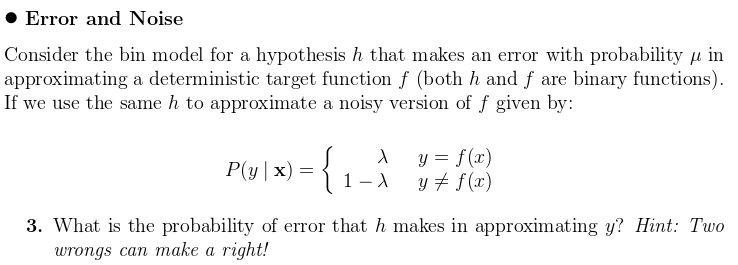
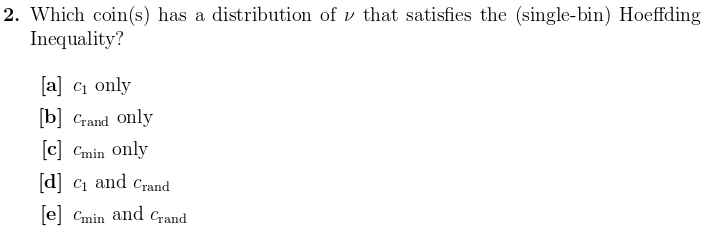


For the program with which this was generated, refer to <https://github.com/cmishra/Learning-from-data-coursework/tree/master/HW1%20-%20PLA%20Implementation>.





is the original (deterministic) target function. is the new, stochastic target function that returns the probability assigned to event given input based upon the distribution of defined above. is an arbitrary hypothesis function.

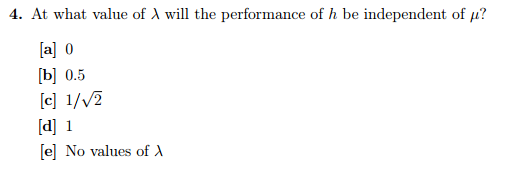
Recall we are given:

Derivation:

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

Correction: I found when I was meant to find .

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |



Recall from the last question:

This can be rearranged to:

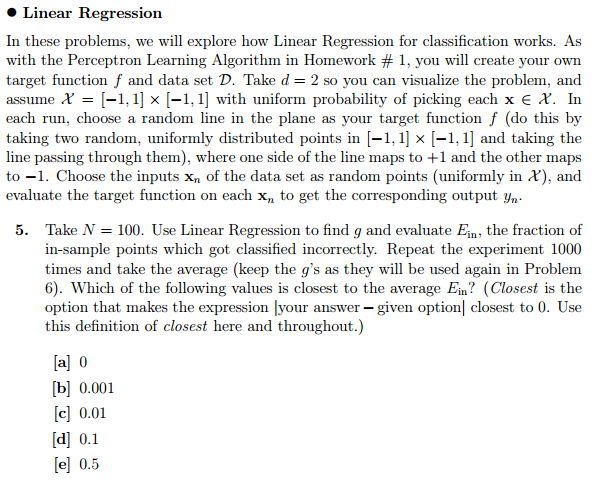
Since we have the added at the end, we can say that regardless of the value of will always matter.

Correction: #3 was wrong, so this was too. Correct answer is below:

Recall from the last question:

This can be rearranged to:

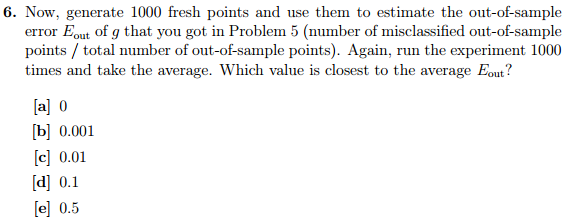
Since is 0 when , is reduced to , and is independent of .



My implemented program had an average of 0.17. The implemented program can be found at <https://github.com/cmishra/Learning-from-data-coursework/blob/master/hw2Programs/src/LinRegImplementation.java>.

Correction: Output was 0/1 instead of -1/1 (accuracy higher with the latter)

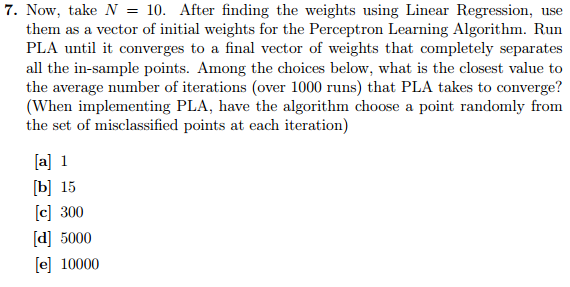
My implemented program had an average of 0.02. The implemented program can be found at <https://github.com/cmishra/Learning-from-data-coursework/blob/master/hw2Programs/src/LinRegImplementation.java>.



