

Problem Set 2

Τρίμας Χρήστος

AM:2016030054

Στατιστική Μοντελοποίηση και Αναγνώριση Προτύπων
ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

May 6, 2020

Ερώτηση 1

Λογιστική Παλινδρόμηση, Αναλυτική εύρεση κλίσης:

Στην 1η άσκηση της 2ης εργασίας, θα ασχοληθούμε με την λογιστική παλινδρόμηση. Πρίν όμως γίνει μαθηματική ερμηνεία του αλγορίθμου παλινδρόμησης, θα κάνω visualize το dataset στο οποίο θα δουλέψω πάνω.

Συγκεκριμένα, τα δεδομένα αφορούν βαθμούς εξετάσεων μαθημάτων φοιτητών. Σκοπός είναι

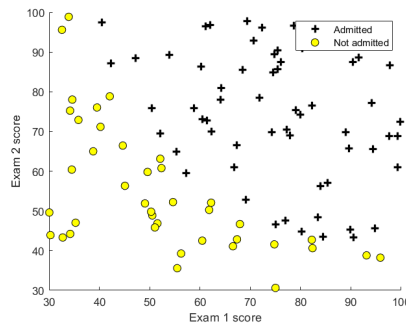


Figure 1: Dataset

να γίνει μια καλή πρόβλεψη με παλινδρόμηση για το εάν ο φοιτητής θα γίνει δεκτός ή όχι σε ένα πανεπιστήμιο με βάση τους βαθμούς του στις δυο αυτές εξετάσεις (άξονες του figure).

Για να τρέξει με επιτυχία ο κώδικας μας, χρειάστηκε να υλοποιηθούν μεταξύ άλλων και κάποιες άλλες συναρτήσεις. Συγκεκριμένα, η sigmoid συνάρτηση δέχεται σαν είσοδο έναν πίνακα z , και εφαρμόζει σε αυτό τον πίνακα τον ακόλουθο μετασχηματισμό:

$$f(z) = \frac{1}{1 + e^{-z}}$$

Η συνάρτηση f , είναι η λογιστική συνάρτηση, η οποία για μεγάλες θετικές τιμές του x , πρέπει να είναι κοντά στο 1, ενώ για μεγάλες αρνητικές τιμές κοντά στο μηδέν. Σε περίπτωση που $x=0$, τότε πρέπει να μας δίνει ακριβώς $\frac{1}{2}$.

Επιπλέον, υλοποιήθηκε η συνάρτηση `costFunction`, η οποία επιστρέφει το κόστος $J(\theta)$ και την κλίση `gradient` αυτής, όπως ορίζεται παρακάτω:

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m [-y^{(i)} \log(h_{\theta}(x^{(i)})) - (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)}))]$$

Και το `gradient` του κόστους ορίζεται ως εξής:

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

Παρατηρείται, ότι αν και η εξίσωση του `gradient` είναι πανομοιότυπη με αυτή της γραμμικής παλινδρόμησης, λόγω του διαφορετικού τρόπου ορισμού της υπόθεσης της παλινδρόμησης $h_{\theta}(x) = g(\theta^T x)$, έχουμε λογιστική παλινδρόμηση και όχι γραμμική.

Μπορούμε με ασφάλεια να συμπεράνουμε ότι μέχρι αυτό το σημείο τα πάντα τρέχουν ομαλά στον κώδικα μας, καθώς για $\theta=0$, τόσο το κόστος, όσο και η κλίση έχουν τα επιθυμητά αποτελέσματα.

Τέλος, για να γίνει αξιολόγηση του μοντέλου, υλοποιήθηκε η συνάρτηση `predict`, η οποία παρήγαγε 1 ή 0 προβλέψεις, δοσμένου κάποιου συνολού δεδομένων και ενός διανύσματος θ . Για έναν μαθητή με σκορ 45,85 η πιθανότητα να περάσει είναι σχεδόν 0.77.

