|  |
| --- |
| Χατζηθεοδώρου Ιάσων 03117089 |
| Αναγνώριση Προτύπων |
| 1η Αναλυτική Εργασία |

|  |
| --- |
| el17089@mail.ntua.gr |

## Άσκηση 1.1

Η κατανομή των δειγμάτων είναι η

Άρα περνώντας το λογάριθμο

Παραγωγίζοντας ως προς

Άρα προκύπτει η παρακάτω εξίσωση για τη μεγιστοποίηση της πιθανοφάνειας

Η οποία λύνεται ως προς ως εξής

## Άσκηση 1.2

Η εξίσωση για τη μέθοδο minimax είναι η εξής

Αντικαθιστώντας τις τιμές και έχουμε

Η παραπάνω λύση δεν είναι μοναδική ως προς τις δύο κατανομές. Σαν αντιπαράδειγμα μπορούμε να σκεφτούμε οποιεσδήποτε δύο κατανομές που περιορίζονται η κάθε μία στην περιοχή της. Άρα και τα δύο ολοκληρώματα προκύπτουν μηδέν και ισχύει η σχέση όμως οι κατανομές μπορούν να είναι οποιεσδήποτε.

## Άσκηση 1.3

Στις δύο διαστάσεις η πιθανότητα σφάλματος είναι

όπου προφανώς τα ορίζουν τις περιοχές απόφασης κάθε κατηγορίας. Για να υπολογίσουμε την προβολή ολοκληρώνουμε ως προς , οπότε η πιθανότητα σφάλματος στη μία διάσταση είναι

όπου αντίστοιχα τα ορίζουν τις περιοχές απόφασης. Τα όμως δεν μπορούν να μην περιέχουν τα , γιατί στις δύο διαστάσεις οι περιοχές έχουν επιλεγεί έτσι ώστε να ελαχιστοποιείται η πιθανότητα σφάλματος χρησιμοποιώντας περισσότερη πληροφορία από ότι στη μία διάσταση. Άρα προκύπτουν οι εξής σχέσεις

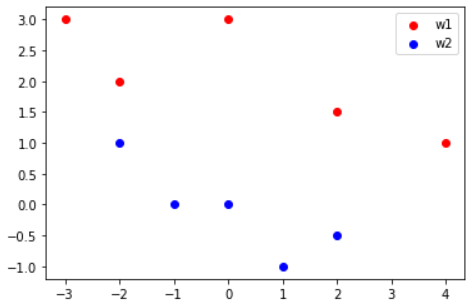
επειδή η πιθανότητα είναι πάντα μη αρνητική και

επειδή . Ομοίως για το ολοκλήρωμα στη περιοχή του . Άρα τελικά ισχύει

Το παραπάνω αποτέλεσμα ισχύει και στην κανονική κατανομή ως ειδική περίπτωση.

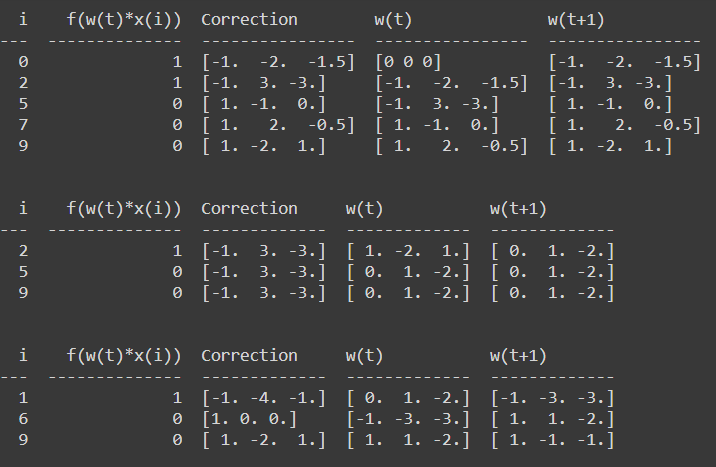
## Άσκηση 1.4

Σχεδιάζοντας τα σημεία προκύπτει η εξής εικόνα, από την οποία είναι προφανές πως οι κλάσεις είναι γραμμικά διαχωρίσιμες

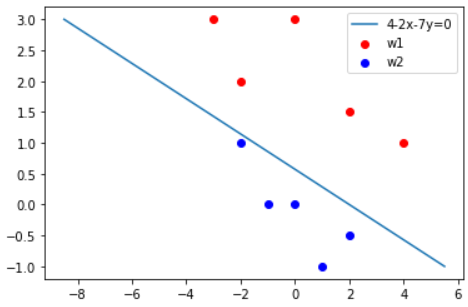


Για διάνυσμα βαρών το perceptron από ένα σημείο και μετά κάνει πάντα λάθος στα δείγματα (-2, 1) και (2, -0.5). Αυτό συμβαίνει γιατί χωρίς το , η ευθεία που προσπαθεί να βρει περνάει από την αρχή των αξόνων και όπως φαίνεται στο σχήμα μια τέτοια ευθεία δεν θα μπορούσε να διαχωρίσει τις δύο κλάσεις.

