

---

## Αναφορά αποτελεσμάτων HomeWork 2

Μπανέλας Δημήτριος: 2018030140

Μάθημα: Στατιστική Μοντελοποίηση και Αναγνώριση Προτύπων

Διδάσκων θεωρίας: Ζερβάκης Μιχάηλ

Διδάσκων εργαστηρίου: Διακολουκάς Βασίλειος

---

## Περιεχόμενα αναφοράς και οδηγίες εκτέλεσης κώδικα

Ο κώδικας που αναπτύχθηκε για τις ανάγκες των ασκήσεων βρίσκεται στα αρχεία με κώδικα που έχουν ήδη δοθεί. Ο τρόπος εκτέλεσης έμεινε ίδιος με πριν. Πιο συγκεκριμένα, μετά απο κάθε ενέργεια, το πρόγραμμα γίνεται pause και περιμένει το πάτημα ενός πλήκτρου από τον χρήστη για να συνεχιστεί. Σημαντικό είναι, να υπάρχει παρατήρηση του console καθ' όλη τη διάρκεια εκτέλεσης του κάθε προγράμματος, καθώς εκεί εμφανίζονται σημαντικές πληροφορίες.

Σε πολλά σημεία υπάρχουν blocks κώδικα που βρίσκονται μέσα σε σχόλια. Ο κώδικας αυτός γράφτηκε για την καλύτερη κατανόηση των εννοιών και δεν είναι απαραίτητος για την ορθή λειτουργία του κάθε προγράμματος. Τέλος, ο κώδικας περιέχει επεξηγηματικά σχόλια έτσι ώστε να μπορεί να διαβαστεί με σχετική ευκολία.

# Θέμα 1: Λογιστική Παλινδρόμηση: Αναλυτική εύρεση κλίσης (Gradient)

Στο α) κομμάτι της άσκησης ασχοληθήκαμε με την αναλυτική εύρεση της κλίσης της συνάρτησης κόστους. Η διαδικασία έγινε χειρόγραφα και θα βρεθεί στο αντίστοιχο αρχείο.

Στο β) κομμάτι ασχοληθήκαμε με την εφαρμογή της λογιστικής παλινδρόμησης πάνω σε ένα dataset με βαθμούς απο φοιτητές, με στόχο να προβλέψουμε εάν θα γίνουν δεκτοί σε ένα πανεπιστήμιο. Το dataset με το οποίο δουλέψαμε φαίνεται παρακάτω.

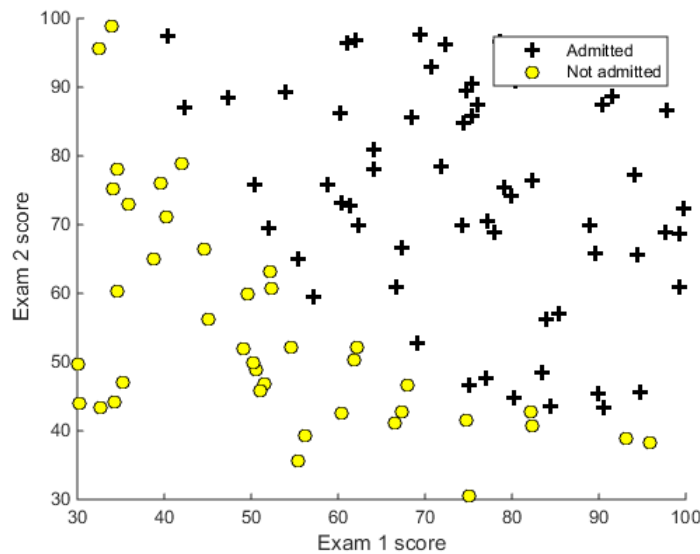


Figure 1: Visualized dataset

Ελαχιστοποιώντας την cross entropy συνάρτηση κόστους υπολογίζουμε τις παραμέτρους  $\theta$  και έπειτα σχεδιάζουμε την ευθεία παλινδρόμησης που ορίζουν τα δεδομένα. Τέλος, χρησιμοποιώντας την sigmoid function και το υπολογισμένο  $\theta$  προβλέπουμε εάν ένας φοιτητής θα γίνει αποδεκτός, με βάση τους βαθμούς του. Παρατίθεται το διάγραμμα με το όριο απόφασης.