Τα training data με το σύννορο απόφασης φαίνονται στην εικόνα:

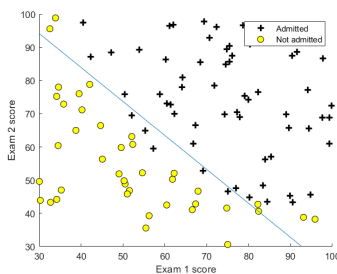


Figure 2: Training data with boundary

Με μια ματιά, μπορεί κανείς να συμπεράνει ότι το αποτέλεσμα είναι αρκετά καλό, και τα δεδομένα τα οποία χάνονται στην πρόβλεψη σημαντικά λίγα, άλλωστε το accuracy είναι 0.89/1. Ωστόσο, αυτό συμβαίνει, διότι τα δεδομένα είναι και σχεδόν γραμμικά διαχωρίσιμα. Στην αμέσως επόμενη άσκηση θα μελετήσουμε την εφαρμογή της λογιστικής παλινδρόμησης σε ένα dataset με μη γραμμικά διαχωρίσιμα δεδομένα.

Ερώτηση 2

Λογιστική Παλινδρόμηση με Ομαλοποίηση:

Στο συγκεκριμένο πρόβλημα, εφαρμόστηκε η ομαλοποιημένη λογιστική παλινδρόμηση σε ένα σετ από δεδομένα μιας εταιρείας παραγωγής τσιπ. Συγκεκριμένα, θέλουμε να αυτοματοποιήσουμε την διαδικασία του ποιοτικού ελέγχου, και έχοντας τα δεδομένα από διαφορετικές δοκιμές θέλουμε να κατασκευάσουμε έναν regressor με την βέλτιστη δυνατή ακρίβεια.

Όπως και στην άσκηση 1, ξεκινάμε με την οπτικοποίηση των δεδομένων.

Παρατηρείται, ότι ο διαχωρισμός των δεδομένων δεν είναι δυνατόν να πραγματοποιηθεί, από

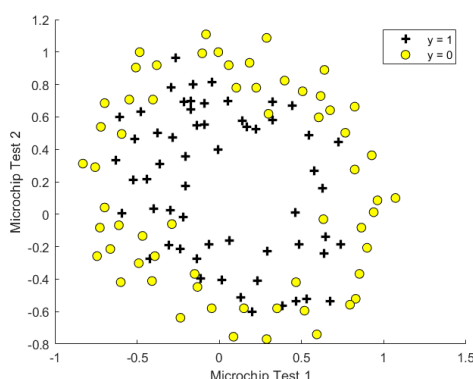


Figure 3: Original Dataset

μια γραμμή. Για αυτό το λόγο, η απευθείας χρήση της λογιστικής παλινδρόμησης δεν είναι καλή επιλογή για λύση του προβλήματος μας.

Ένας τρόπος για να γίνει αποτελεσματικά η παλινδρόμηση, είναι να προσθέσουμε επιπλέον χαρακτηριστικά στα δεδομένα μας ή αλλιώς να απεικονίσουμε το dataset σε χώρο υψηλότερων διαστάσεων, όπου ο διαχωρισμός είναι εφικτός. Η διαδικασία αυτή, επιτυγχάνεται με την συνάρτηση `mapFeature`, η οποία κάνει `map` τα χαρακτηριστικά στους πολυωνυμικούς όρους της κάθε κλάσης, μέχρι και την 6η δύναμη.

$$\text{mapFeature}(x) = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_1^2 \\ x_2^2 \\ \vdots \\ x_1 x_2^5 \\ x_2^6 \end{bmatrix}$$

Αποτέλεσμα αυτού του μετασχηματισμού, είναι η δημιουργία ενός 28x1 διανύσματος. Η παλινδρόμηση με λογιστικό παλινδρομητή, θα δημιουργήσει ένα αρκετά πιο σύνθετο όριο διαχωρισμού, όπως θα δούμε και παρακάτω.

