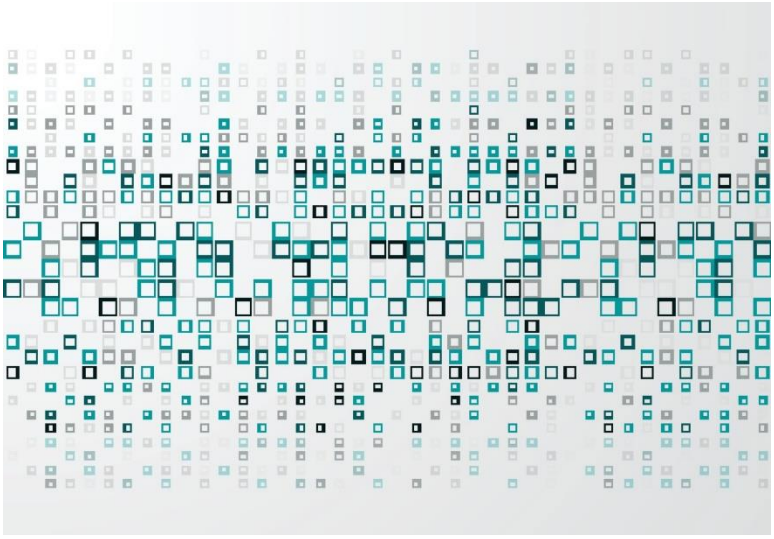


ΕΡΓΑΣΙΑ 5

IRIS DATA SET ΜΕ ΓΡΑΜΜΙΚΟΥΣ ΤΑΞΙΝΟΜΗΤΕΣ

ΚΑΡΑΠΕΠΕΡΑ ΕΛΠΙΔΑ | 57423 | ΑΝΑΓΝΩΡΙΣΗ ΠΡΟΤΥΠΩΝ

Πολλές από τις συναρτήσεις που χρησιμοποιούνται έχουν ίδιο όνομα με τα αρχεία που παρουσιάστηκαν στις διαφάνειες του μαθήματος. Ωστόσο έχουν μικρές διαφορές (μερικά plot γίνονται και μερικά errors επιστρέφονται) οπότε δε μπορούν να χρησιμοποιηθούν οι συναρτήσεις όπως ακριβώς είναι στις διαφάνειες για να τρέξει το πρόγραμμα.

A

Αρχικά, να οριστεί ο ορισμός της συνάρτησης κριτηρίου:
Είναι μια βαθμωτή συνάρτηση των βαρών $(\omega_1, \dots, \omega_n)$, την οποία προσπαθούμε να ελαχιστοποιήσουμε. Για να επιτύχουμε την ελαχιστοποίησή της (η οποία είναι δύσκολο να επιτευχθεί) χρησιμοποιούμε εναλλακτικά κριτήρια και επαναληπτικές μεθόδους βελτιστοποίησης (gradient decent). Με τις μεθόδους αυτές επιτυγχάνεται διαχωρισμός των στοιχείων με γραμμικές συναρτήσεις. Το διάνυσμα των βαρών ω καθορίζει τον προσανατολισμό του υπερεπιπέδου απόφασης.

Η διαδικασία που ακολουθείται είναι η εξής:

Επιλέγουμε ένα αρχικό σημείο α_1 και υπολογίζουμε την τιμή της συνάρτησης κριτηρίου καθώς και την κλίση της. Στη συνέχεια, κινούμενοι στην κατεύθυνση αρνητικής κλίσης κατά μία ποσότητα η(κ) (learning rate) και επαναλαμβάνουμε την διαδικασία για το επόμενο σημείο α_2.

- PERCEPTRON:

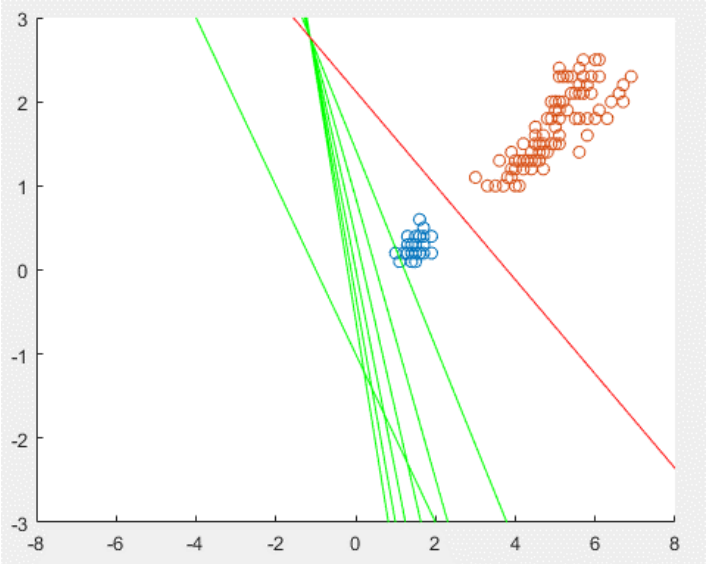
  Ο αλγόριθμος Perceptron χρησιμοποιεί ως συνάρτηση κριτηρίου την $J_p(a) = \sum_{y \in Y}(-a^t y)$, όπου Υ(α) είναι το σύνολο των δειγμάτων που δεν έχουν ταξινομηθεί σωστά από το α.
  Αν το Υ(α) είναι κενό, τότε $J_p(a) = 0$.
  Όταν το y δεν είναι σωστά ταξινομημένο θα ισχύει $a^t y < 0$, οπότε η συνάρτηση κριτηρίου δεν είναι ποτέ αρνητική και μηδενίζεται όταν το α είναι το διάνυσμα λύσης.

