ΑΝΑΓΝΩΡΙΣΗ ΠΡΟΤΥΠΩΝ

ΠΡΟΑΙΡΕΤΙΚΗ ΕΡΓΑΣΙΑ 2

Σταματόπουλος Βασίλειος – 1115201400188

1.

Στον παρακάτω πίνακα περιγράφονται τα αποτελέσματα που δόθηκαν από τις διάφορες εκτελέσεις του αλγορίθμου perceptron, πρώτα για **p = 0.01**.

X	**Συντελεστές**	**Iterations**	**Misclassified Data**
[3 3]’	[0.3 0.26 -1.1]’	255	0
[2 2]’	[0.3 0.26 -1.1]’	255	0
[0 2]’	[0.04 0.49 -1.02]’	811	0
[1 1]’	[0.014 0.014 -0.03]	20000 = MAX	36

Όπως φαίνεται, στις δύο πρώτες περιπτώσεις στις οποίες οι κλάσεις βρίσκονται μακριά η μία από την άλλη και είναι ευκόλως διαχωρίσιμες, ο αλγόριθμος τελείωσε γρήγορα και χωρίς λάθη. Στην τρίτη περίπτωση, όπου οι κλάσεις ήρθαν πιο κοντά η μία στην άλλη, ο αλγόριθμος χρειάστηκε λίγο περισσότερη ώρα για να τελειώσει. Τέλος, στην τελευταία περίπτωση όπου οι κλάσεις επικαλύπτονται, ο αλγόριθμος δεν σύγκλινε (δηλαδή χρειάστηκε και τις 20000 iterations για να τελειώσει) ενώ επιπλέον δεν κατάφερε να ταξινομήσει σωστά 36 στοιχεία. Αυτό είναι λογικό καθώς το perceptron είναι γραμμικός διαχωριστής και στην περίπτωση επικάλυψης τα δεδομένα δεν διαχωρίζονται γραμμικά.

Για p = 0.05

X	**Συντελεστές**	**Iterations**	**Misclassified Data**
[3 3]’	[0.3 0.26 -1.1]’	10	0
[2 2]’	[0.3 0.26 -1.1]’	10	0
[0 2]’	[0.036 0.502 -1.057]’	34	0
[1 1]’	[0.28 0.27 -0.9]	20000 = MAX	23

Στην περίπτωση του learning rate = 0.05, για τις μη επικαλυπτόμενες κλάσεις ο αλγόριθμος τελείωσε πολύ πιο γρήγορα από το 0.01. Για την τελευταία περίπτωση των επικαλυπτόμενων κλάσεων, χρειάστηκαν πάλι 20000 επαναλήψεις (δεν σύγκλινε), αλλά είχε μικρότερο σφάλμα. Επιπλέον, λόγω του μεγαλύτερου δείκτη μάθησης οι συντελεστές που βρήκε ήταν μεγαλύτεροι.

2.