Σε ότι αφορά την συνάρτηση κόστους και την κλίση της, αυτά δίνονται από τις σχέσεις:

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m (-y^{(i)} \ln(h_{\theta}(x^{(i)})) - (1 - y^{(i)}) \ln(1 - h_{\theta}(x^{(i)}))) + \frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2$$

Και το gradient από την σχέση:

$$\nabla J(\theta) = \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)} + \frac{\lambda}{m} \theta_j$$

Με χρήση της αρχικής τιμής θ , το αρχικό κόστος είναι 0.693147

Παρακάτω, απεικονίζεται το σύνορο απόφασης για διαφορετικές τιμές του $\lambda = 0, 1, 10, 100$.

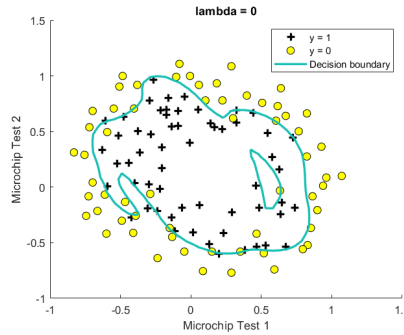


Figure 4: Decision Boundary, $\lambda=0$

Με την χαμηλότερη τιμή του $\lambda=0$, ο regressor πετυχαίνει το καλύτερο accuracy '88.983051', ωστόσο, το όριο απόφασης είναι όπως περιμέναμε αρκετά πολύπλοκο. Το φαινόμενο αυτό ονομάζεται over-fitting of the data. Αυτό, δεν είναι καλό σύνολο απόφασης, καθώς ένα σημείο πχ στο (0.6,0), το δέχεται σαν σημείο με $y=0$, ενώ θα έπρεπε να ανήκει στην άλλη κλάση.

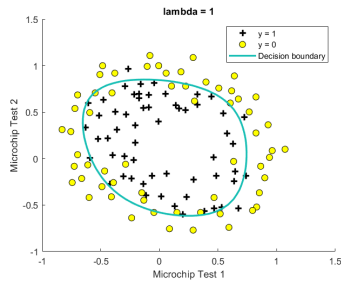


Figure 5: Decision Boundary, $\lambda=1$

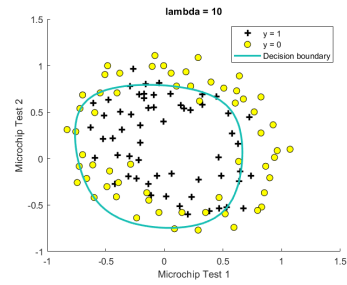


Figure 6: Decision Boundary, $\lambda=10$

Για $\lambda=1$, το σύνολο απόφασης, γίνεται όλο και πιο απλό στο σχεδιασμό του, ωστόσο η ακρίβεια πέφτει με την αύξηση του λ . Συγκεκριμένα για $\lambda=1$ και $\lambda=10$, δεν συναντάμε φαινόμενο over-fitting ή under-fitting(οπως θα δούμε για μεγάλο λ , πχ $\lambda=100$). Σε κάθε περίπτωση, το accuracy για $\lambda=1$ είναι '83.050847' και για $\lambda=10$, '74.576271'

Τέλος, όπως υπόθηκε και προηγουμένως, για $\lambda=100$ έχουμε και την χειρότερη απόδοση '61.016949', και φαινόμενο under-fitting.

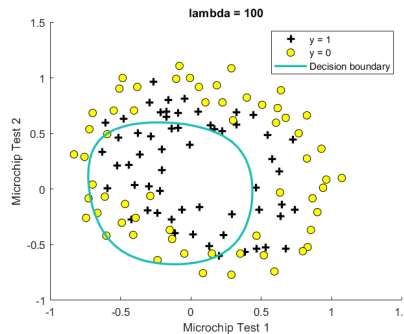


Figure 7: Decision Boundary, $\lambda=100$

Ερώτηση 4

Εκτίμηση Παραμέτρων και Ταξινόμηση :

Στην άσκηση 4 θα πειραματιστούμε με τους εκτιμητές Maximum Likelihood, και συγκεκριμένα χρησιμοποιήσουμε την κατανομή Bernoulli για να μοντελοποιήσουμε το πρόβλημα μας.

