Mehryar Mohri
Introduction to Machine Learning
Courant Institute of Mathematical Sciences
Homework assignment 2
Solution (written by Andres Muñoz)

Perceptron Algorithm

Download the following data sets from the UC Irvine ML repository:

```
http://archive.ics.uci.edu/ml/datasets/Iris
http://archive.ics.uci.edu/ml/datasets/Connectionist+Bench+(Sonar,+Mines+vs.+Rocks)
http://archive.ics.uci.edu/ml/datasets/Spambase
```

The first two are linearly separable, the third one is not. For each data set, reserve the first half for training, the second half for testing.

1. Implement the Perceptron algorithm or use the Weka software library instead. Run it on the two separable data sets. How many updates were made by the algorithm? Compare with the upper bound known for the perceptron algorithm. What is the margin ρ_0 of the solution obtained?

   Solution: For the first dataset we separate Iris Versicolor into one category and join the other two in another. Since the sample is ordered by labels I rearrange them in a random way and divide into training and testing. The results of the perceptron for this case are :

   Number of updates: 4
   Training size:75
   Radius of training set:1.78
   Experimental margin:0.084
   Accuracy=1.0

   Using the experimental margin as the actual margin we see that the number of updates expected is 449. We can also use the information obtained to bound the actual margin of the training set, in this case:

   $$.084 < \rho < .89.$$

For the second data set we had to center the data and we got the following results
Number of updates:8841
Training size:104
Radius of training set:1.125
Experimental margin:0.0005
Accuracy=0.731

In this case the theoretical number of updates is bounded (using the experimental margin) by 5017600. Again we can invert this inequality to obtain a bound on the actual margin:

$$.0005 < \rho < .01\,.$$

2. As we saw in the midterm exam, when the training sample S is linearly separable with a maximum margin $\rho > 0$, there exists a modified version of the Perceptron algorithm that returns a solution with margin at least $\rho/2$ when run cyclically over S. Furthermore, that algorithm is guaranteed to converge after at most $16R^2/\rho^2$ updates, where R is the radius of the sphere containing the sample points.

   Fix $\epsilon \in (1/2, 1)$. Generalize that algorithm to guarantee that under the same conditions the solution has margin at least $\rho(1-\epsilon)$ (give the pseudocode). Adapting the proof given for the mid-term questions, show that the number of updates is upper bounded by $\frac{R^2/\rho^2}{(1-\epsilon)(\epsilon-1/2)}$.

   More generally, it can be proven that the same algorithm achieves a margin of at least $\rho(1-\epsilon)$ for any $\epsilon \in (0,1)$ with at most a polynomial number of updates in $O\big(\frac{R^2/\rho^2}{(1-\epsilon)}\big)$.

   Solution: The pseudocode of the algorithm is given below.
   We wish to prove that the number of mistakes made is less than $\frac{R^2/\rho^2}{(1-\epsilon)(\epsilon-1/2)}$.
   As seen in class, there exists a vector w such that for every training point:

$$\rho < \frac{y_t x_t \cdot w}{\|w\||}\,.$$

   Summing up these inequalities over all the points at which there was an

Modified Perceptron algorithm

```
$w1 \leftarrow 0$
for $t \leftarrow 1$ to $T$ do:
    RECEIVE($x_t$)
    RECEIVE($y_t$)
    if $(w_t = 0)$ or $(\frac{y_t w_t \cdot x_t}{\|w_t\|} < \rho(1-\epsilon))$ then:
        $w_{t+1} \leftarrow w_t + y_t x_t$
    else:
        $w_{t+1} \leftarrow w_t$
return $w_T$
```

update we obtain

$$\begin{aligned} M\rho &< \left| \sum \frac{y_t x_t \cdot w}{\|w\|} \right| \\ &= \left| \frac{w}{\|w\|} \cdot \sum w_{t+1} - w_t \right| \\ &\leq \|w_{T+1}\| \end{aligned}$$

we can then assume that $\|w_{T+1}\| > \frac{R^2}{(\epsilon - 1/2)}$ because otherwise the bound for M is obtained trivially.

Using the update rule we have that every time there is an update:

$$\begin{aligned} \|w_{t+1}\|^2 &= \|w_t + y_t x_t\|^2 \\ &= \|w_t\|^2 + 2 y_t x_t \cdot w_t + \|x_t\|^2 \\ &\leq \|w_t\|^2 + 2\|w_t\|\rho(1-\epsilon) + R^2 \\ &\leq (\|w_t\| + \rho(1-\epsilon))^2 + R^2 \,. \end{aligned}$$

From here, a straightforward manipulation shows that:

$$\|w_{t+1}\| \leq \|w_t\| + \rho(1-\epsilon) + \frac{R^2}{\|w_{t+1}\| + \|w_t\| + \rho(1-\epsilon)} \,.$$

If $\|w_{t+1}\| > \frac{R^2}{(\epsilon - 1/2)}$ or $\|w_t\| > \frac{R^2}{\rho(\epsilon - 1/2)}$ we can substitute the denominator in the previous inequality and find out that:

$$\|w_{t+1}\| \leq \|w_t\| + \frac{\rho}{2} < \|w_t\| + \rho\epsilon \,.$$

Since $\|w_1\| < R$ and $R > \rho(\epsilon - 1/2)$ it is clear that $\|w_1\| \leq \frac{R^2}{\rho(\epsilon-1/2)}$; thus there exists t_0 such that $\|w_{t_0}\| \leq \frac{R^2}{\rho(\epsilon-1/2)}$ and $\|w_{t_0+1}\| \geq \frac{R^2}{\rho(\epsilon-1/2)}$, so using the inequality above recursively we can conclude:

$$\|w_{T+1}\| \leq \|w_{t_0}\| + M\rho\epsilon \leq \frac{R^2}{\rho(\epsilon - 1/2)} + M\rho\epsilon\,.$$

Finally using the lower bound for $\|w_{T+1}\|$ and solving for M we have:

$$M \leq \frac{R^2}{\rho^2(1-\epsilon)(\epsilon - 1/2)}\,.$$

3. Implement the algorithm of the previous question (or modify Weka's perceptron code). Use $\rho = \rho_0$ and run the algorithm with $\epsilon = 1/4$ on the same datas set as the first question. How many updates are made by the algorithm? What is the margin of the solution obtained? Compare with ρ_0.

   Solution: Using the experimental margin as the actual margin we got the following results

   Iris data set

   Number of updates:6
   Training size:75
   Radius of training set:1.90
   Experimental margin:0.112
   Accuracy=1.0

   Observe that the margin had a small improvement.

   Sonar data set

   Number of updates:3909
   Training size:104
   Radius of training set:1.034
   Experimental margin:0.0019
   Accuracy=0.7307

   Note that even though the margin improved, the accuracy did not.

4. Run the perceptron algorithm on the third data set and stop it after n passes over the training data with $n = 10, 50, 100$. Report the test error of the solution obtained in each case.

   Solution: After centering and training on 50, 100 and 150 examples the following results were obtained for the spam database:
   Accuracy for 50=0.78
   Accuracy for 100=0.77
   Accuracy for 150=0.79

   The differences are small, so the results probably will keep oscillating around these numbers.