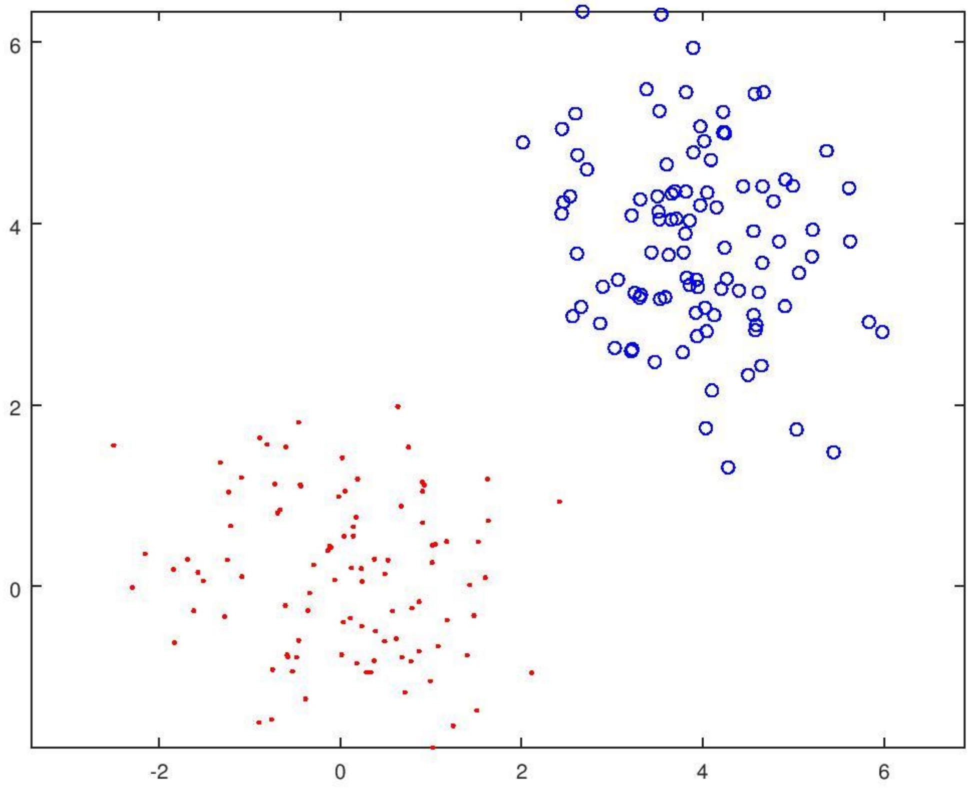
Στον παρακάτω πίνακα δίνονται τα αποτελέσματα των εκτελέσεων του αλγορίθμου.

N	**Bayes Error**	**SSE Error**
200	0.11	0.12
10000	0.1488	0.152
100000	0.14714	0.15072

Γενικότερα, για τις τρεις περιπτώσεις ο ταξινομητής Bayes έδωσε καλύτερα αποτελέσματα από την απλή πρόσθεση τετραγώνων, κι΄ αυτό γιατί πρόκειται για έναν πολυπλοκότερο αλγόριθμο. Οι γραμμικοί διαχωριστές προσφέρουν ταχύτητα στην ταξινόμηση θυσιάζοντας την ακρίβεια των αποτελεσμάτων.

3.

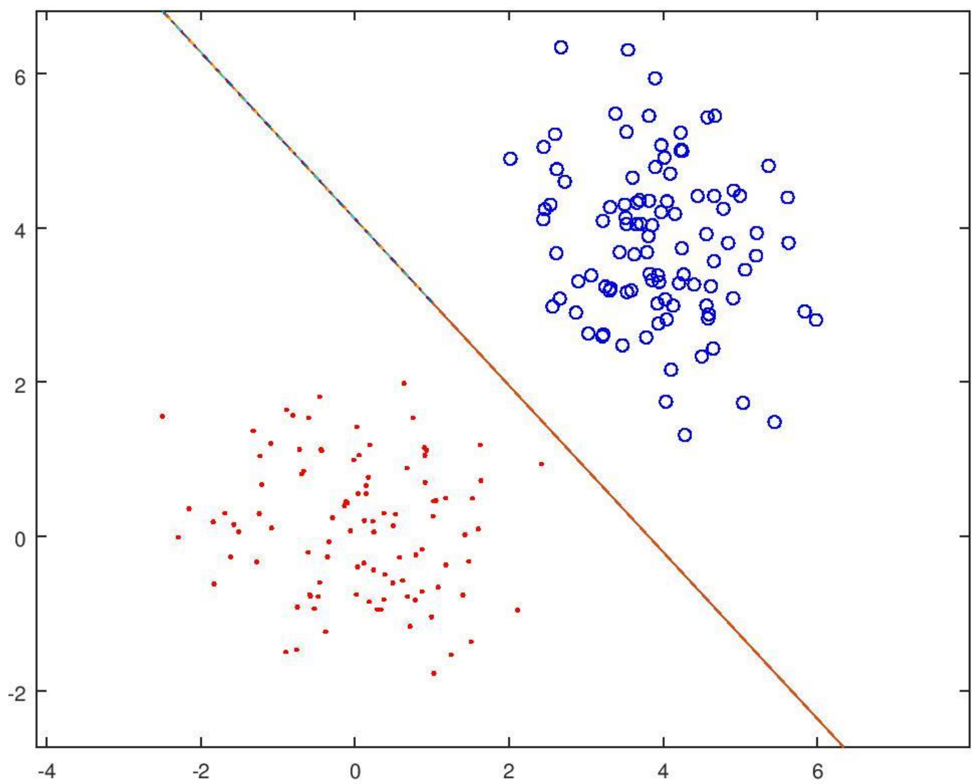
Αρχικά αναπαριστούμε γραφικά τα δεδομένα του συνόλου X1:



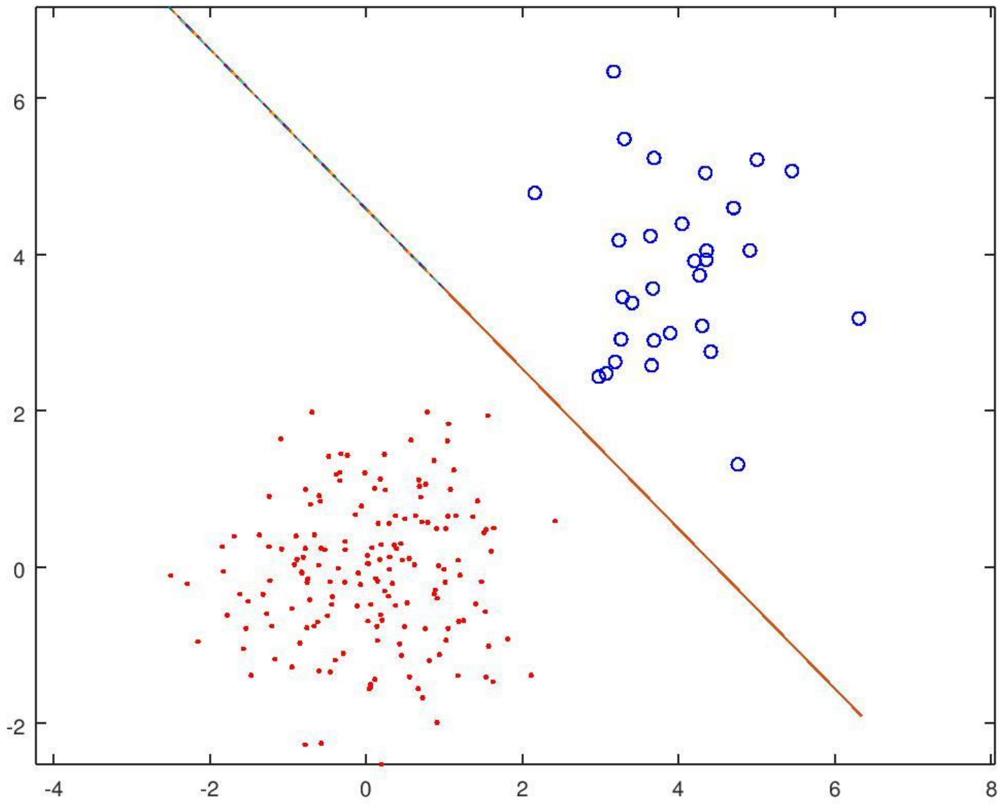
Δημιουργούμε όπως και στο ερώτημα 2, τον ταξινομητή αθροίσματος των ελαχίστων τετραγώνων και παίρνουμε πολύ μικρό σφάλμα ίσο με $1.0000e - 004$. Με τη χρήση του παρακάτω κώδικα, παίρνουμε τον παρακάτω διαχωριστή.

```
% Compute the classification error of the LS classifier based on X2
[w]=SSErr(X1,y1,0)
SSE_out=2*(w'*X2>0)-1;
err_SSE=sum(SSE_out.*y2<0)/sum(N2)
figure(1), plot(X1(1,y1==1),X1(2,y1==1),'bo',...
X1(1,y1==-1),X1(2,y1==-1),'r.')
hold on;
figure(1), axis equal

xp = linspace(min(X1), max(X1), 100);
yp = - (w(1)*xp + w(3))/w(2);
plot(xp, yp);
hold off;
```