Αλγόριθμος 3. Batch Perceptron

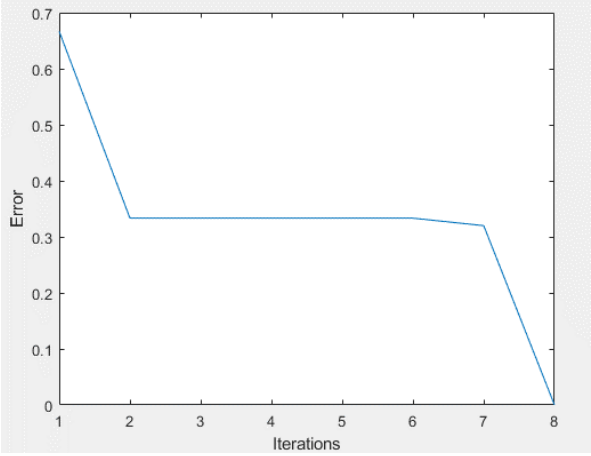
1 **begin initialize** a, κριτήριο θ, η(0)>0, k=0
2 **do** k ← k+1
3 $\mathbf{a} \leftarrow \mathbf{a} + \eta(k) \sum_{y \in Y_k} \mathbf{y}$
4 **until** $\left|\eta(k) \sum_{y \in Y_k} \mathbf{y}\right| < \theta$
5 **return** a
6 **end**

Η συνάρτηση κριτηρίου είναι ανάλογη του αθροίσματος των αποστάσεων των λάθος ταξινομημένων δειγμάτων από το σύνορο απόφασης.
Το διάνυσμα των κλίσεων θα είναι $\nabla J_p(a) = \sum_{y \in Y}(-y)$ και η αναδρομική σχέση για να βρεθεί το επόμενο α θα είναι $a(k+1) = a(k) + \eta(k) \sum_{y \in Y_k} y$.
Μία σύντομη περιγραφή του αλγορίθμου φαίνεται παρακάτω:

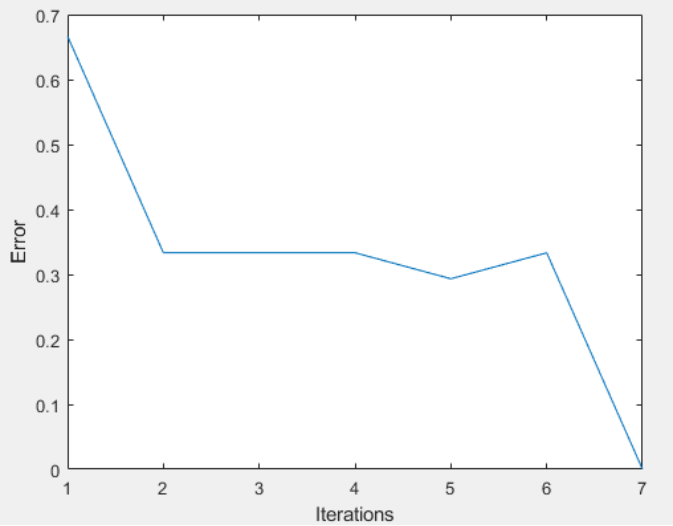
Η μέθοδος αυτή εφαρμόστηκε και για 2 και για 4 χαρακτηριστικά.
Αρχικά παρουσιάζεται η ταξινόμηση για 2 χαρακτηριστικά, ώστε και παρουσιαστεί και σχηματικά σε 2D:



Στο παραπάνω σχήμα φαίνεται η εξέλιξη της ευθείας που χωρίζει τα δεδομένα με την πάροδο των epochs, η οποία, όπως φαίνεται, στην τελευταία εποχή (8η) χωρίζει επιτυχώς όλα τα σημεία.
Ο τελικός γραμμικός ταξινομητής είναι ω = [199, -52.5, -93.7].
Η εξέλιξη του error φαίνεται την επόμενη εικόνα:



Η ίδια διαδικασία εφαρμόστηκε στη συνέχεια και για τα 4 χαρακτηριστικά των λουλουδιών. Αυτή τη φορά τα epochs (iterations) που χρειάστηκαν για τον επιτυχή διαχωρισμό των λουλουδιών ήταν 7.
Ο τελικός γραμμικός ταξινομητής είναι ω = [57, 114.5, 275.1, -381.6, -175].



Το γεγονός ότι χρειάστηκαν λιγότερες επαναλήψεις για το διαχωρισμό των δεομένων αυτή τη φορά είναι πολύ λογικό, καθώς έχουμε πλέον περισσότερα δεδομένα και περισσότερες διαστάσεις στις οποίες μπορούμε να χωρίσουμε τα στοιχεία.