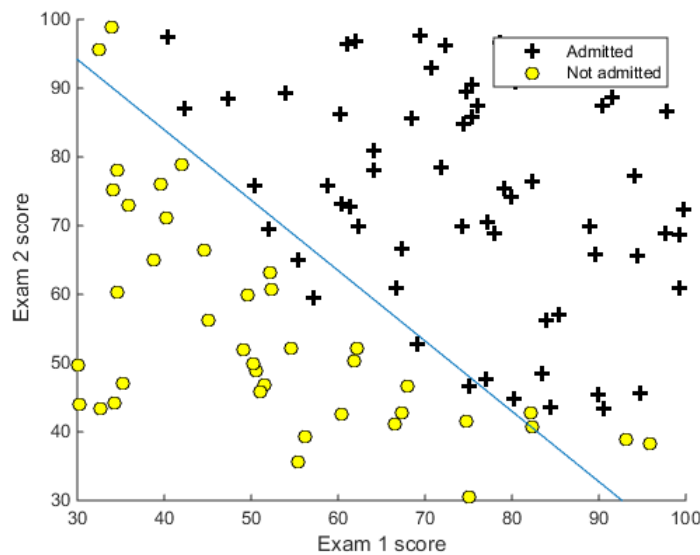


Figure 2: Decision boundary

Το παραπάνω μοντέλο επιτυγχάνει ποσοστά σωστής ταξινόμησης της τάξης του 89 - 90%, κάτι το οποίο είναι αναμενόμενο δεδομένης της γραμμικής διαχωρισιμότητας των δεδομένων του dataset. Παρατηρούμε ότι το όριο απόφασης μας δείχνει με αρκετή ακρίβεια τη γραμμή που διαχωρίζει admitted και not admitted φοιτητές. Στην επόμενη άσκηση θα μελετήσουμε ένα dataset του οποίου τα δεδομένα δεν είναι γραμμικά διαχωρίσιμα.

## Θέμα 2: Λογιστική Παλινδρόμηση με Ομαλοποίηση

Στο α) κομμάτι της άσκησης ασχοληθήκαμε με την αναλυτική εύρεση της κλίσης της συνάρτησης κόστους. Η διαδικασία έγινε χειρόγραφα και θα βρεθεί στο αντίστοιχο αρχείο.

Στο β) κομμάτι ασχοληθήκαμε με την εφαρμογή της λογιστικής παλινδρόμησης με ομαλοποίηση πάνω σε ένα dataset με 2 tests από microchips με σκοπό να πραγματοποιήσουμε quality analysis, δηλαδή να βρούμε ποιά από τα chips δουλεύουν σωστά. Αρχίζουμε με το να οπτικοποιήσουμε τα δεδομένα.

