ΑΦΜ	ΕΠΩΝΥΜΟ	ΟΝΟΜΑ	ΕΞΑΜΗΝΟ
58352	ΤΟΚΑΤΛΙΔΗΣ	ΓΕΩΡΓΙΟΣ	9^{Ο}

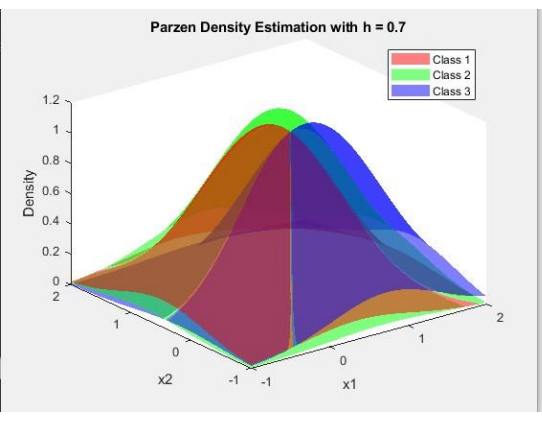
ΕΡΓΑΣΙΑ #2

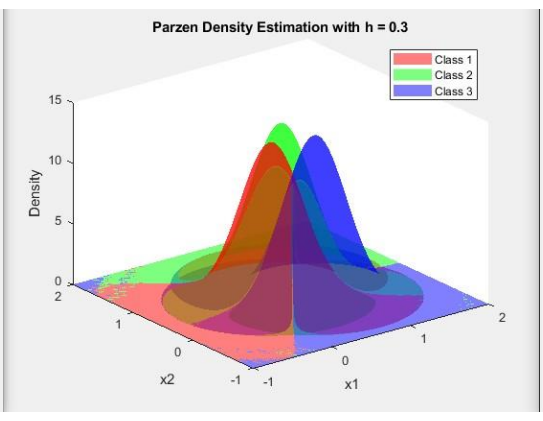
ΜΑΘΗΜΑ : ΑΝΑΓΝΩΡΙΣΗ ