) and the final hypothesis g will be close to the target function f.

1.4 (c)

g is closer to f in (b) than in (c) because the data is more spread out in (b) than in (c). However, it is still close to f in both cases because the data is linearly separable. The comparison is completely random because both (b) and (c) generate data of the same size and dimensionality.

1.4 (d)

When we compare hypotheses f and g, we see that g is closer to f, and this difference is more noticeable than in situation b.

Hoeffding's inequality tells us that when we have more data points, there's a higher chance that our hypothesis function will be close to the target function. This idea supports our findings. With more training points, the chance of our hypothesis function being close to the target function increases because the bound decreases. However, having more data points doesn't guarantee that the hypothesis function will be closer to the target function compared to situation b.

In simpler terms, adding more data points improves how accurately our hypothesis function can predict outcomes.

1.4 (e)

The number of training points is very high compared to b, so the bound decreases significantly and g becomes closer to f. This is consistent with the idea that the more data points we have, the more accurate our hypothesis function will be.

1.4 (h)

The number of dimensions and the number of training examples don't affect how many points are classified correctly. However, when it comes to making predictions that apply to new, unseen data (generalization), having more training examples makes our hypothesis function closer to the true function. This idea is supported by Hoeffding's inequality. Increasing the number of training examples makes the average of our sample data closer to the average of the entire population.

Adding more training examples also increases the complexity of the algorithm because it requires more updates to the model. Similarly, increasing the number of dimensions also increases the complexity because it requires more calculations to handle each data point.