- BATCH RELAXATION WITH MARGIN:

Ο αλγόριθμος Batch relaxation with margin χρησιμοποιεί ως συνάρτηση κριτηρίου την

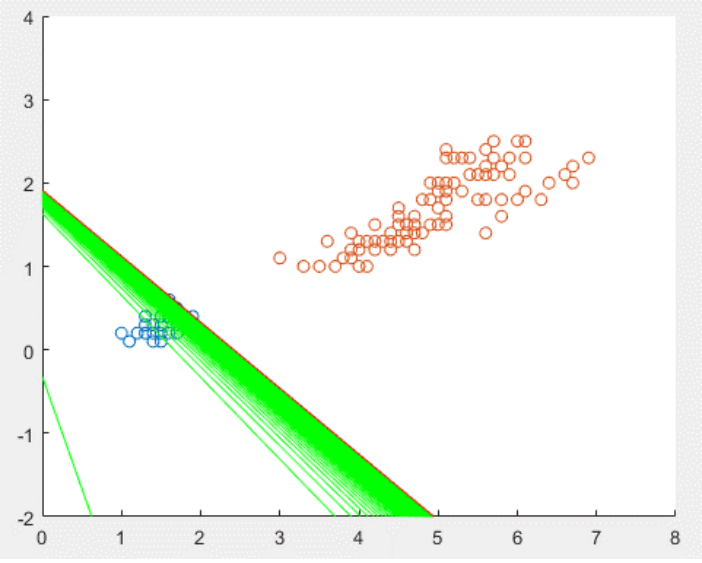
$J_r(a) = \frac{1}{2}\sum_{y\in Y} \frac{(a^t y - b)^2}{||y||^2}$, όπου Υ(α) είναι το σύνολο των δειγμάτων για τα οποία ισχύει $a^t y < b$.
Αν το Υ(α) είναι κενό, τότε $J_r(a) = 0$. Η $J_r(a)$ δεν είναι ποτέ <ο και μηδενίζεται αν και μόνο αν $a^t y > b$ για όλα τα δείγματα εκπαίδευσης.
Το διάνυσμα κλίσεων είναι $\nabla J_r(a) = \sum_{y\in Y} \frac{a^t - b}{||y||^2} y$ και η αναδρομική σχέση για να βρεθεί το επόμενο α θα είναι $a(k+1) = a(k) + \eta(k) \sum_{y\in Y_k} \frac{b - a^t y}{||y||^2} y$.
Μία σύντομη περιγραφή του αλγορίθμου φαίνεται παρακάτω:

Αλγόριθμος 6. Batch Relaxation with Margin

1 **begin initialize** **a**, margin b, $\eta(0)$, $k=0$
2 **do** $k \leftarrow (k+1) \bmod n$
3 $\mathcal{Y}_k=\{\}$
4 $j=0$
5 **do** $j \leftarrow j+1$
6 **if** $\mathbf{a}^t\mathbf{y}^j<b$, **then** append $\mathbf{y}^j$ to $\mathcal{Y}_k$
7 **until** $j=n$
8 $\mathbf{a} \leftarrow \mathbf{a}+\eta(k)\sum_{\mathbf{y}\in\mathcal{Y}}\frac{b-\mathbf{a}^t\mathbf{y}}{\|\mathbf{y}\|^2}\mathbf{y}$
9 **until** $\mathcal{Y}_k=\{\}$
10 **return** **a**
11 **end**

Η μέθοδος αυτή εφαρμόστηκε και για 2 και για 4 χαρακτηριστικά.
Αρχικά παρουσιάζεται η ταξινόμηση για 2 χαρακτηριστικά, ώστε και παρουσιαστεί και σχηματικά σε 2D:

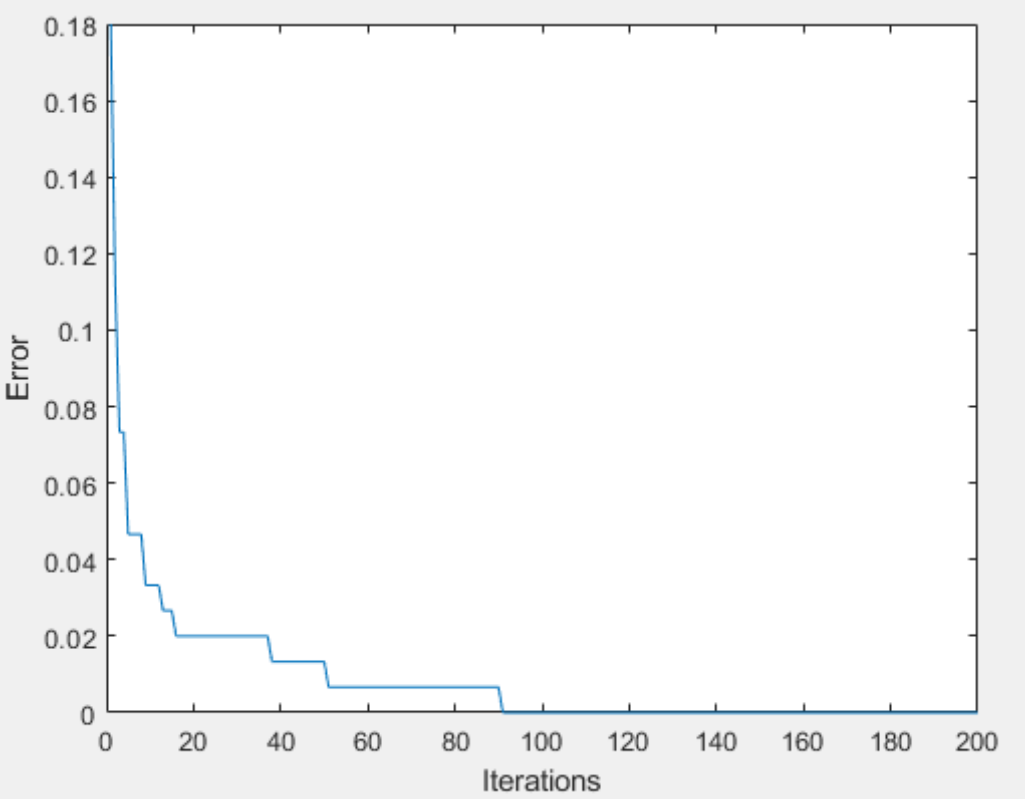
Τελικά τα δεδομένα καταλήγουν να χωρίζονται:



Στο παραπάνω σχήμα φαίνεται η εξέλιξη της ευθείας που χωρίζει τα δεδομένα με την πάροδο των epochs, η οποία, όπως φαίνεται, στην τελευταία εποχή χωρίζει επιτυχώς όλα τα σημεία.

