CS 5350/6350: Machine Learining Fall 2017

Yucheng Yang
Homework 2

Handed out: 12 September, 2017
Due date: 26 September, 2017

1 Warm up: Linear Classifiers and Boolean Functions

Solution:

1. $y = 1 \quad \text{if} \quad x_1 - x_2 + x_3 \geq 1$ So

$$w = [1, -1, 1], \quad b = -1$$

2. $y = 1 \quad \text{if} \quad x_1 + 2 * x_2 + x_3 \geq 2$ So

$$w = [1, 2, 1], \quad b = -2$$

3. $y = 1 \quad \text{if} \quad x_1 - 2 * x_2 + 3 * x_3 \geq 1$ So

$$w = [1, -2, 3], \quad b = -1$$

4. This boolean functions is not linearly separable.

5. $y = 1 \quad \text{if} \quad -x_1 + x_2 - x_3 \geq 3$ So

$$w = [-1, 1, -1], \quad b = -3$$

2 Mistake Bound Model of Learning

Solution:

1. 80.
The l is restricted to be an integer value, thus l has 80 different value, and the hypothesis functions rely on l.

2.

$$f_l(x_1^t, x_2^t) \neq y^t$$

3. if there is an error in f_l, then remove this f_l from C.
   if
   $$y^t = 1 \quad \text{and} \quad f_l(x_1^t, x_2^t) \neq y^t$$
   , so remove all l that are smaller than the largest absolute values of x_1^t and x_2^t.
   if
   $$y^t = -1 \quad \text{and} \quad f_l(x_1^t, x_2^t) \neq y^t$$
   , so remove all l that are larger than the largest absolute values of x_1^t and x_2^t.

4. The maximum number of mistakes is $80 - 1 = 79$.
   The Pseudocode is below:

   (a) Initialize $C_0 = C$, the set of all possible functions

   (b) When an example i arrives:
   Randomly choose a function f_l that is still in C_i
   if $y_i \neq f_l(x_1^i, x_2^i)$:
   Update $C_i = C_i$ removes f_l that does not agree with y_i.
   Wait a new example.

   (c) Learning ends when there is only one function in Ci.

Solution:
If there are M perfect experts, then when there are less or equal $2M - 1$ elements left, the Halving algorithm will always make correct predictions. So the mistake bound will be

$$\log N - \log(2M - 1) = log(\frac{N}{2M - 1}) = log(\frac{N}{M})$$

3 The Perceptron Algorithm and its Variants

Solution:

1. MATLAB is the language to implement this experiment. First, I build a readTxt funciton to read the data and reform to set the label to column 71, and full the 67 feature in data table, all other grids set as 0. Second, I build all 4 kinds of perceptron functions. Third, I make a cross-valid script to run all of these 4 function to find out the best hyper-parameter. Next, I create all 4 kinds of perceptron functions again which will return a list of weight, bias and update time for each epoch. Finally, I create a develop and test script to find the best weight, and bias from 20 epochs based on develop dataset, then run the weight and bias in test data base, in the end plot the picture of 4 function's accuracy in each epoch.

2. Majority baseline accuracy on test set is 0.5731.
   Majority baseline accuracy on development set is 0.5492.

3.
   Simple Perceptron:

(a) The best learning rate is $\eta = 0.1$ from table.

(b)

Learning Rate	train00	train01	train02	train03	train04	mean Correct Rate
1	0.9125	0.9349	0.9276	0.9258	0.8311	0.9064
0.1	0.9168	0.9355	0.9343	0.9343	0.9258	0.9293
0.01	0.9138	0.9343	0.8643	0.9288	0.9294	0.9141

(c) The highest accuracy epoch is number 19 , and the weight has 13383 updates at this point. The end of 20 epochs the weight has 14046 updates.

(d) The end of 20 epochs the accuracy is 0.922575976845152, but the 19 epoch has the higher accuracy 0.924746743849494.

(e) The accuracy of test is 0.927641099855282.