Δεδομένου ενός dataset που περιλαμβάνει εικόνες από χειρόγραφα ψηφία, θέλουμε να κατασκευάσουμε έναν εκτιμητή, ο οποίος θα κάνει σωστή ταξινόμηση των ψηφίων, δηλαδή όταν σαν είσοδο έχουμε ένα χειρόγραφο 0, ο classifier μας, θα προβλέπει ποιο ψηφίο είναι αυτό, χρησιμοποιώντας την MLE πιθανότητα του ψηφίου. Με πολύ απλά λόγια, θα βρίσκει ποιο ψηφίο έχει την μεγαλύτερη πιθανότητα από τα άλλα ψηφία, και θα επιλέγει αυτό.

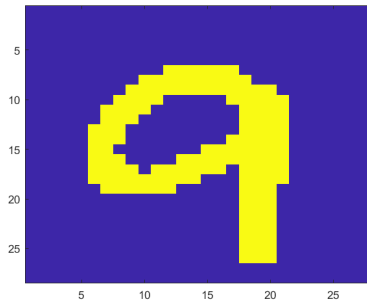


Figure 8: Χειρόγραφο 9

Για το ερώτημα α δείτε την εικόνα στον φάκελο.

1) Τελικά, καταλήγουμε ότι:

$$MLE : \quad p = \frac{1}{n} \sum_{\forall i} x^{y^i}$$

2) Στη συνέχεια, υπολογίζουμε με τον τύπο παραπάνω τις πιθανότητες, χρησιμοποιώντας τα δεδομένα εκπαίδευσης train0,...,train9. Επειδή, χρησιμοποιείται η λογαριθμική μορφή της πιθανοτικής εκτίμησης Bernoulli, όσες τιμές $p(0,...,9)$ είναι μηδέν, τις θέτουμε σε μια τιμή πάρα πολύ μικρή και στην συνέχεια υπολογίζουμε το $\log L(p)$, για κάθε ψηφίο, για κάθε δείγμα του ψηφίου.

Τέλος, για να οπτικοποιήσουμε το αποτέλεσμα μας, πολλαπλασιάζουμε όλα τα δείγματα κάθε train set, με την $\log L(p)$ του και παίρνουμε ένα διάνυσμα 784×1 , το οποίο κάνουμε reshape, με αποτέλεσμα να λάβουμε μια εικόνα, της οποίας τα pixel αντιπροσωπεύουν την πιθανότητα p^{y^i} της κατανομής Bernoulli.

3) Για το 3ο σκέλος της άσκησης, θέλαμε να τεστάρουμε την απόδοση του Naive Bayes Classifier μας, δοκιμάζοντας τον σε ένα σύνολο από δεδομένα δοκιμής (test0,...,test9). Κάθε τέτοιο

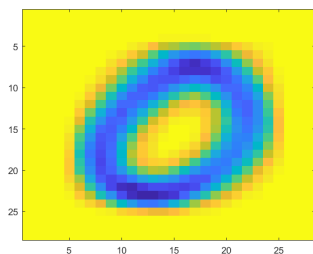


Figure 9: Digit with probability pixels

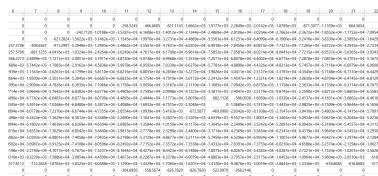


Figure 10: Απόσπασμα από τον πίνακα οπτικοποίησης

σέτ, περιλαμβάνει 500 εικόνες, 28x28 πίξελ η κάθε εικόνα.

Το ερώτημα σε αυτό το κομμάτι είναι, πως θα αξιολογήσουμε τον classifier μας ? Η απάντηση είναι αρκετά απλή. Καθώς έχουμε γνώση τι ψηφία έχει το κάθε test σέτ, θα τα δώσουμε στον classifier και θα περιμένουμε να επιλέξει το σωστό. Πχ αν του δώσουμε εικόνα από το train0, θα περιμένουμε να μας δείξει ότι το στοιχείο μηδέν έχει την μεγαλύτερη πιθανότητα από τα υπόλοιπα ψηφία.