Για επαυξάνουμε τα δείγματα βάζοντας το 1 στην πρώτη θέση τους. Παρακάτω φαίνονται τα updates που γίνονται στις πρώτες εποχές



Μετά από 14 εποχές ολοκληρώνεται η εκτέλεση και προκύπτουν τα βάρη οπότε η ευθεία είναι η όπως φαίνεται και στη γραφική παράσταση

**

Παρατίθεται ο κώδικας του perceptron

def f(vect, weight):

  res = np.round(np.dot(vect, weight), decimals=2)

  if res < 0:

    return 0

  else:

    return 1

def one\_cycle(weight, x, y, b):

  stuff = []

  temp\_w = weight

  for i in range(len(x)):

    estimate = f(x[i], temp\_w)

    if y[i] == estimate:

      continue

    if np.round(np.dot(weight, x[i]), 2) <= 0 and y[i] == 1:

      weighted\_x = b \* x[i]

    elif np.round(np.dot(weight, x[i]), 2) >= 0 and y[i] == 0:

      weighted\_x =  - b \* x[i]

    new\_w = weight + weighted\_x

    stuff.append([i, estimate, weighted\_x, temp\_w, new\_w])

    temp\_w = new\_w

  return (stuff, temp\_w)

def train(weight, x\_train, y\_train, b\_param, max\_epochs=5):

  stuff = []

  for i in range(max\_epochs):

    (stuff, weight) = one\_cycle(weight, x\_train, y\_train, b\_param)

    if not stuff:

      break

    print(tabulate(stuff, headers=['i', 'f(w(t)\*x(i))', 'Correction', 'w(t)', 'w(t+1)']))

    print('\n')

  return weight

## Άσκηση 1.5

Επιλέγω να αντικαταστήσω τη δεδομένη κατανομή στην και να θεωρήσω ότι άγνωστη είναι η οπότε

Για την ελαχιστοποίηση αρκεί να πάρω τις παραγώγους ως προς και ως εξής

Η παραπάνω σχέση απλοποιείται σε

Παραγωγίζοντας ως προς

Η σχέση απλοποιείται σε

## Άσκηση 1.6

Ερώτημα 1

Το σφάλμα του linear regression προκύπτει από τη διαφορά μεταξύ της σωστής τιμής και της πρόβλεψης δηλαδή

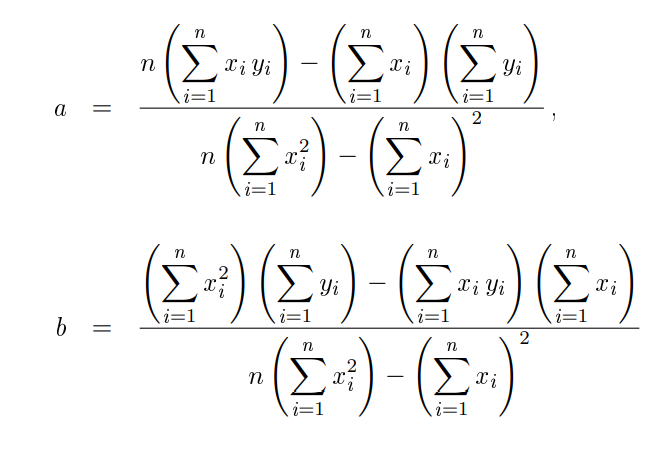
Αν υποθέσουμε ότι αυτό το σφάλμα ακολουθεί κανονική κατανομή τότε η λύση μέγιστης πιθανοφάνειας προκύπτει ως εξής

Η συγκεκριμένη ποσότητα μεγιστοποιείται όταν ελαχιστοποιείται το

Όμως αυτό είναι ακριβώς η λύση που ψάχνει ο αλγόριθμος LMS. Άρα οι λύσεις είναι ισοδύναμες.

