Homework 1

Kaitlyn Mulligan

2/13/19

1 Instructions

This assignment should be written in LaTeX. Please check the template that has been provided to you in `hw.template.tar.gz` that contains the basic elements for you to get started. You must show your work and how you derivated or arrived at your solution. Write in understandable, easy to follow, English.

Write all your programs in Python and make sure that your programs are included in your assignment. Also include any and all figures and graphs produced using Python in your writeup.

Your assignment should be submitted in two ways: through GitHub, and in hardcopy (in class). You should already have a **single** repository, that will be used for all assignments organized in folders named "`lastname-xx`", where lastname is your last name (or the first work of your last name if you have multiple words in it), and xx is the number of the assignment. For example my GitHub folder for this assignment would be `rivas-01` and I would put all my sources there. Writeup: latex or word file (.tex/.doc), main pdf file (.pdf), and any graphics (.pdf or .eps). Code: Python files (.py).

2 Problem Set

The following is a list of problems you will solve. These are from your textbook. When providing your solutions (hopefully using LaTeX), do not simply give the final answer, show how you arrived to the solution, justify your assumptions, and explain your results clearly.

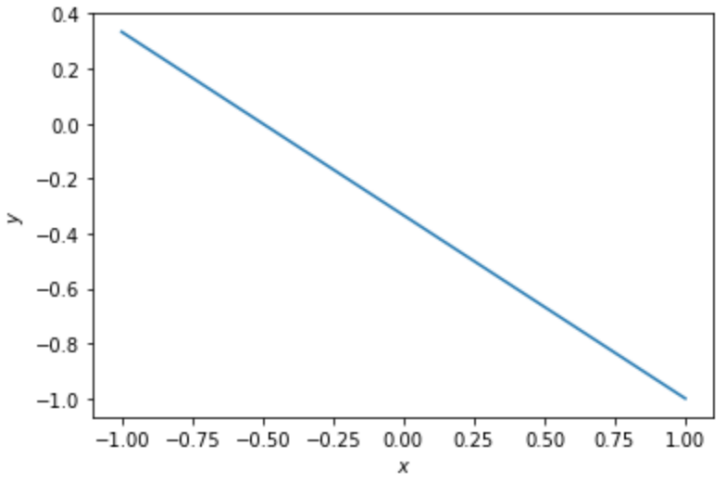
- (extra credit) **Problem 1.2**. Use Python with matplotlib to create the drawings.
- **Problem 1.4**. Again, use Python and the necessary packages (installed for Homework 0) to do your experiments. The purpose is for you to illustrate and shed light on otherwise obscure machine learning concepts, not to produce pages and pages of code.

2.1 Problem 1.2 (extra credit)

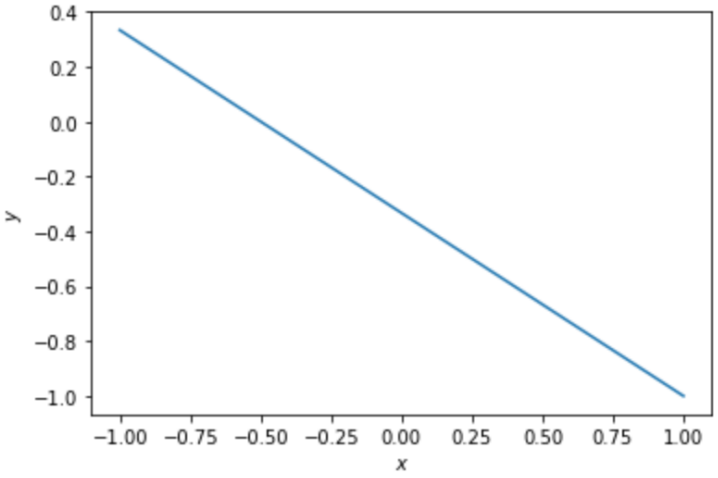
Consider the perceptron in two dimensions: $h(\mathtt{x}) = \text{sign}(\mathtt{w\hat{}Tx})$ where $\mathtt{w} = [w_0, w_1, w_2]^T$ and $\mathtt{x} = [1, x_1, x_2]^T$. Technically, $\mathtt{x}$ has three coordinates, but we call this perceptron two-dimensional because the first coordinate is fixed at 1.

(a) Show that the regions on the plane where $h(x) = +1$ and $h(x) = -1$ are separated by a line. If we express this line by the equation $x_2 = ax_1 + b$, what are the slope a and intercept b in terms of w_0, w_1, w_2?
Solution: If we have $h(x) = +1$ we know that $\mathtt{w}^T x$ will be greater than 0. Also if we have $h(x) = -1$ we know that $\mathtt{w}^T x$ is less than 0. From this we can conclude that somewhere between these two regions, there is a line who equation can be written as $\mathtt{w}^T x = 0$ or $w_0 + w_1 x_1 + w_2 x_2 = 0$. This can be rewritten as $x_2 = ax_1 + b$. From these equations we can determine that the slope, a, is$a = -w_1/w_2$ and the intercept, b, is $b = -w_0/w_2$.