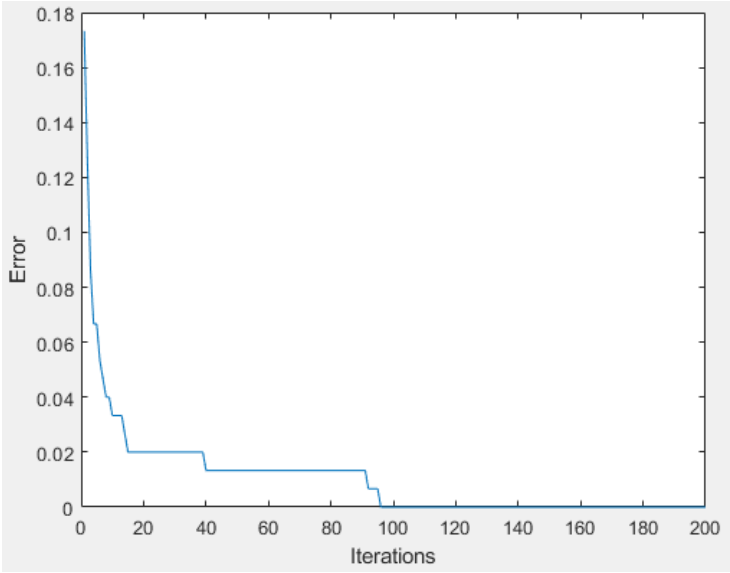
Ο τελικός γραμμικός ταξινομητής είναι ω = [-75.5613, -95.4823, 182.7534].

Για τον επιτυχή διαχωρισμό χρειάστηκαν 91 epochs. Η εξέλιξη του error φαίνεται την επόμενη εικόνα:

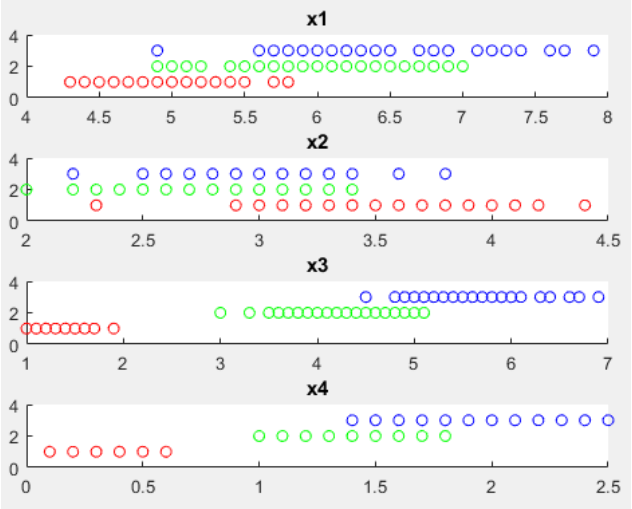


Η ίδια διαδικασία εφαρμόστηκε στη συνέχεια και για τα 4 χαρακτηριστικά των λουλουδιών. Αυτή τη φορά τα epochs (iterations) που χρειάστηκαν για τον επιτυχή διαχωρισμό των λουλουδιών ήταν 96.

Ο τελικός γραμμικός ταξινομητής είναι:
ω = [14.3707, 148.4463, -302.0909, -136.1965, 28.4892].



Το γεγονός ότι χρειάστηκαν περισσότερες επαναλήψεις για το διαχωρισμό των δεδομένων αυτή τη φορά είναι πολύ λογικό, καθώς έχουμε πλέον περισσότερα δεδομένα για επεξεργασία και περισσότερες διαστάσεις στις οποίες πρέπει να χωρίσουμε τα στοιχεία. Επιπλέον, κάνοντας plot τα στοιχεία των 3 ειδών λουλουδιών με βάση κάθε ένα χαρακτηριστικό τους (ω1 κόκκινο, ω2 πράσινο, ω3 μπλε), μπορούμε να παρατηρήσουμε ότι μόνο στα χαρακτηριστικά x3 και x4 είναι γραμμικώς διαχωρίσιμο το iris-setosa από τα άλλα 2 είδη. Είναι επομένως λογικό λαμβάνοντας υπόψιν τα άλλα 2 χαρακτηριστικά να φτάσουμε δυσκολότερα σε μία λύση, καθώς επιβαρύνουμε τον αλγόριθμο με περιττά στοιχεία.



B

Η μέθοδος των ελαχίστων τετραγώνων, σε αντίθεση με τις 2 προηγούμενες, μετατρέπει το πρόβλημα επίλυσης ενός συνόλου γραμμικών ανισοτήτων σε πρόβλημα επίλυσης γραμμικών εξισώσεων.