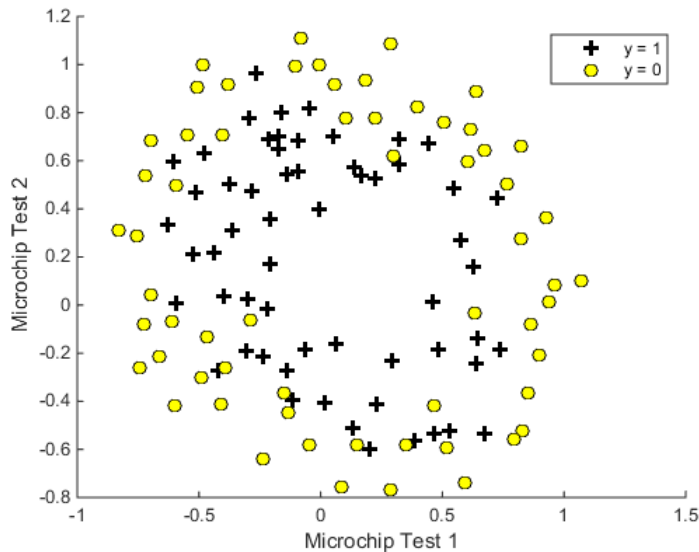


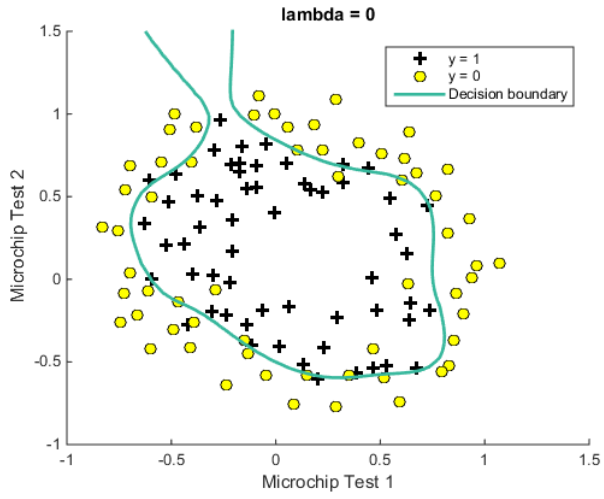
Figure 3: Decision boundary

Επειδή τα δεδομένα δεν είναι γραμμικώς διαχωρίσιμα θα χρειαστεί να τα προβάλουμε σε χώρο μεγαλύτερης διάστασης έτσι ώστε να υπολογίσουμε ένα σύνορο απόφασης. Αυτό επιτυγχάνεται με το να προβάλουμε τα δεδομένα στον χώρο που ορίζουν όλοι οι όροι των πολυωνύμων  $x_1$  και  $x_2$  βαθμού μέχρι και 6. Παρακάτω φαίνονται οι σχέσεις που χρησιμοποιήθηκαν.

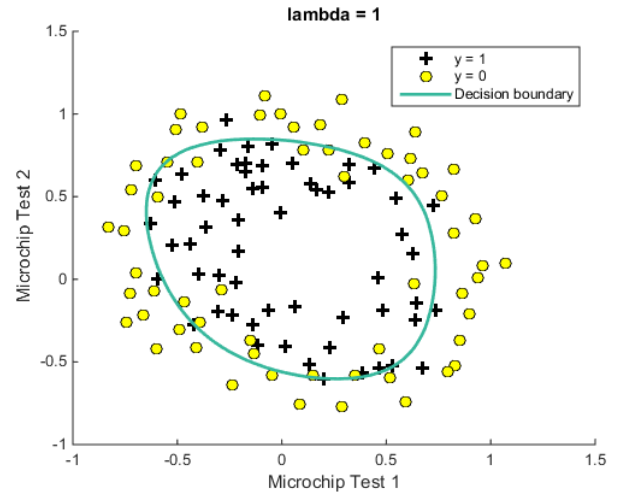
$$P(x_1, x_2) = \sum_{i=0}^6 \sum_{j=0}^i x_1^{i-j} x_2^j$$

$$\text{mapFeature}(x) = [1, x_1, x_2, x_1^2, x_1x_2, x_2^2, x_1^3, \dots, x_1x_2^5, x_2^6]^T$$

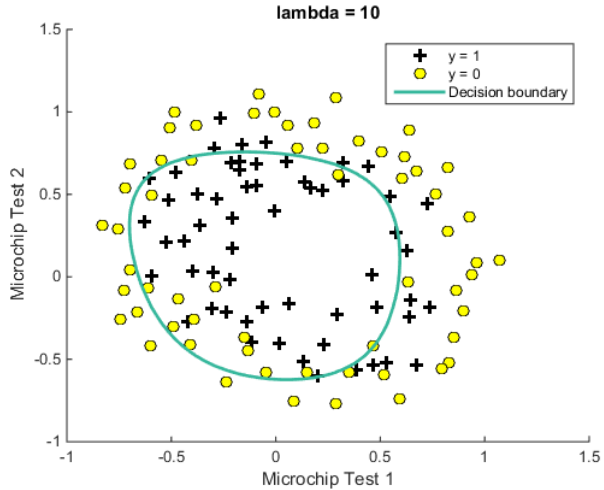
Τέλος, βελτιστοποιώντας τους παράγοντες  $\theta$  σχεδιάσαμε και παρήρασαμε τα σύνορα απόφασης για τις διάφορες τιμές του  $\lambda$ .