(f)

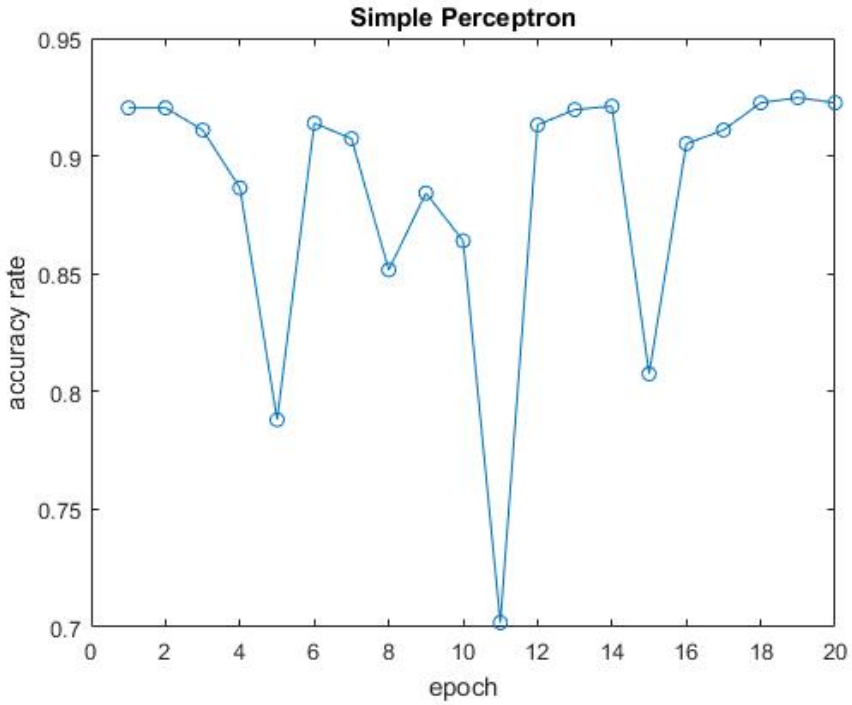

Figure 1: Simple Perceptron with $\eta_0 = 0.1$

Perceptron with dynamic learning rate (Update learning rate on each example)

(a) The best learning rate is $\eta = 1$ from table.

(b)

Learning Rate	train00	train01	train02	train03	train04	meanCorrectRate
1	[0.85947]	[0.91074]	[0.89505]	[0.87274]	[0.86671]	[0.88094]
0.1	[0.87214]	[0.88239]	[0.89324]	[0.84741]	[0.85826]	[0.87069]
0.01	[0.72738]	[0.57117]	[0.65742]	[0.73402]	[0.68456]	[0.67491]

(c) The highest accuracy epoch is number 20 , and the weight has 26052 updates at this point.

(d) The end of 20 epochs the accuracy is 0.829956584659913 which is the highest in the set of epoch.

(e) The accuracy of test is 0.863241678726483.

(f)

