Machine Learning homework 2

Bodong Zhang u0949206

1 w=(-1,-1), ϕ(x1,x2)=($4x_1^4$, $16x_2^4$), b=-r, if $w^T \phi \geq b$, then it means $0 \geq -4x_1^4 - 16x_2^4 \geq -r$, $4x_1^4 + 16x_2^4 \leq r$, $f(x^1, x^2) = +1$

2.1 The |C| mainly depends on range of r. $1 \leq r \leq 80$, so if r is integer, then |C|=80.

2.2 If $(x_1^{t2} + x_2^{t2} - r^2)\, y^t > 0$, then the current hypothesis is correct, otherwise there is mistake.

2.3 For positive examples, if $x_1^{t2} + x_2^{t2} > r^2$ and $y^t = +1$, then there is an error, which means r is too small, r should be increased to square root of $x_1^{t2} + x_2^{t2}$.

For negative examples, if $x_1^{t2} + x_2^{t2} \leq r^2$ and $y^t = -1$, then there is an error, which means r is too large, r should be decreased to square root of $x_1^{t2} + x_2^{t2} - \varepsilon$($\varepsilon$ is a small positive number)

2.4

1 Initialize r=80 and ε=1

2 Scan the whole training example, t=1…T

if $x_1^{t2} + x_2^{t2} > r^2$ and $y^t = +1$, update $r = \sqrt{x_1^{t2} + x_2^{t2}}$

if $x_1^{t2} + x_2^{t2} \leq r^2$ and $y^t = -1$, update $r = \sqrt{x_1^{t2} + x_2^{t2}} - \varepsilon$

3 return r

The worst case is that r should be updated gradually to reach correct answer. Assume r should be 1, so the update decrease r by step 1, 80→79→78……→1, so the maximum mistake is 79.

2.5 a The set of hypothesis should have all possible values r could have. The initial set is the whole set from 1 to 80. By learning from mistake, we could know whether real r should be larger or smaller than a number, so we could use two integers to record minimum bound and maximum bound.

2.5 b Assume we have minimum bound a, and maximum bound b, then if $x_1^{t2} + x_2^{t2} > (a/2+b/2)^2$ and $y^t = +1$, then the majority of r set would give wrong assumption, so there is an error. If $x_1^{t2} + x_2^{t2} \leq (a/2+b/2)^2$ and $y^t = -1$, then the majority of r set would give wrong assumption, so there is an error.

2.5 c

1 First initialize a=1, b=80

2 For each training example,

if $x_1^{t2} + x_2^{t2} > (a/2+b/2)^2$ and $y^t = +1$, then $a \leftarrow \sqrt{x_1^{t2} + x_2^{t2}}$

if $x_1^{t2}+x_2^{t2} \leq (a/2+b/2)^2$ and $y^t=-1$, then $b \leftarrow \sqrt{x_1^{t2}+x_2^{t2}}$

Then until there is only one element in set

3 Return final weight vector

3.3 Experiments

1 For each training example (xi , yi), if sgn($w_t^T x_i$)≠y', then $w_{t+1}<w_t+r(y_i x_i)$, so after running the code(T3_3_1.java), the weight after one pass is [0 1 0 -1 2 0], one mistake was made.

(The following solutions are based on random initialization produced by program, so exact same result may not be got by running same program again!)

2a First learning rate should be tuned, the accuracies on testing set by different learning rate is below.

learning rate	accuracies on testing		
0.0001	0.23987	0.23987	0.24018
0.001	0.75125	0.70864	0.73855
0.01	0.73014	0.79102	0.76938
0.1	0.7942	0.79186	0.81145
0.5	0.78448	0.78953	0.75943
1	0.80781	0.82055	0.79821
10	0.77381	0.773817	0.77381
100	0.7738	0.77381	0.77381

According to experiment, set rate=1. First initialize weight with random numbers between 0 and 1, the number of updates made by simple perceptron algorithm is 1385. The accuracy of final weight vector on training data is 0.80480, the accuracy of final weight vector on testing data is 0.79668.(T3_3_2_Bodong.java)

2b

Set margin to different numbers and compare results.

margin= 0.5

Totally 1471 updates in simple Perceptron Algorithm

Accuracy in training set is 0.8228874337386967

Accuracy in testing set is 0.8169579684093777

margin= 1.0

Totally 1472 updates in simple Perceptron Algorithm

Accuracy in training set is 0.8083879014655442

Accuracy in testing set is 0.8032661490802004

margin= 1.5

Totally 1528 updates in simple Perceptron Algorithm

Accuracy in training set is 0.8143124415341441

Accuracy in testing set is 0.8176463839063755

margin= 2.0

Totally 1585 updates in simple Perceptron Algorithm

Accuracy in training set is 0.8029310882444652

Accuracy in testing set is 0.7974911079664971

margin= 2.5

Totally 1618 updates in simple Perceptron Algorithm

Accuracy in training set is 0.8295915185531649

Accuracy in testing set is 0.8254101809002945

margin= 3.0

Totally 1642 updates in simple Perceptron Algorithm

Accuracy in training set is 0.8327096975366386

Accuracy in testing set is 0.8267870118942899

margin= 3.5

Totally 1685 updates in simple Perceptron Algorithm

Accuracy in training set is 0.8369192391643281

Accuracy in testing set is 0.8378399051516426

margin= 4.0

Totally 1682 updates in simple Perceptron Algorithm

Accuracy in training set is 0.8297474275023387

Accuracy in testing set is 0.8234979156308563

margin= 4.5

Totally 1734 updates in simple Perceptron Algorithm

Accuracy in training set is 0.7968506392266916

Accuracy in testing set is 0.7906451983019085

margin= 5.0

Totally 1756 updates in simple Perceptron Algorithm

Accuracy in training set is 0.8322419706891175

Accuracy in testing set is 0.8292347114391708

(T3_3_2_margin_perceptron_Bodong.java)

So generally, with the increase of margin threshold, the performance is the best.

3 By performing three passes, the number of updates made by margin perceptron algorithm is 4036. The accuracy of final weight vector on training data is 0.79404, the accuracy of final weight vector on testing data is 0.78353.(T3_3_3_multiple_passes_Bodong.java)

By performing five passes, the number of updates made by margin perceptron algorithm is 6845. The accuracy of final weight vector on training data is 0.81197, the accuracy of final weight vector on testing data is 0.81561.(T3_3_3_multiple_passes_Bodong.java)

With the increase of iteration steps, the accuracy becomes better. Shuffle helps to avoid periodic vibration.

4 By following the aggressive update, finally the number of updates **with shuffle** is 5215, the accuracy of final weight vector on training data is 0.81867, the accuracy of final weight vector on testing data is 0.81378.(T3_3_4_aggressioin_perceptron_Bodong.java)

The number of updates **without shuffle** is 5331, the accuracy of final weight vector on training data is 0.81478, the accuracy of final weight vector on testing data is 0.80808.(T3_3_4_aggressioin_perceptron_Bodong.java)