B) Σε αυτή τη περίπτωση προκύπτει ο παρακάτω διαχωριστής



Το σφάλμα τώρα είναι μηδενικό.

4.

Στο ερώτημα αυτό θα συλλέξουμε τα μετρικά του αλγορίθμου για διαφορετικές τιμές του c.

C	**Pe. train**	**Pe. test**	**# Supp. Vectors**	**Margin**
0.1	0.025	0.015	79	0.97644
0.2	0.0175	0.01	61	0.82255
0.5	0.0175	0.01	44	0.61152
1	0.0175	0.0075	37	0.51728
2	0.015	0.0075	30	0.42102
20	0.015	0.01	19	0.27172

Τα αποτελέσματα αυτά είναι αρκετά λογικά καθώς για μεγαλύτερες τιμές του c, το margin είναι μικρότερο και συνεπώς το σφάλμα ταξινόμησης είναι μικρότερο. Γενικότερα δεν πρέπει το margin να είναι πολύ μεγάλο (πολλά σφάλματα), αλλά ούτε πολύ μικρό (κακή γενίκευση).

5.

Δεν μπόρεσα να τρέξω τα αρχεία.

6.

Α) Ο γραμμικός SVM ταξινομητής είχε σημαντικά μεγάλο ποσοστό σφάλματος (44%). Αυτό συνέβη γιατί τα δεδομένα του συνόλου δεν είναι γραμμικώς διαχωρίσιμα.

Β) Με την εκτέλεση του μη-γραμμικού SVM για C = [0.2 2 20 200 2000 20000] και σ = [1 1.5 2.5], εξάγουμε τα παρακάτω αποτελέσματα.

C	Pe_train	Pe_test	Sup_vec	Σύγκλιση
0.2	0.37407	0.42593	267	ΟΧΙ
2	0	0.085185	226	ΝΑΙ
20	0	0.081481	218	ΝΑΙ
200	0	0.081481	218	ΝΑΙ
2000	0	0.081481	218	ΝΑΙ
20000	0	0.081481	218	ΝΑΙ

Φαίνεται πως για πολύ χαμηλό C, ο αλγόριθμος δίνει και κακά αποτελέσματα αλλά και αποτυγχάνει να συγκλίνει. Όμως, καθώς το αυξάνουμε η απόδοση του αλγορίθμου βελτιώνεται εκθετικά. Τέλος, για τα τρία τελευταία C δεν υπάρχει κάποια βελτίωση, ενώ τα διανύσματα υποστήριξης κυμαίνονται στα ίδια μεγέθη.

Γ) Για τον πολυωνυμικό πυρήνα λαμβάνουμε τα παρακάτω αποτελέσματα

C	Pe_train	Pe_test	Sup_vec	Σύγκλιση
0.2	0.52593	0.51481	10	ΟΧΙ
2	0.44444	0.44444	265	ΝΑΙ
20	0.44444	0.44444	256	ΝΑΙ
200	0.44444	0.44444	267	ΝΑΙ
2000	0. 47778	0.47407	185	ΟΧΙ
20000	0.51111	0.50741	42	ΟΧΙ

Ο χρόνος που πήρε ο αλγόριθμος με τη χρήση του polyonomial kernel ήταν αρκετά μεγαλύτερος από αυτόν του RBF. Γενικότερα, η απόδοση του πυρήνα αυτού δεν ήταν καθόλου καλή, με τα καλύτερα αποτελέσματα στα C = 2,20,200.