Αυτή η διαδικασία θα γίνει για κάθε σέτ εικόνων δοκιμής, και στη συνέχεια θα αξιολογήσουμε την απόδοση μέσω των εξής metrics:

- 1) Accuracy.
- 2) Confusion Matrix.

Τα διανύσματα x_0, \dots, x_9 , επιστρέφουν την θέση, των ψηφίων που προβλέπει ο ταξινομητής μας. Καθώς το matlab ξεκινάει να μετράει τις τιμές από το 1, το 1 θα αντιστοιχεί στο ψηφίο 0, το 2 στο 1 κλπ. Άρα, στο διάνυσμα x_0 , περιμένουμε οι τιμές να είναι 1. Ωστόσο, λόγω λάθος ταξινόμησης των ψηφίων, παρατηρούνται κάποιες αστοχίες.

Συγκεκριμένα, στον ακόλουθο πίνακα φαίνεται η ακρίβεια κάθε ταξινομητή για το δικό του MLE:

$$AccuracyMatrix = \begin{array}{cc} & \begin{array}{c} digit \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array} \\ \begin{array}{c} accuracy \\ 0.8820 \\ 0.9460 \\ 0.7480 \\ 0.8300 \\ 0.7520 \\ 0.6900 \\ 0.8240 \\ 0.7880 \\ 0.6960 \\ 0.8180 \end{array} \end{array}$$

Τέλος, υπολογίστηκαν και οι πίνακες confusion(δες εκφώνηση για τον ορισμό τους) κάθε ψηφίου. Παρακάτω απεικονίζονται για διευκόλυνση έτσι ώστε να μην χρειαστεί να διαβαστούν από το matlab.CM είναι ο conf matrix κάθε ψηφίου.

$$CM0 = \begin{array}{cc} & \begin{array}{c} wrongdigit \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array} \\ \begin{array}{c} numOfMissClassification \\ 0 \\ 1 \\ 0 \\ 2 \\ 30 \\ 13 \\ 0 \\ 12 \\ 1 \end{array} \end{array}$$

$$CM1 = \begin{array}{cc} \textit{wrongdigit} & \textit{numOfMissClassification} \\ \begin{array}{c} 0 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array} & \left(\begin{array}{c} 0 \\ 2 \\ 3 \\ 0 \\ 13 \\ 4 \\ 0 \\ 5 \\ 0 \end{array} \right) \end{array}$$

$$CM2 = \begin{array}{cc} \textit{wrongdigit} & \textit{numOfMissClassification} \\ \begin{array}{c} 0 \\ 1 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array} & \left(\begin{array}{c} 8 \\ 12 \\ 36 \\ 6 \\ 2 \\ 11 \\ 14 \\ 33 \\ 4 \end{array} \right) \end{array}$$

$$CM3 = \begin{array}{cc} \textit{wrongdigit} & \textit{numOfMissClassification} \\ \begin{array}{c} 0 \\ 1 \\ 2 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array} & \left(\begin{array}{c} 1 \\ 10 \\ 5 \\ 4 \\ 23 \\ 6 \\ 13 \\ 10 \\ 13 \end{array} \right) \end{array}$$

$$CM4 = \begin{array}{cc} \textit{wrongdigit} & \textit{numOfMissClassification} \\ \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array} & \left(\begin{array}{c} 1 \\ 1 \\ 3 \\ 0 \\ 4 \\ 16 \\ 2 \\ 5 \\ 92 \end{array} \right) \end{array}$$

$$CM5 = \begin{array}{cc} \textit{wrongdigit} & \textit{numOfMissClassification} \\ \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 6 \\ 7 \\ 8 \\ 9 \end{array} & \left(\begin{array}{c} 15 \\ 2 \\ 2 \\ 67 \\ 19 \\ 7 \\ 6 \\ 14 \\ 23 \end{array} \right) \end{array}$$

$$CM6 = \begin{array}{cc} \textit{wrongdigit} & \textit{numOfMissClassification} \\ \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 7 \\ 8 \\ 9 \end{array} & \left(\begin{array}{c} 7 \\ 10 \\ 24 \\ 0 \\ 11 \\ 33 \\ 0 \\ 3 \\ 0 \end{array} \right) \end{array}$$