(b) Draw a picture for the cases $\mathtt{w} = [1, 2, 3]^T$ and $\mathtt{w} = -[1, 2, 3]^T$.
Solution: For $\mathtt{w} = [1, 2, 3]^T$ we have the following:



For $\mathtt{w} = -[1, 2, 3]^T$ we have the following:

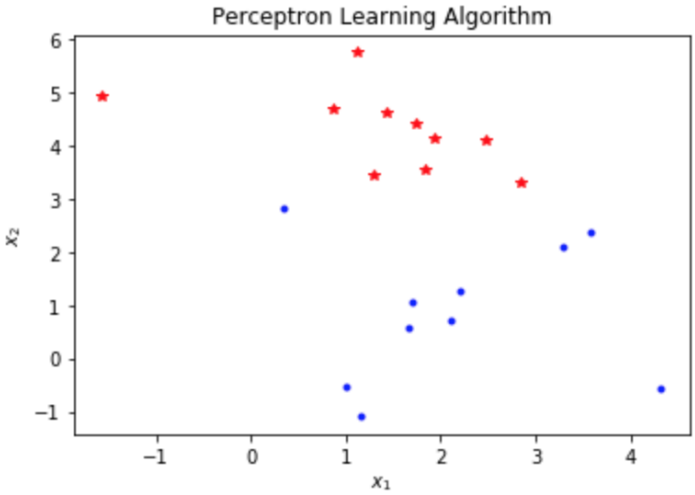


In more than two dimensions, the $+1$ and -1 regions are separated by a *hyperplane*, the generalization of a line.

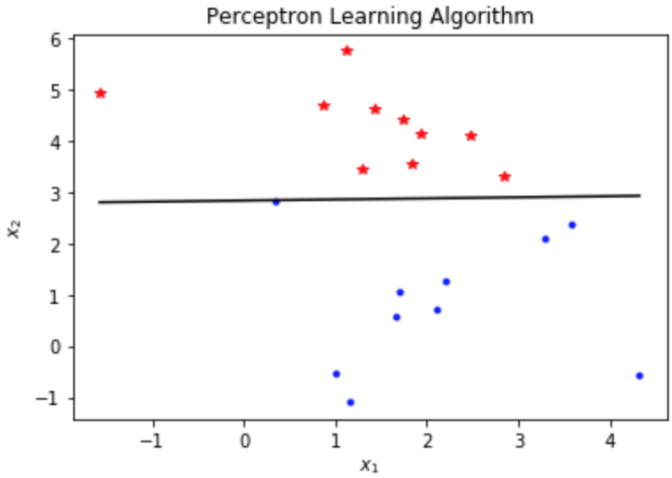
2.2 Problem 1.4

In Exercise 1.4, we use an artificial data set to study the perceptron learning algorithm. This problem leads you to explore the algorithm further with data sets of different sizes and dimensions.

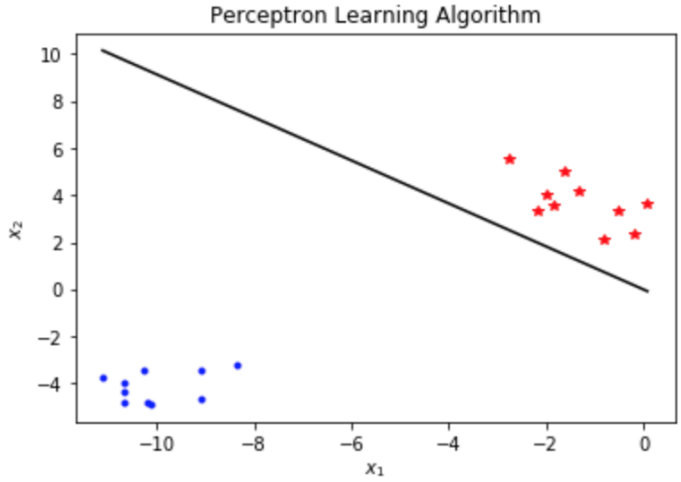
(a) Generate a linearly separable data set of size 20 as indicated in Exercise 1.4. Plot the examples $\{(\mathtt{x}_n, y_n)\}$ as well as the target function f on a plane. Be sure to mark the examples from different classes differently, and add labels to the axes of the plot.
Solution:



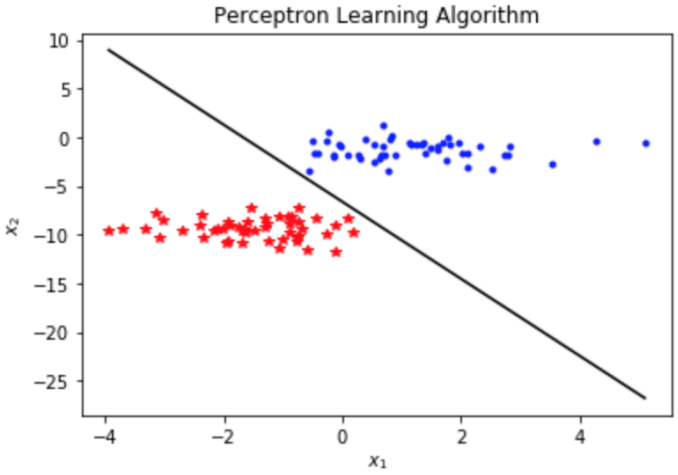
(b) Run the perceptron learning algorithm in the data set above. Report the number of updates that the algorithm takes before converging. Plot the examples $\{(\mathbf{x}_n, y_n)\}$, the target function f, and the final hypothesis g in the same figure. Comment on whether f is close to g.
Solution: The algorithm took 30 iterations before converging.