Πλέον, αντί να προσπαθούμε να βρούμε α τέτοια ώστε $a^t y_i > 0$ για κάθε πρότυπο y_i, ψάχνουμε α τέτοια ώστε $a^t y_i = b_i$.

Η μέθοδος των ελαχίστων τετραγώνων στηρίζεται στο γεγονός ότι τα πρότυπα n είναι περισσότερα από τις διαστάσεις d+1, με αποτέλεσμα το σύστημα να μην έχει ακριβή λύση. Επομένως προσπαθούμε να ελαχιστοποιήσουμε το τετράγωνο του μήκους του διανύσματος σφάλματος e=Ya-b.

Η συνάρτηση κριτηρίου που προσπαθούμε να ελαχιστοποιήσουμε παίρνει πλέον τη μορφή:

$$J_s(a) = ||Ya - b||^2 = \sum_{i=1}^{n} (a^t y_i - b_i)^2$$

Πρέπει οπότε να ισχύει $\nabla J_s(a) = 0 \rightarrow Y^\tau Y a = Y^\tau b$. Αν $Y^t Y$ είναι ομαλός τότε γίνεται:

$$a = (Y^t Y)^{-1} Y^t b$$

Η μέθοδος Widrow-Hoff (Least Mean Squares - LMS) προσπαθεί να πετύχει την παραπάνω ελαχιστοποίηση με αναδρομικές σχέσεις.

Το διάνυσμα κλίσεων θα είναι $\nabla J_s(a) = 2Y^t(Ya - b)$ και η βασική αναδρομική σχέση που θα χρησιμοποιηθεί είναι $a(k+1) = a(k) + \eta(k) Y^t(b - Ya(k))$.

Μία σύντομη περιγραφή του αλγορίθμου φαίνεται παρακάτω:

Αλγόριθμος 8. Widrow-Hoff (LMS)

1 **begin initialize** a, b, κριτήριο θ, η(), k=0
2 **do** $k \leftarrow (k+1) \bmod n$
3 $\mathbf{a} \leftarrow \mathbf{a} + \eta(k)\left(b_k - \mathbf{a}^t \mathbf{y}^k\right)\mathbf{y}^k$
4 **until** $\left|\eta(k)\left(b_k - \mathbf{a}^t \mathbf{y}^k\right)\mathbf{y}^k\right| < \theta$
5 **return** a
6 **end**

Εφαρμόζοντας τον αλγόριθμο για 4 χαρακτηριστικά παρατηρούμε ότι

το error με την τεχνική LS είναι 0

με ω = [0.1313 0.4849 -0.4455 -0.1267 -0.7551],

ενώ με την τεχνική LMS είναι 0.051613 (ελαφρώς χειρότερο)

με ω = [219.9793 -154.2312 -90.7864 55.2999 55.2999].

Αυτό συμβαίνει γιατί

C

Οι τεχνικές ελαχίστων τετραγώνων (LMS που είδαμε στο προηγούμενο ερώτημα) δίνουν πάντα ένα διάνυσμα λύσης που ελαχιστοποιεί το $||Ya - b||^2$. Ωστόσο, αυτό δεν είναι πάντα διαχωριστικό για κάποια προβλήματα.

Ο αλγόριθμος Ho-Kashyap λύνει αναδρομικά το ίδιο πρόβλημα ελαχιστοποίησης, με τον περιορισμό ότι το b>0 δεν συγκλίνει στο 0. Αυτό το πετυχαίνει θέτοντας όλες τις θετικές συνιστώσες του διανύσματος κλίσης ίσες με το 0.

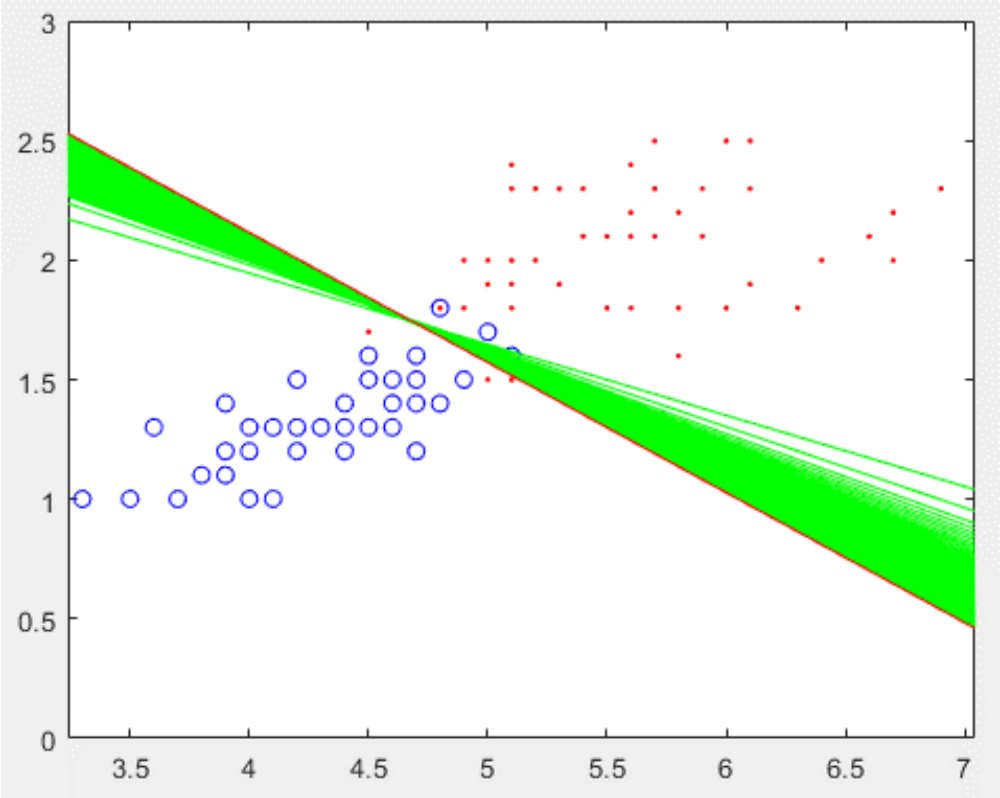
Μία σύντομη περιγραφή του αλγορίθμου φαίνεται παρακάτω:

Αλγόριθμος 9. Ho-Kashyap

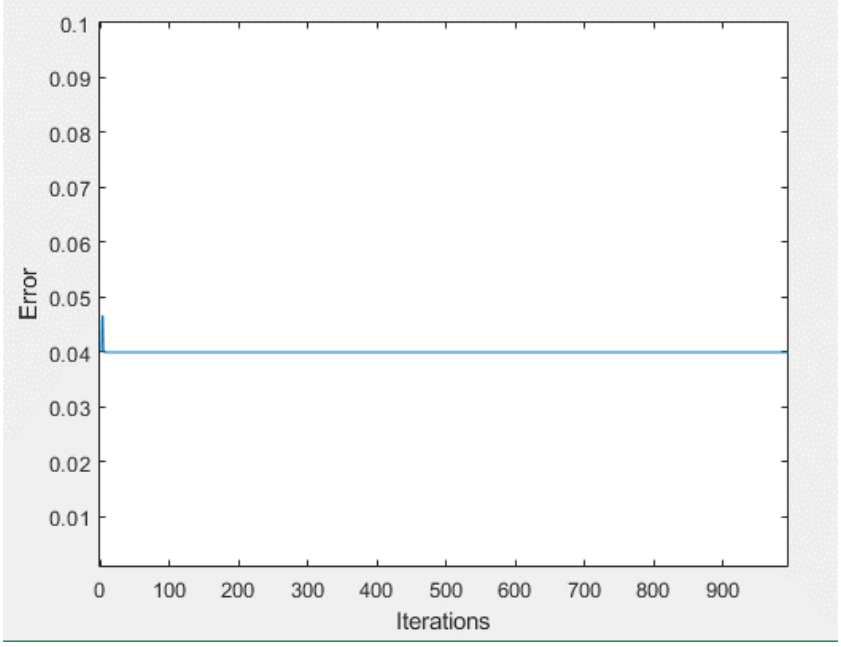
```
1  begin initialize a, b, η()<1, threshold b_min, k_max
2    do  k ← (k+1) mod n
3        e ← Ya-b
4        e+ ← (e+|e|)/2
5        b ← b + 2η(k)e+
6        a ← (YtY)-1Yb
7        if Abs(e)< b_min then return a, b and exit
8    until k= k_max
9    print "No solution found"
10 end
```

Για τον επιτυχή διαχωρισμό χρειάστηκαν μόλις 4 epochs.

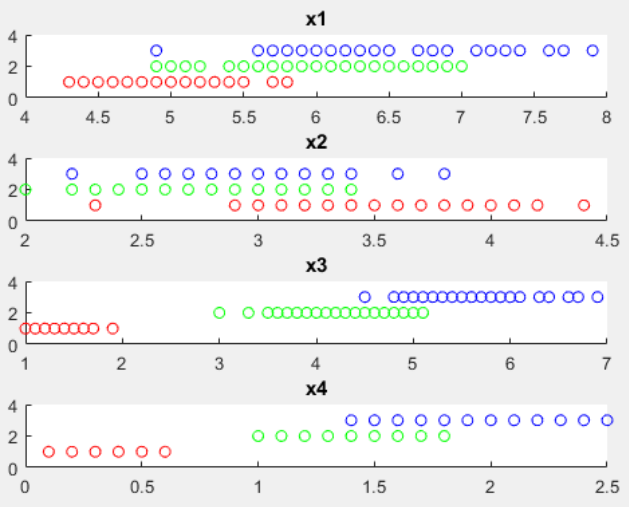
Ο γραμμικός ταξινομητής που βρέθηκε είναι 0
ω = [] λαμβάνοντας υπόψιν 2 χαρακτηριστικά και 0
ω = [0.8758 2.185 -2.8999 -6.0482 12.7335] λαμβάνοντας υπόψιν 4 χαρακτηριστικά.



Η εξέλιξη του error φαίνεται την επόμενη εικόνα:



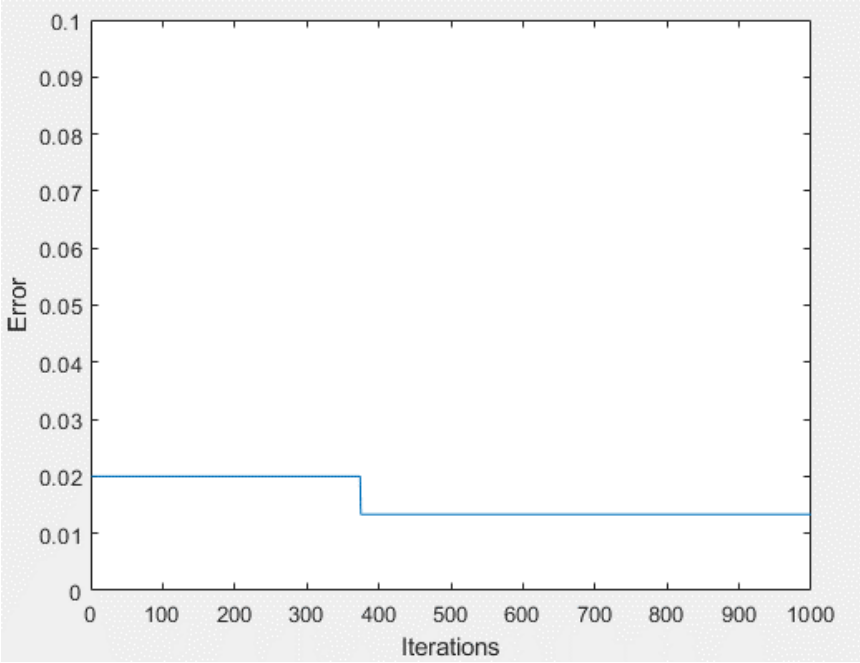
όπως φαίνεται και από το σχήμα, αν λάβουμε υπόψιν μόνο τα 2 χαρακτηριστικά, τα δείγματα (πράσινο και μπλε) δε μπορούν να χωριστούν με μία γραμμή, καθώς τα 2 είδη που εξετάζονται μπλέκονται μεταξύ τους. Έτσι το καλύτερο error που μπορεί να επιτευχθεί φαίνεται να είναι το 0.04.



Η ίδια διαδικασία εφαρμόστηκε στη συνέχεια και για τα 4 χαρακτηριστικά των λουλουδιών, όπου πάλι τα 2 λουλούδια δεν είναι γραμμικά διαχωρίσιμα.

Η διαδικασία ολοκληρώθηκε σε 375 epochs με error 0.0133.
Η επίτευξη μικρότερου error πιθανότατα οφείλεται στην μεγαλύτερη δυνατότητα διάκρισης των ω2 και ω3 συγκρίνοντας τα άλλα 2 χαρακτηριστικά, τουλάχιστον σίγουρα όσον αφορά τα σημεία στα οποία έγινε λάθος διάκριση χρησιμοποιώντας τα 2 μόνο χαρακτηριστικά. Ωστόσο, πάλι δε φαίνεται να μπορούν τα δύο λουλούδια να διακριθούνε τελείως, σε αντίθεση με το σύνολο λουλουδιών ω1 που φαίνεται να διαφέρει πολύ σε σχέση με τα άλλα δύο.

Η μέθοδος Ho-Kashyap προσπαθεί να συνδυάσει τους αλγόριθμους perceptron και LS γι' αυτό και τα αποτελέσματα είναι τόσο ικανοποιητικά



D

Για να ταξινομηθούν τα σημεία για περισσότερες κλάσεις (πχ 3) χρειάζεται να ελεγχθούν οι συναρτήσεις διάκρισης (3 συναρτήσεις εδώ) που ταξινομούν τα σημεία σε 2 κατηγορίες η κάθε μία. Για να ταξινομηθεί πχ ένα σημείο στην κλάση 3 θα πρέπει να ισχύουν:

$$g_3(x) > g_1(x)$$

$$g_3(x) > g_2(x)$$

Χρησιμοποιώντας τη μέθοδο αυτή το αποτέλεσμα του error είναι 0.1533, για γραμμικούς ταξινομητές:

$$\omega_1 = [0.1313, 0.4849, -0.4455, -0.1267, -0.7551]$$

$$\omega_2 = [-0.0431, -0.8814, 0.4370, -0.9664, 2.1260]$$

$$\omega_3 = [-0.0882, 0.3965, 0.0085, 1.0931, -2.3709]$$

Φαίνεται πως, χρησιμοποιώντας και τα 4 χαρακτηριστικά, το x4 επηρεάζει πολύ την ταξινόμηση σε σχέση με τα άλλα 3 χαρακτηριστικά (βλ. 3ο στοιχείο κάθε πίνακα μεγάλο σε σχέση με τα 3 προηγούμενα), ενώ το x1 επηρεάζει το λιγότερο (βλ. 1ο στοιχείο κάθε πίνακα μικρό σε σχέση με τα 3 επόμενα).

E

Εφαρμόζοντας την παραπάνω τεχνική για τα χαρακτηριστικά x1,x2,x3 το error που προκύπτει είναι 0.1867 για τους γραμμικούς ταξινομητές:

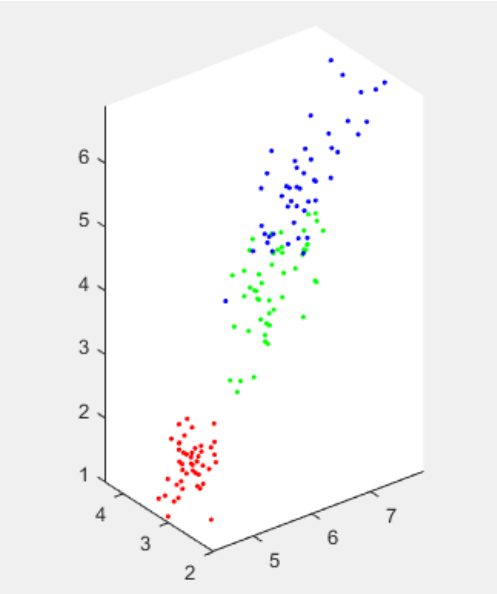
$$\omega_1 = [0.0790, 0.2280, -0.2561, 0.1382]$$