$$CM7 = \begin{array}{cc} \textit{wrongdigit} & \textit{numOfMissClassification} \\ \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 8 \\ 9 \end{array} & \left(\begin{array}{c} 1 \\ 21 \\ 9 \\ 3 \\ 17 \\ 0 \\ 0 \\ 10 \\ 45 \end{array} \right) \end{array}$$

$$CM8 = \begin{array}{cc} \textit{wrongdigit} & \textit{numOfMissClassification} \\ \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 9 \end{array} & \left(\begin{array}{c} 6 \\ 17 \\ 11 \\ 49 \\ 11 \\ 18 \\ 2 \\ 3 \\ 35 \end{array} \right) \end{array}$$

$$CM9 = \begin{array}{cc} \textit{wrongdigit} & \textit{numOfMissClassification} \\ \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array} & \left(\begin{array}{c} 4 \\ 8 \\ 4 \\ 8 \\ 49 \\ 7 \\ 0 \\ 6 \\ 5 \end{array} \right) \end{array}$$

Ερώτηση 5

Bayesian εκτίμηση παραμέτρου :

Σε αυτή την άσκηση μελετήθηκε η Bayesian εκτίμηση της μ , γνωρίζοντας ότι η συνάρτηση πυκνότητας πιθανότητας των μετρήσεων του σετ δεδομένων ακολουθεί γκαουσιανή κατανομή, και συγκεκριμένα: $p(x) \sim N(\mu, \sigma^2)$, και γνωστή τυπική αποκλιση $\sigma=1.25$. Υποθέτοντας ότι έχουμε κάποια εκ των προτέρων πληροφορία για την κατανομή της μέσης τιμής μ :

$$p(\mu) \sim N(\mu_0, \sigma_0^2)$$

Για το α ερώτημα, ξέρουμε ότι $\mu_0 = 0$ και $\sigma_0^2 = 10\sigma^2$. Ακόμα ισχύει (τύποι σημειώσεων) ότι:

$$p(\mu|D) \sim N(\mu_n, \sigma_n^2)$$

ή αλλιώς:

$$p(\mu|D) = \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left[-\frac{1}{2}\left(\frac{\mu - \mu_n}{\sigma_n}\right)^2\right]$$

Με

$$\mu_n = \left(\frac{n\sigma_0^2}{n\sigma_0^2 + \sigma^2}\right) \bar{x}$$

και

$$\sigma_n^2 = \frac{\sigma_0^2 \sigma^2}{n\sigma_0^2 + \sigma^2}$$

Καθώς οι τιμές του n μεταβάλλονται από το 1 ως το 25, η γραφική της $p(\mu, H(n))$ αλλάζει και αυτή όπως φαίνεται στο ακόλουθο σχήμα.

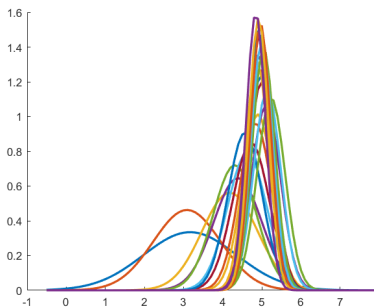


Figure 11: $p(\mu, H(n))$, για $n=1:25$

Στο ερώτημα Β, η πυκνότητα πιθανότητα ακολουθεί την εξής Gaussian κατανομή:

$$p(x|D) \sim N(\mu_n, \sigma^2 + \sigma_n^2)$$

Έτσι, προσαρμόζοντας τους τύπους για το σ_n και το μ_n , κρατώντας σταθερό το $n=25$, και αλλάζοντας την τιμή του $\sigma_0^2 = 10\sigma^2$, $\sigma_0^2 = \sigma^2$, $\sigma_0^2 = 0.1\sigma^2$, $\sigma_0^2 = 0.01\sigma^2$ προκύπτει η ακόλουθη γραφική.