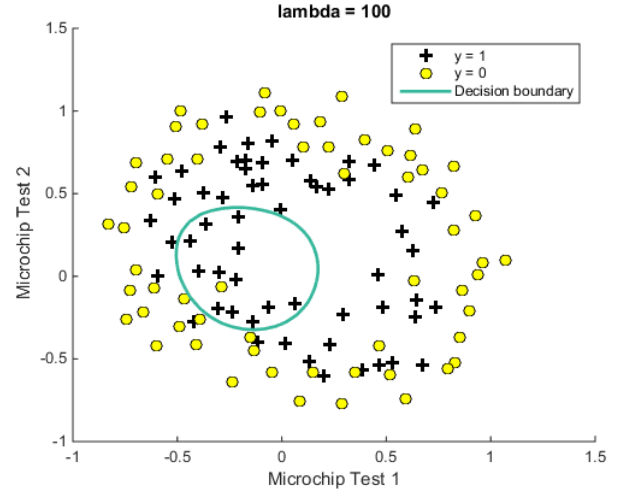
(a) Accuracy = 87.28 %



(b) Accuracy = 82.20 %



(c) Accuracy = 74.57 %



(d) Accuracy = 60.16 %

Παρατηρούμε πως για  $\lambda = 0$  το μοντέλο μας πετυχαίνει το μεγαλύτερο accuracy πάνω στο δοσμένο dataset, ωστόσο είναι φανερό πως έχει γίνει overfitting πάνω σε αυτό το dataset. Αυτό θα έχει ως αποτέλεσμα το μοντέλο να έχει κακή γενίκευση όταν προκειται για δεδομένα εκτός του training dataset. Αντίθετως,  $\lambda = 100$  παρατηρούμε το φαινόμενο του underfitting. Το μοντέλο δεν έχει καταφέρει να μοντελοποιήσει σωστά το training dataset, επομένως έχει πολύ κακό accuracy πάνω σε αυτό, καθώς και καμία πιθανότητα γενίκευσης. Για  $\lambda = 1$  και  $\lambda = 10$  παρατηρούμε πως έχουμε βρεί ένα αρκετά γενικό όριο απόφασης, το οποίο έχει αρκετά καλό accuracy στα δεδομένα του training dataset και θα μπορεί να γενικεύσει και σε δεδομένα εκτός αυτού.

## Θέμα 4: Εκτίμηση Παραμέτρων και Ταξινόμηση (ML-Naive Bayes Classifier)

Σε αυτή την άσκηση θα χρησιμοποιήσουμε τον Maximum Likelihood ετιμητή και τον Naive Bayes ταξινομητή για την αναγνώριση ψηφίων του MNIST dataset, το οποίο περιέχει εικόνες  $28 \times 28$  με χειρόγραφα ψηφία, από το 1 μέχρι το 9.

Αρχικά, υποθέτοντας ότι το κάθε pixel της εικόνας μοντελοποιείται με κατανομή Bernoulli, θα υπολογίσουμε, χρησιμοποιώντας τον εκτιμητή μέγιστης πιθανοφάνειας, την παράμετρο  $p^{y_i}$  που ζητείται στην εκφώνηση. Οι αναλυτικοί υπολογισμοί παρατίθενται στο αντίστοιχο αρχείο.

$$p^{y_i} = \frac{1}{n} \cdot \sum_{k=1}^n x_k$$

Στη συνέχεια, χρησιμοποιώντας τον παραπάνω τύπο, εκπαιδεύσαμε τα μονέλα, υπολογίζοντας το διάνυσμα παραμέτρων  $p$  για κάθε ψηφίο. Η διαδικασία αυτή έγινε ως εξής: Για κάθε ψηφίο, πήραμε όλες τις training εικόνες και απο αυτές υπολογίσαμε την πιθανότητα το κάθε pixel να έχει την τιμή 1, δηλαδή να είναι άσπρο. Παρακάτω φαίνεται η οπτικοποίηση του

διανύσματος  $p$  αυτού για τα ψηφία 8 και 9.

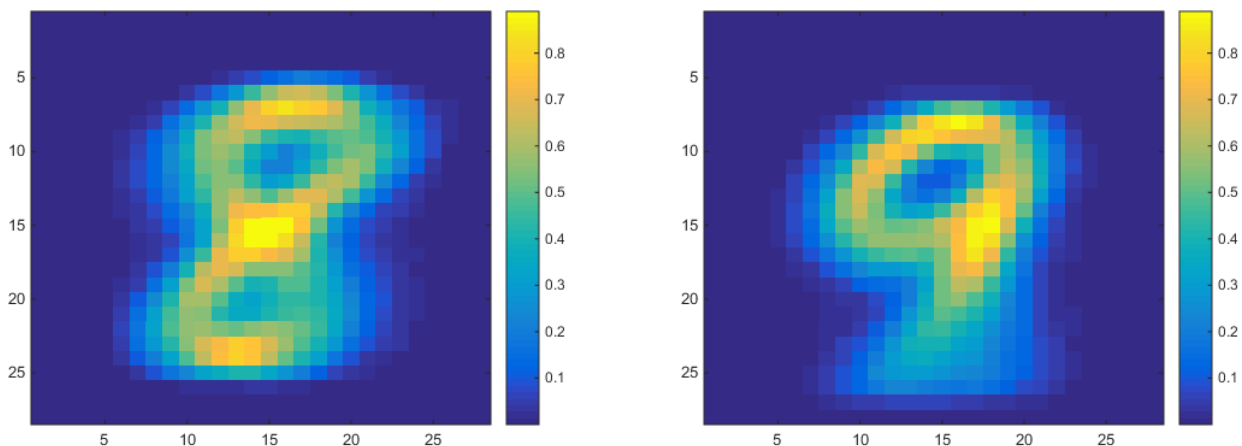


Figure 5:  $p$  parameters visualization of digits 8 and 9

Όπως μπορούμε να παρατηρήσουμε οι παραπάνω οπτικοποιήσεις μας δίνουν μία αρκετά ακριβή αναπαράσταση των πραγματικών ψηφίων, μιας και τα pixels της εικόνας είναι είναι όσο φώτεινα, όσο η πιθανότητα τους να είναι ίση με 1.

Τέλος, δοκιμάσαμε την ικανότητα του μοντέλου μας δινόντας του να ταξινομήσει τις εικόνες από τα testsets. Η ταξινόμηση έγινε με βάση το μεγαλύτερο log-Likelihood των pixels της κάθε εικόνας, δεδομένης της παραμέτρου  $p$ . Δηλαδή:

$$decided\_digit = \operatorname{argmax}_i X \cdot \log p_i + (1 - X) \cdot \log(1 - p_i)$$

Όπου  $X$  το διάνυσμα των pixels μίας εικόνας και  $p_i$  το διάνυσμα παραμέτρων του  $i$ -οστού ψηφίου. Παρακάτω παρατίθεται το confusion matrix της ταξινόμησης, το οποίο μας δείχνει πόσες φορές ένα κάποιο στοιχείο ταξινομήθηκε σωστά και πόσες ως ένα άλλο ψηφίο. Η πρώτη αριστερά στήλη αντιπροσωπεύει τα πραγματικά ψηφία ενώ η πρώτη πάνω γραμμή αντιπροσωπεύει τα classified ψηφία.

digit	0	1	2	3	4	5	6	7	8	9
0	441	0	1	0	2	30	13	0	12	1
1	0	472	2	3	0	13	4	0	6	0
2	8	12	375	37	7	2	9	13	33	4
3	1	10	4	416	5	23	5	13	10	13
4	1	1	5	0	374	4	15	2	5	93
5	14	2	3	67	20	345	7	6	15	21
6	7	10	28	0	7	33	412	0	3	0
7	1	21	9	3	15	0	0	395	10	46
8	6	16	12	49	11	18	1	3	349	35
9	3	8	3	8	51	7	0	6	5	409

Παρατηρούμε πως συνήθως, όταν ένα ψηφίο θα γίνει missclassified ως ένα ψηφίο πολύ παρόμοιο, πχ 4 με 7, 4 με 9, 7 με 9. Γενικά, βλέπουμε πως τα ψηφία ταξινομούνται σωστά τις περισσότερες φορές. Η ακρίβεια του μοντέλου φαίνεται στον παρακάτω πίνακα.

