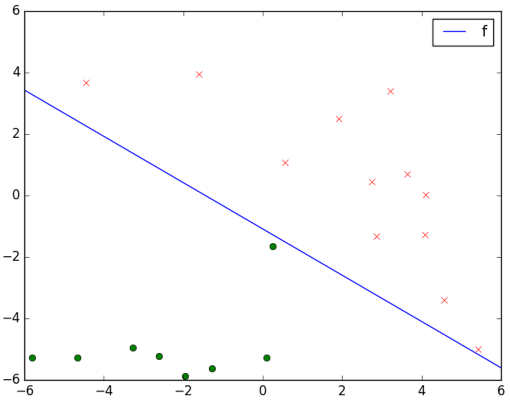
Homework 1. Machine Learning (MIIS)

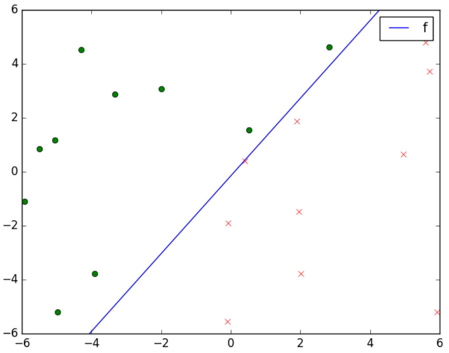
Marc Juvillà Garcia

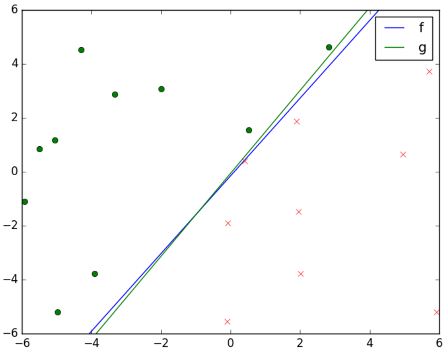
October 1, 2016

(a) Generate a dataset of size 20. Plot the examples $\{(\mathrm{x_n}\ ,\ \mathrm{y_n})\}$ as well as the target function f on a plane.



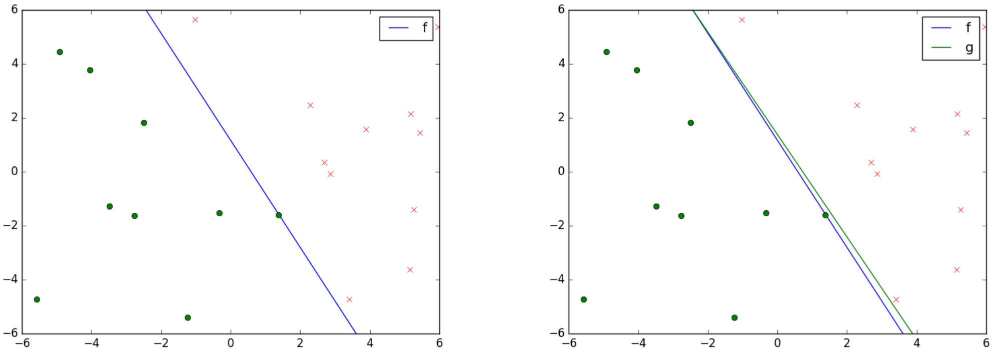
(b) Run the perceptron algorithm on the dataset. Report the number of updates that the algorithm takes before converging. Plot the examples $\{(\mathrm{x_n}\ ,\ \mathrm{y_n})\}$, the target function f, and the final hypothesis g in the same figure.





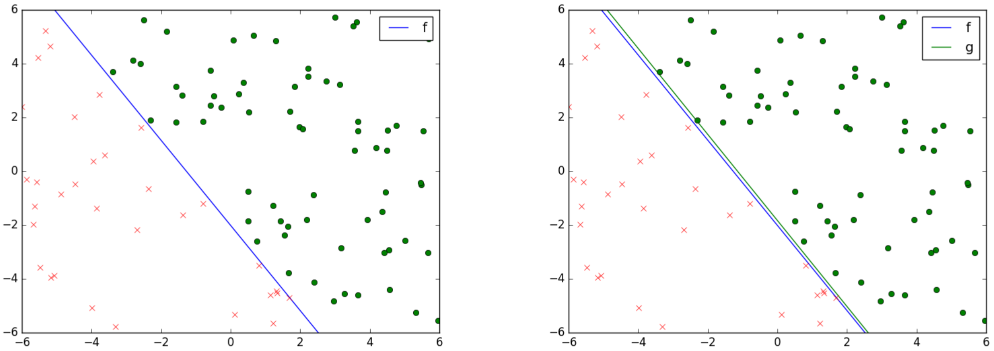
For the model shown in the figures, it took 12 updates of the model (that is, correcting 12 mistakes) and 0.292 ms before converging.

(c) Repeat everything in b) with another randomly generated dataset of size 20, and compare the result to b).



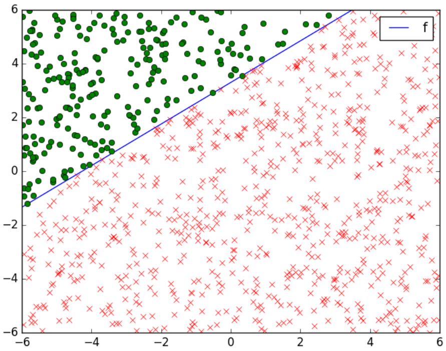
It took 16 updates of the model and 0.414 ms before converging, for the shown instance.
However, I noticed that these numbers can vary a lot so I computed the average time and training steps for 50 instances of the problem (different dataset and different *f*, but same number of data points and dimension of the space), which resulted in an average of 12.2 updates and 0.218 ms.

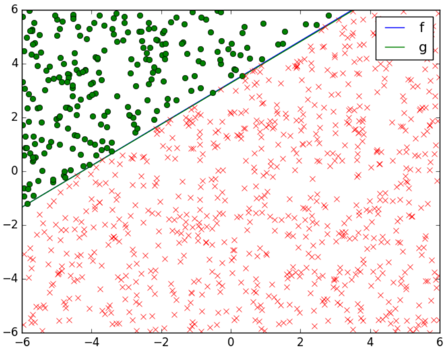
(d) Repeat everything in b) with another randomly generated dataset of size 100, and compare the result to b).



It took 78 updates of the model and 4.221 ms before converging.
Averaging for 50 instances of the problem, it resulted on 87.48 update steps and 2.841 ms of training time.

(e) Repeat everything in b) with another randomly generated dataset of size 1000, and compare the result to b).





It took 2176 updates of the model and 195.669 ms before converging.
Averaging for 50 instances of the problem, it resulted on 452.7 update steps and 50.815 ms of training time.

(f) Modify the experiment such that $x_n \in R^{10}$ instead of R^2 . Run the algorithm on a randomly generated dataset of size 1000. How many updates does the algorithm take to converge?

It took 6529 updates of the model and 979.125 ms before converging.
Averaging for 50 instances of the problem, it resulted on 5908.7 update steps and 737.511 ms of training time.

(g) Summarize your conclusions regarding the accuracy and running time of the algorithm as a function of N (the number of data points) and d (the number of dimensions).

	N = 20; d = 2	N = 100; d = 2	N = 1000; d = 2	N = 1000; d = 10
Steps	12.2	87.48	452.7	5908.7
Time (ms)	0.218	2.841	50.815	737.511