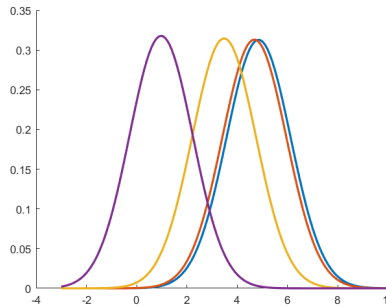


Figure 12: $\sigma_0^2 = 10\sigma^2$, $\sigma_0^2 = \sigma^2$, $\sigma_0^2 = 0.1\sigma^2$, $\sigma_0^2 = 0.01\sigma^2$

Παρατηρείται, ότι το πλάτος των Gaussian είναι ίδιο, και μάλιστα εξαρτάται από την τιμή του n . Από την άλλη, όσο μικραίνει η διασπορά τόσο η καμπύλη μας ολισθαίνει προς τα αριστερά. Αρχικά, πέφτοντας μια τάξη μεγέθους η ολίσθηση δεν είναι αισθητή, όσο όμως πέφτει και άλλο η τάξη μεγέθους, η καμπύλη ολισθαίνει όλο και περισσότερο.

Ερώτηση 7

Support Vector Machines :

Στην τελευταία άσκηση της παρούσας εργασίας, ασχοληθήκαμε με τον αλγόριθμο των support vector machines, και τον εφαρμόστηκε σε ένα σύνολο από τεχνητά δεδομένα. Πιο συγκεκριμένα, ορίστηκε ένα σύνολο n παραδειγμάτων $\{(x^{(1)}, y^{(1)}), \dots, (x^{(n)}, y^{(n)})\}$, όπου $x^{(i)} \in \mathbb{R}^{1 \times 2}$ και $y^{(i)} \in \{-1, 1\}$.

Για να προβλέψουμε τις τιμές του y , από αυτές του x , θα χρησιμοποιηθεί η τεχνική svm. Για την υλοποίηση του svm, απαραίτητη προϋπόθεση, είναι η εύρεση ενός υπερεπιπέδου, και των support vectors. Μοντελοποιώντας μαθηματικά το πρόβλημα, καταλήγουμε στο εξής:

$$\hat{y} = \begin{cases} 1 & , w x^T + w_0 > 0 \\ -1 & , w x^T + w_0 < 0 \end{cases}$$

Οι παράμετροι του προβλήματος βρίσκονται με την λύση του προβλήματος βελτιστοποίησης, κάτω από κάποιους περιορισμούς. Συγκεκριμένα, η συνάρτηση Lagrange δίνεται από την σχέση:

$$\hat{L}(\lambda) = \sum_{i=1}^n \lambda^{(i)} - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \lambda^{(i)} \lambda^{(j)} y^{(i)} y^{(j)} (x^{(i)} (x^{(j)})^T$$

Την οποία συνάρτηση την μεγιστοποιούμε κάτω από τους KKT περιορισμούς:

$$\lambda^{(i)} \geq 0, \forall i \in [1, n]$$

$$\sum_{i=1}^n \lambda^{(i)} y^{(i)} = 0$$

Όπως σε κάθε άσκηση, έτσι και σε αυτή, ξεκινάμε με την οπτικοποίηση του dataset.

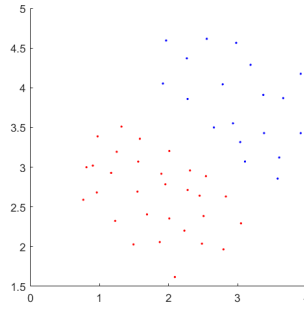


Figure 13: The dataset

Για την εύρεση των πολλαπλασιαστών λ , το πρόβλημα ανάγεται σε πρόβλημα quadratic programming. Στην πραγματικότητα, η quadprog function του matlab, βρίσκει το ελάχιστο για το εξής πρόβλημα:

$$\min_x \frac{1}{2} x^T H_x + f^T x, \text{ such that } \begin{cases} A * x \leq qb, \\ Aeq * x = beq, \\ lb \leq x \leq ub. \end{cases}$$

Με H , να είναι ο πίνακας που περιγράφει την μορφή του hyperplane.