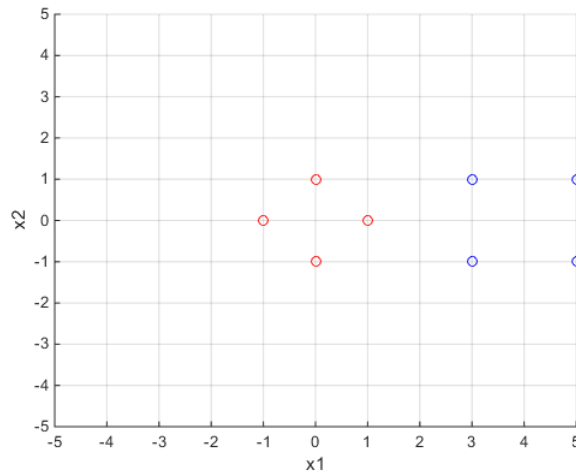
Accuracy of digit(%)	0	1	2	3	4	5	6	7	8	9
	88.20	94.4	75	83.20	74.8	69	82.4	79	69.8	81.8

Η γενική ακρίβεια του μοντέλου κυμαίνεται στα 79.76% κάτι το οποίο είναι αναμενόμενο καθώς έχουμε κάνει τη θεώρηση ότι τα pixels είναι ανεξάρτητα, η οποία είναι λάθος. Πχ. ένα μαύρο pixel στη γωνία της εικόνας θα σημαίνει ότι με μεγάλη πιθανότητα ότι και τα κοντινά του θα είναι μαύρα.



## Θέμα 5: Support Vector Machines (Αναλυτική βελτιστοποίηση με KKT)

Για το Α μέρος, τα δείγματα παρατίθενται στο παρακάτω σχήμα.



Διαισθητικά προτείνουμε ότι το βέλτιστο γραμμικό υπερεπιπεδο διαχωρισμού είναι η ευθεία  $x_1 = 2$

Για το Β μέρος, τα δείγματα, πριν και μετά τον μετασχηματισμό παρατίθενται παρακάτω.