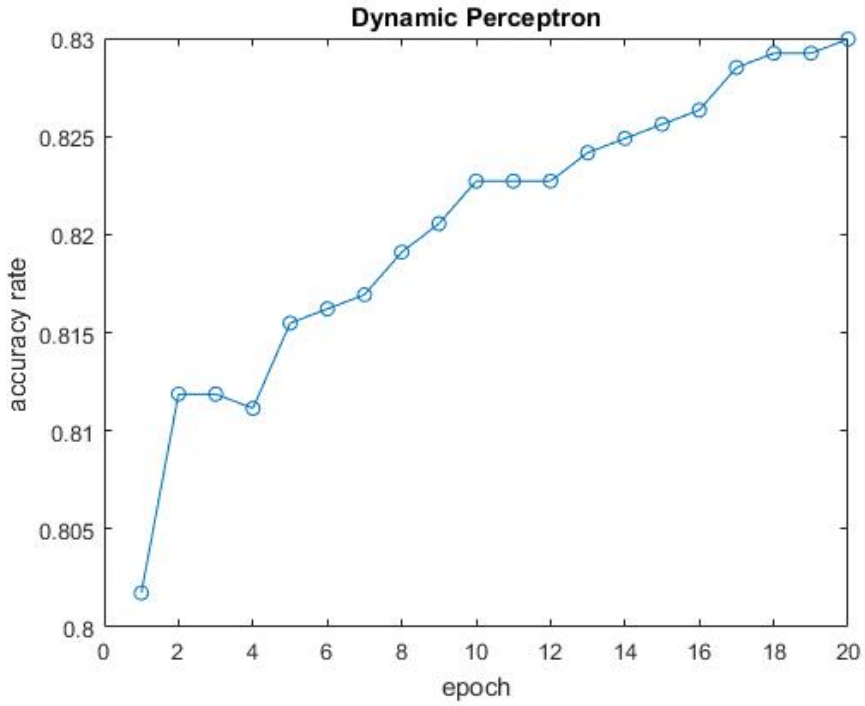

Figure 2: Dynamic Perceptron with $\eta_0 = 1$

Margin Perceptron: (Update learning rate on each example)

(a) The best learning rate is $\eta = 0.1$, and the best margin is 0.01 from table.

(b)

Learning Rate, Margin	train00	train01	train02	train03	train04	meanCorrectRate
1 And 1	[0.54343]	[0.56212]	[0.55489]	[0.55609]	[0.56152]	[0.55561]
1 And 0.1	[0.91194]	[0.88842]	[0.91315]	[0.92521]	[0.86852]	[0.90145]
1 And 0.01	[0.84861]	[0.89324]	[0.89445]	[0.83172]	[0.86369]	[0.86634]
0.1 And 1	[0.54343]	[0.56212]	[0.55489]	[0.55609]	[0.4427]	[0.53185]
0.1 And 0.1	[0.54885]	[0.56212]	[0.55489]	[0.55609]	[0.57177]	[0.55875]
0.1 And 0.01	[0.89747]	[0.88842]	[0.91134]	[0.9035]	[0.91797]	[0.90374]
0.01 And 1	[0.538]	[0.56212]	[0.56092]	[0.5778]	[0.56152]	[0.56007]
0.01 And 0.1	[0.54343]	[0.56212]	[0.55489]	[0.55609]	[0.56152]	[0.55561]
0.01 And 0.01	[0.67129]	[0.64294]	[0.7117]	[0.7117]	[0.71834]	[0.69119]

(c) The highest accuracy epoch is number 18 , and the weight has 55744 updates at this point. The end of 20 epochs the weight has 61631 updates.

(d) The end of 20 epochs the accuracy is 0.897250361794501, but the 18 epoch has the same accuracy. as the learning rate is too small to impact the weight vector and bias.

(e) The accuracy of test is 0.916063675832127.

(f)