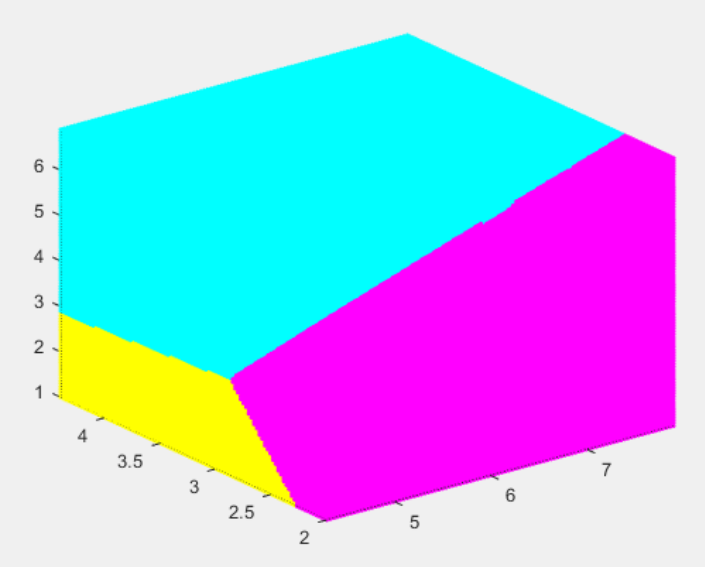
$$\omega_2 = [0.0801, -0.5512, -0.0357, 1,6832]$$

$$\omega_3 = [-0.1590, 0,3233, 0,2918, -0,8214]$$

Σε σχέση με το error που προκύπτει στο D (0.1533), όπου χρησιμοποιούμε και τα 4 χαρακτηριστικά, το error εδώ είναι μεγαλύτερο, όπως και αναμένεται, αφού τώρα έχουμε αφαιρέσει το χαρακτηριστικό x4 που από τα προηγούμενα αποτελέσματα διαπιστώσαμε ότι επηρεάζει περισσότερο από όλα το διαχωρισμό των λουλουδιών. Τα λουλούδια δεν διαχωρίζονται τόσο εύκολα χωρίς αυτό.

Τα υπερεπίπεδα που προέκυψαν παρουσιάζονται στις παρακάτω εικόνες:





Εφαρμόζοντας την παραπάνω τεχνική για τα χαρακτηριστικά x2,x3,x4 το error που προκύπτει είναι 0.1467 για τους γραμμικούς ταξινομητές:

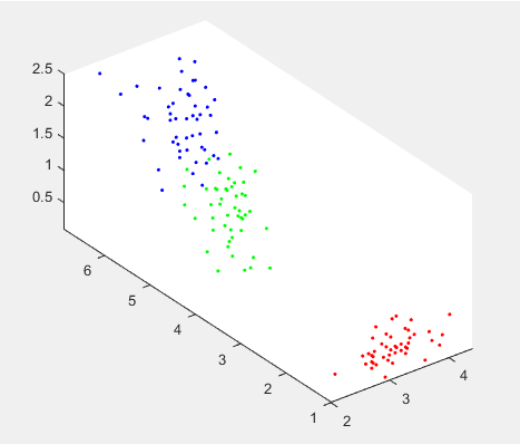
$$\omega_1 = [0.2855, -0.2425, -0.1003, 0.2436]$$

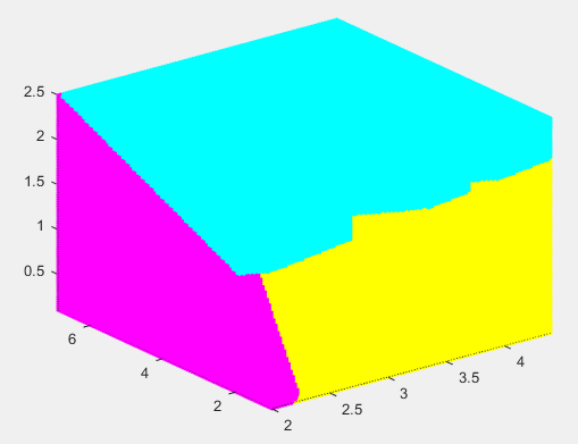
$$\omega_2 = [-0.4548, -0.4407, -0.4711, 1.5232]$$

$$\omega_3 = [0.1694, 0.1982, 0.5714, -0.7668]$$

Σε σχέση με το error που προκύπτει στο D (0.1533), όπου χρησιμοποιούμε και τα 4 χαρακτηριστικά, το error εδώ είναι μικρότερο. Φαίνεται ότι το χαρακτηριστικό x1 μας μπερδεύει περισσότερο τον αλγόριθμο, ίσως γιατί δεν είναι τόσο ξεκάθαρος ο διαχωρισμός των λουλουδιών με βάση το συγκεκριμένο χαρακτηριστικό. Αυτό φάνηκε και στο προηγούμενο ερώτημα, όπου φάνηκε ότι το χαρακτηριστικό x1 επηρρέαζε λιγότερο από όλα το αποτέλεσμα.

Τα υπερεπίπεδα που προέκυψαν παρουσιάζονται στις παρακάτω εικόνες:





F

Το error που προκύπτει είναι 0.3333.