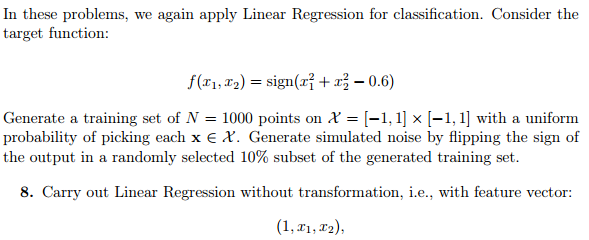
My implemented program had an average of 0.17. This tracks the in-sample error phenomenally well. The implemented program can be found at <https://github.com/cmishra/Learning-from-data-coursework/blob/master/hw2Programs/src/LinRegImplementation.java>.

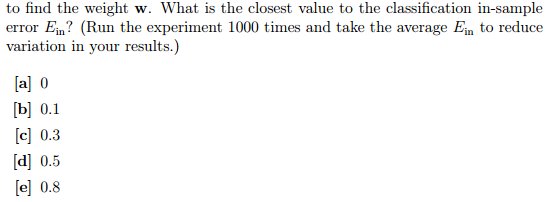
Correction: Output was 0/1 instead of -1/1 (accuracy higher with the latter)

My implemented program had an average of 0.02. The implemented program can be found at <https://github.com/cmishra/Learning-from-data-coursework/blob/master/hw2Programs/src/LinRegImplementation.java>.

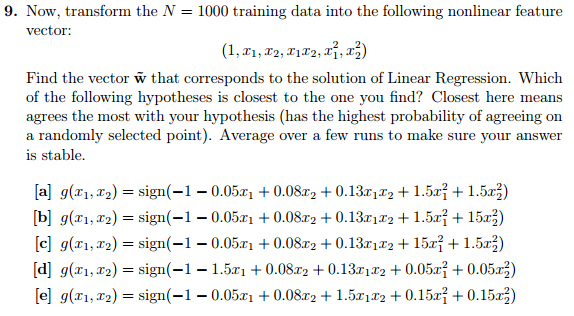


My implemented program had an average iteration count of 3.55. This is much smaller than initializing the weights at 0 (). The implemented program can be found at <https://github.com/cmishra/Learning-from-data-coursework/blob/master/hw2Programs/src/PLA.java>



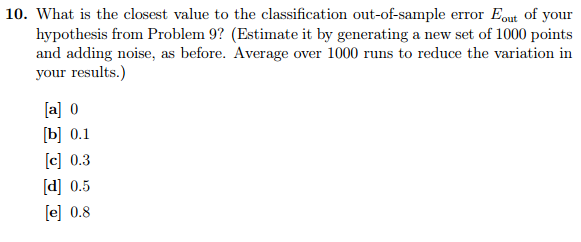


The average came out as 0.58. Implementation program is at <https://github.com/cmishra/Learning-from-data-coursework/blob/master/hw2Programs/src/LinRegNonLinear.java>



The values averaged over 1k iterations of 1k data points:

|  |  |
| --- | --- |
|  | -0.9920648741983545 |
|  | -0.0010410242800370714 |
|  | -6.064162330244251E-4 |
|  | 0.0011122887682191824 |
|  | 1.557843289179927 |
|  | 1.5581759165173998 |
|  |  |



The value of . Implementing program can be found at <https://github.com/cmishra/Learning-from-data-coursework/blob/master/hw2Programs/src/LinRegNonLinear.java>.

Correction: Accidentally relied on old code that had outputs coded as 0/1, not -1/1

The value of . Implementing program can be found at <https://github.com/cmishra/Learning-from-data-coursework/blob/master/hw2Programs/src/LinRegNonLinear.java>

Grade:

Incorrect answers:

* 3 found instead of
* 4 (answer relied on #3)
* 5 (output previously mapped to 0/1 instead of -1/1)
* 6 (answer relied on #5)
* 10 (accidentally used old code mapping outputs to 0/1 instead of -1/1)