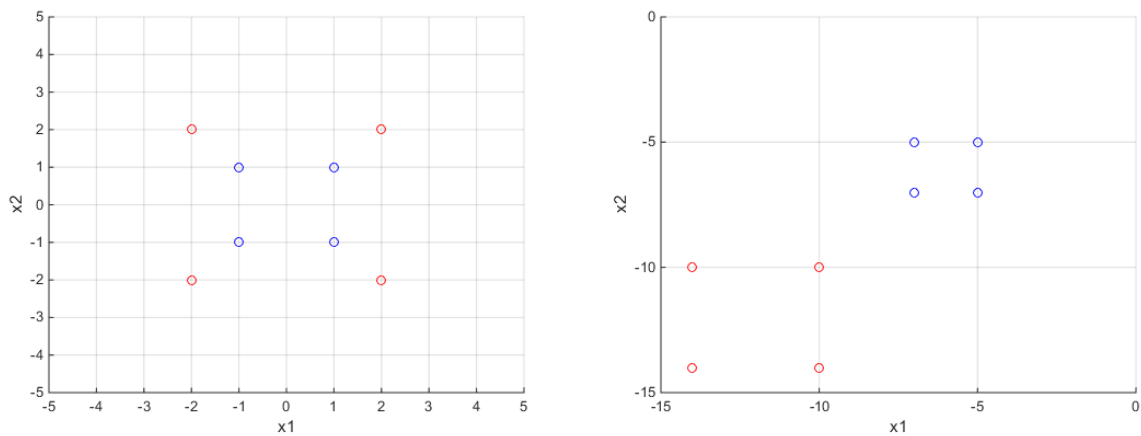


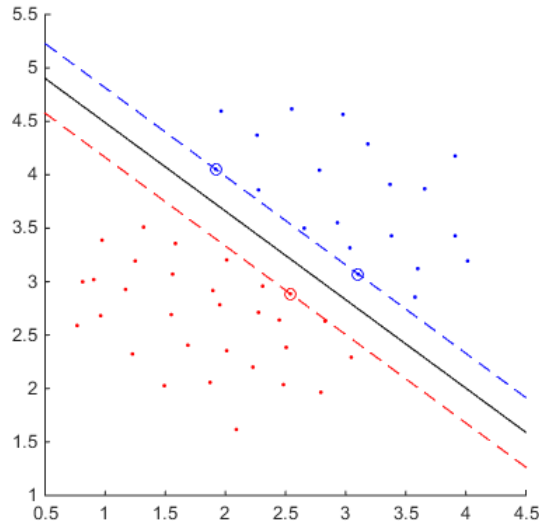
Figure 6: Vectors before and after kernel trick

Σε αυτή την περίπτωση, πριν γίνει το kernel trick δεν μπορούμε να προτείνουμε διασθητικά ένα γραμμικό επίπεδο διαχωρισμού. Η αναλυτική βελτιστοποίηση έγινε χειρόγραφα και βρίσκεται στο αντίστοιχο αρχείο.

## Θέμα 6: Support Vector Machines (Εφαρμογή σε τεχνητό σύνολο δεδομένων)

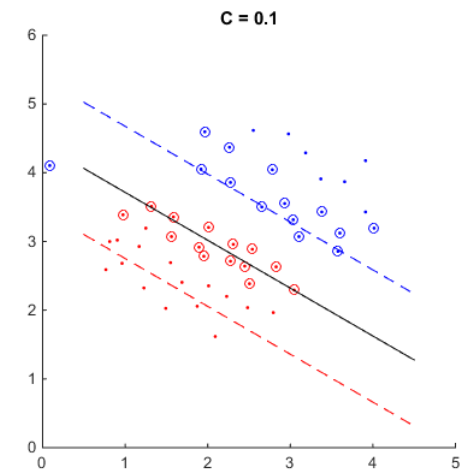
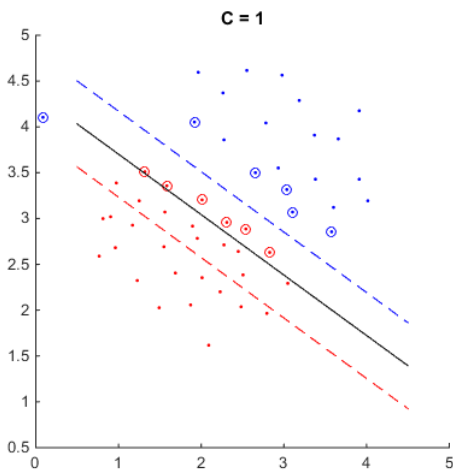
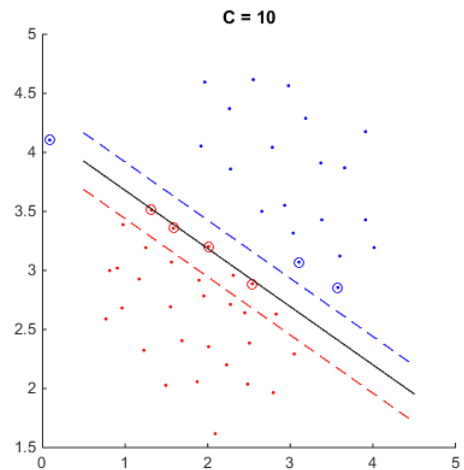
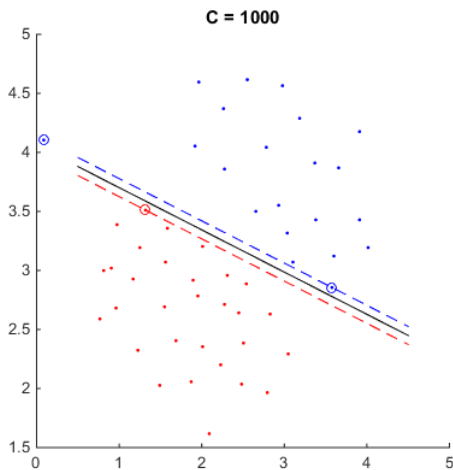
### Μέρος Α