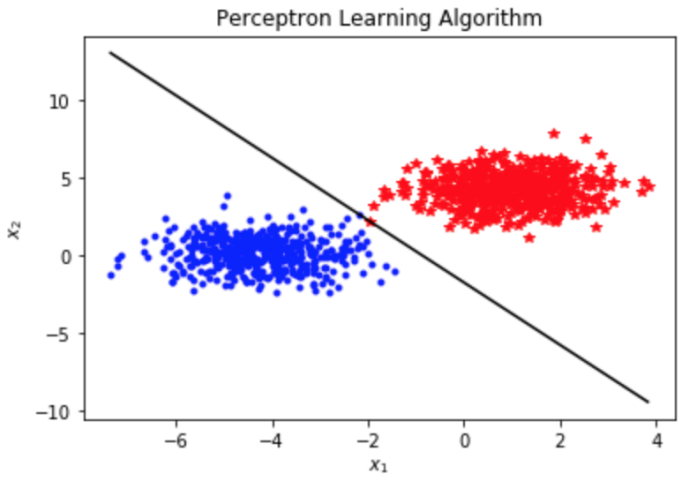
(c) Repeat everything in (b) with another randomly generated data set of size 20. Compare your results with (b).
Solution: With another randomly generated data set of size 20, the algorithm only took 2 iterations to converge. This data set took a lot less number of iterations than the previous data set. Looking at the graph, we can see this data was originally grouped together, which allowed the algorithm to converge faster.



(d) Repeat everything in (b) with another randomly generated data set of size 100. Compare your results with (b).
Solution: With a randomly generated data set of size 100, the algorithm was able to converge in 27 iterations. Compared to part (b), the algorithm took less iterations to converge, but this data was grouped better than it was in part (b).

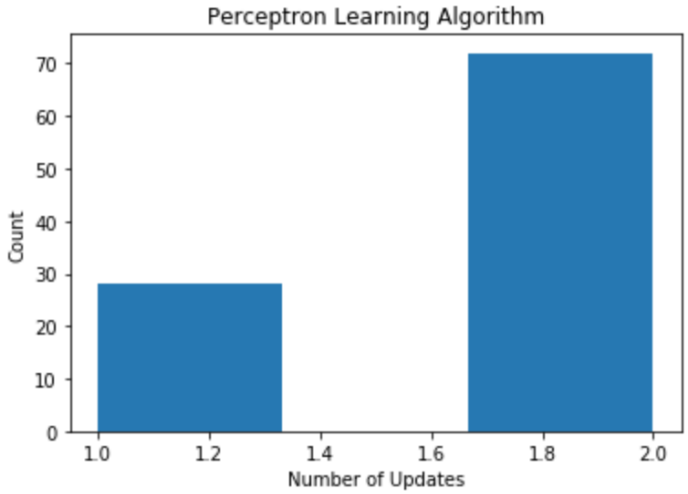


(e) Repeat everything in (b) with another randomly generated data set of size 1000. Compare your results with (b).
Solution: With a randomly generated data set of size 1000, the algorithm converged in 298 iterations.



(f) Modify the algorithm such that it takes $\mathbf{x}_n \in \mathbb{R}^{10}$ instead of $\mathbb{R}^2$. Randomly generate a linearly separable data set of size 1000 with $\mathbf{x}_n \in \mathbb{R}^{10}$ and feed the data set to the algorithm. How many updates does the algorithm take to converge?
Solution: After modifying the algorithm, a randomly generated data set of size 1000, with $\mathbf{x}_n \in \mathbb{R}^{10}$, the algorithm was able to converge in 2 iterations.

(g) Repeat the algorithm on the same data set as (f) for 100 experiments. In the iterations of each experiment, pick $\mathbf{x}(t)$ randomly instead of deterministically. Plot a histogram for the number of updates that the algorithm takes to converge.
Solution: After repeating the algorithm on the same data set as part (f) for 100 experiments, below is a histogram to show the count of number of iterations each experiment took to converge. As can be seen, it looks to be around 30 experiments took 1 iteration and around 70 experiments took 2 iterations.



(h) Summarize your conclusions with respect to accuracy and running time as a function of N and d.
Solution: After running all of these experiments with different data sets and different sizes of the data sets, it seems that the bigger data sets (increasing N) will give you a better approximation, but also cause it to take more time to run. Also the bigger d is, the greater the running time will become.