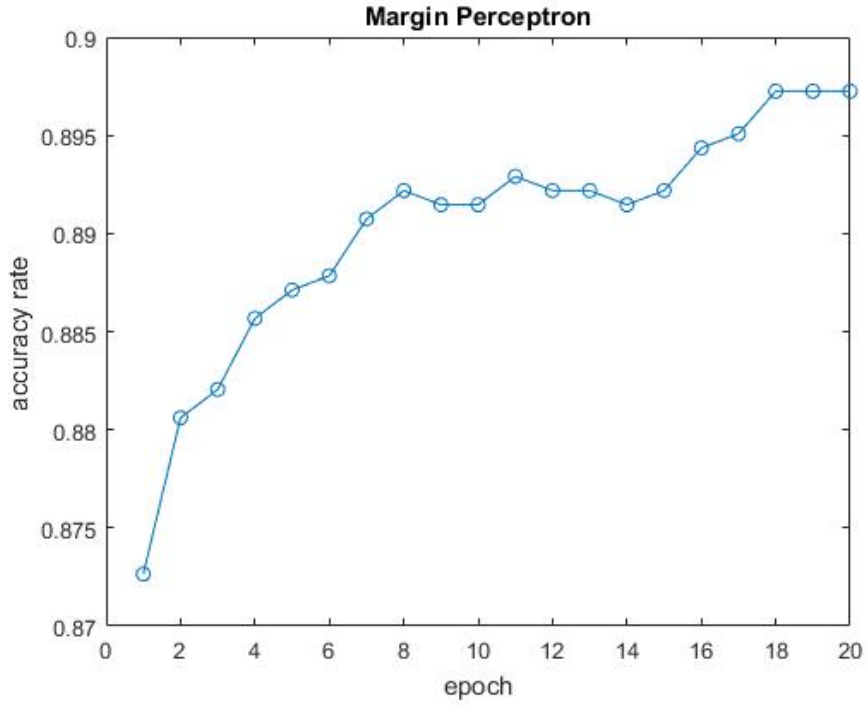

Figure 3: Margin Perceptron with $\eta_0 = 0.1$ and $\mu = 0.01$

Averaged Perceptron:

(a) the best learning rate is $\eta = 0.01$ from table, even though all learning rate perform great.

(b)

Learning Rate	train00	train01	train02	train03	train04	mean Correct Rate
1	[0.91797]	[0.94632]	[0.94632]	[0.94029]	[0.93124]	[0.93643]
0.1	[0.91677]	[0.94511]	[0.95054]	[0.93908]	[0.93124]	[0.93655]
0.01	[0.91556]	[0.94813]	[0.94934]	[0.93969]	[0.93064]	[0.93667]

(c) 14084 updates

(d) The highest accuracy epoch is number 13, and the weight has 9208 updates at this point. The end of 20 epochs the weight has 14084 updates.

(e) The end of 20 epochs the accuracy is 0.924023154848046, but the 13 epoch has the higher accuracy 0.924746743849494.

(f) The accuracy of test is 0.930535455861071.

(g)

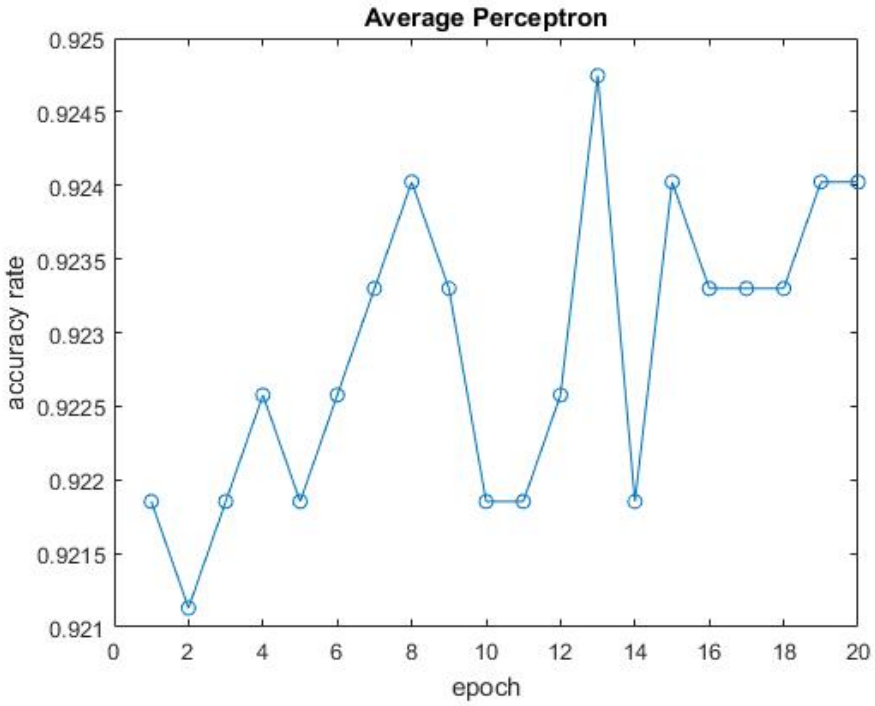

Figure 4: Average Perceptron with $\eta_0 = 0.01$