Σε ότι έχει να κάνει με την επιλογή των υπολοίπων μεταβλητών, όλες σχεδόν επιλέχτηκαν με βάση παραδείγματα που βρέθηκαν στον διαδίκτυο. Έξαίρεση αποτελεί το upper limit του λ , καθώς χωρίς τον περιορισμό από το C , δεν θα είχαμε τα επιθυμητά αποτελέσματα.

Με τον υπολογισμό των λ , είναι εφικτή η επιλογή των support vectors για το πρόβλημα. Στο σχήμα που ακολουθεί, έχουν "τονιστεί" τα στοιχεία εκείνα, από τα οποία θα διέρχονται τα sv, για τιμή $C=10$.

Τέλος, για να σχεδιαστούν σωστά τα support vectors, υπολογίζεται η max norma, με σκοπό τον υπολογισμό του width.

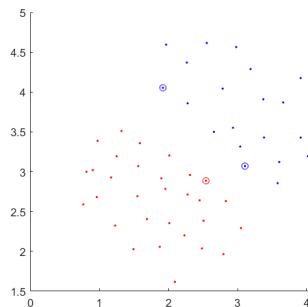


Figure 14: C=10

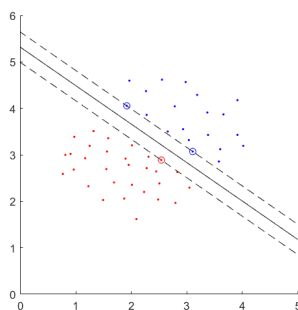


Figure 15: Support vectors and 2-D Hyperplane line,C=10

Για το 2ο ερώτημα της άσκησης προστέθηκε ένα ακόμα δείγμα το οποίο ωστόσο με το μάτι δημιουργεί "πρόβλημα" στο classification.

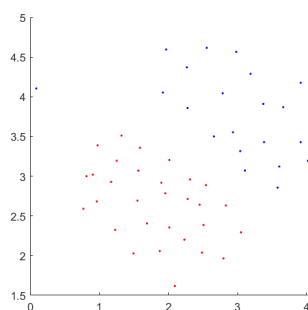


Figure 16: 51 samples dataset

Σε αυτή την περίπτωση και αλλάζοντας την τιμή του C από 10 σε 100, και συμπεριλαμβάνοντας τον περιορισμό $0 \leq \lambda^i \leq C^i \forall i$, καταλήγουμε στο ακόλουθο classification:

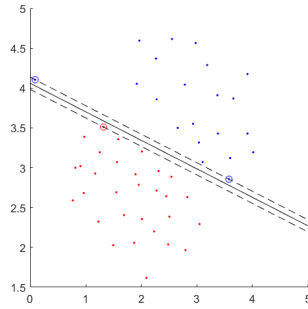


Figure 17: Support vectors and Hyperplane, $C=100$

Για το 3ο και τελευταίο μέρος της άσκησης, τα δεδομένα μας είναι μη γραμμικά διαχωρίσιμα, με αποτέλεσμα η άμεση εφαρμογή του svm αλγορίθμου να μην είναι δυνατή.

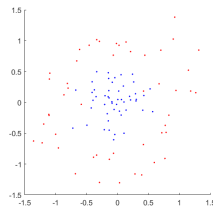


Figure 18: Non linear separable dataset

Για να γίνει λοιπόν ο διαχωρισμός, έγινε προσθήκη ενός επιπλέον χαρακτηριστικού στα δεδομένα μας, και συγκεκριμένα προστέθηκε η norm-2 στο τετράγωνο του x . Με την προσθήκη αυτή, και κρατώντας την μορφή του hyperplane, γραμμική πετυχαίνεται μια σχετικά καλή ταξινόμηση των δειγμάτων κάθε κλάσης.

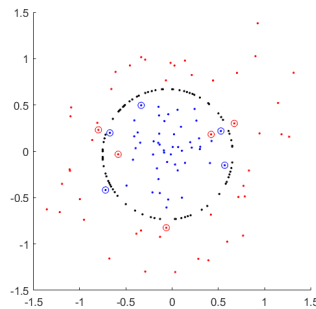


Figure 19: Classification of the dataset