Ερώτημα 2

Ψάχνουμε ευθεία της μορφή οπότε χρησιμοποιώντας τη μέθοδο ελάχιστων τετραγώνων έχουμε τους εξής τύπους



Κάνοντας τις πράξεις προκύπτει η ευθεία

Ερώτημα 3

Ο αλγόριθμος LMS κάνει update τα βάρη με βάση τον παρακάτω κανόνα

Για την συγκεκριμένη εκτέλεση αρχικοποιώ το και επιλέγω και η εκτέλεση φαίνεται στον πίνακα

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 0.38 | 2.05 | [1, 1] | [0.067, 0.025] |
| 0.44 | 2.23 | [1.067, 1.025] | [0.071, 0.031] |
| 0.48 | 2.13 | [1.14, 1.056] | [0.048, 0.023] |
| 0.54 | 2.33 | [1.186, 1.08] | [0.056, 0.03] |
| 0.58 | 2.67 | [1.242, 1.11] | [0.078, 0.045] |
| 0.64 | 2.68 | [1.32, 1.155] | [0.062, 0.039] |
| 0.71 | 2.81 | [1.38, 1.195] | [0.057, 0.041] |
| 0.76 | 2.97 | [1.44, 1.236] | [0.058, 0.044] |
| 0.82 | 3.12 | [1.499, 1.281] | [0.057, 0.0467] |
| 0.96 | 3.20 | [1.556, 1.327] | [0.368, 0.353] |

Τελικά προκύπτει η ευθεία

Ερώτημα 4

Εκτελώντας την παραπάνω διαδικασία για 100 εποχές στην Python παίρνω μια πολύ καλύτερη προσέγγιση .

Παρακάτω δίνεται ο κώδικας

def initialise\_weights(size):

  return np.ones(size)

def update\_weights(w, x, y, r = 0.05):

  pred = np.dot(w, x)

  diff = y - pred

  update = r \* diff \* x

  return np.add(w, update)

def mean\_square\_error(w, samples, values):

  error = 0

  for x, y in zip(samples, values):

    pred = np.dot(w, x)

    diff = y - pred

    error += diff \*\* 2

  return 0.5 \* error

def lms(samples, values, r=0.05, max\_epoch=5):

  w = initialise\_weights(samples.shape[1])

  w\_new = initialise\_weights(samples.shape[1])

  epoch = 0

  while epoch < max\_epoch:

    for x, y in zip(samples, values):

      error = mean\_square\_error(w, samples, values)

      w\_new = update\_weights(w, x, y, r)

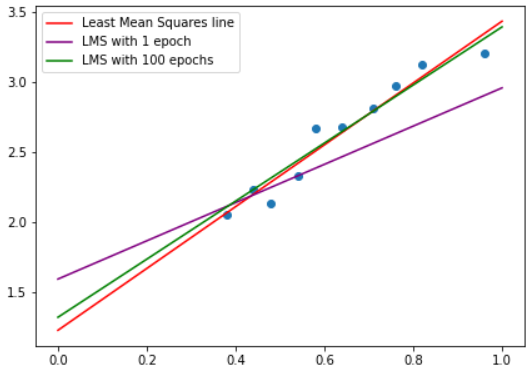
      w = w\_new.copy()

    epoch += 1

  return w

Ερώτημα 5

Όλα τα παραπάνω αποτελέσματα αποτυπώνονται στην γραφική παράσταση



## Άσκηση 1.7

Ερώτημα 1

Η κλάση ενός δείγματος αποφασίζεται με βάση το πρόσημο της διαφοράς των δύο κατανομών

Άρα για ισχύει

Άρα το σημείο απόφασης είναι το

Η πιθανότητα σφάλματος είναι η εξής

Ερώτημα 2

Έστω ότι επιλέξαμε τα και . Η πιθανότητα σφάλματος του νέου ταξινομητή είναι η πιθανότητα το να βρίσκεται πιο κοντά στο από ότι στο . Άρα έχουμε την ανισότητα

Άρα

## 

## Άσκηση 1.8

Το υπολογίζεται ως εξής

Θα πρέπει να ισχύει αλλιώς , οπότε

Λόγω κανονικοποίησης της πρώτης κατανομής θα πρέπει

Αντικαθιστώντας

Η οποία συνάρτηση μεγιστοποιείται για λόγω του περιορισμού και επειδή είναι φθίνουσα.

Άρα οι κατανομές πριν και μετά είναι οι εξής