Για το Α μέρος, ελαχιστοποιήσαμε την κατάλληλη συνάρτηση κάνοντας χρήση της συνάρτησης `chadprog` της MATLAB αφού υπολογίσαμε τους απαραίτητους για τη λειτουργία της πίνακες. Τα δείγματα, support vectors και το γραμμικό υπερεπίπεδο διαχωρισμού παρατίθενται στο παρακάτω plot.



## Μέρος Β

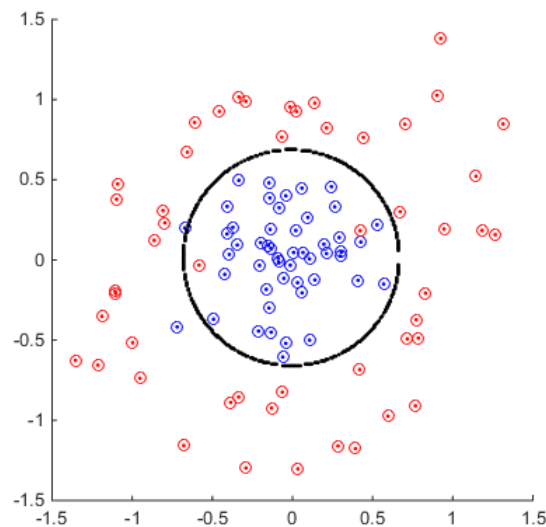
Στο μέρος Β στα δεδομένα που έχουμε προστίθεται ένας outlier. Αυτό είναι κακό καθώς είτε θα μας κάνει τα δεδομένα να μην είναι γραμμικά διαχωρίσιμα, είτε θα μας δώσει ένα υπερεπίπεδο γραμμικού διαχωρισμού το οποίο δεν είναι βέλτιστο για τα υπόλοιπα δεδομένα. Για να λύσουμε αυτό το πρόβλημα, εισάγουμε τις μεταβήτες περιθωρίου slack variables, οι οποίες επιτρέπουν σε μερικά δείγματα να βρίσκονται από την λάθος πλευρά της επιφάνειας απόφασης. Το επίπεδο ανοχής συμβολίζεται με τη μεταβλητή  $C$ . Υλοποιώντας την λαγκριανζιανή του προηγούμενου ερωτήματος και προσθέτοντας τον περιορισμό  $0 \leq \lambda^{(i)} \leq C$  παίρνουμε τα παρακάτω αποτελέσματα.



Παρατηρούμε πως όσο μεγαλύτερο είναι το  $C$  τόσο πλησιάζουμε στο hard margin SVM του πρώτου μέρους. Όσο το  $C$  τόσο η ανοχή αυξάνεται και περισσότερα δείγματα βρίσκονται στο εσωτερικό της ζώνης διαχωρισμού των κλάσεων.

### Μέρος Γ

Στο μέρος Γ έχουμε στη διάθεση μας ένα dataset με μη γραμμικά διαχωρίσιμα δείγματα. Για να καταφέρουμε να βρούμε ένα υπερεπίπεδο διαχωρισμού θα χρειαστεί να εφαρμόσουμε kernel trick. Θα προβάλουμε τα δεδομένα στον τρισδιάστατο χώρο και θα βρούμε ένα υπερεπίπεδο διαχωρισμού εκεί. Υλοποιώντας τα προηγούμενα, προκύπτει το παρακάτω όριο απόφασης.



Χρησιμοποιώντας το kernel trick καταφέραμε να διαχωρίσουμε ένα dataset από μη γραμμικά διαχωρίσιμα δείγματα.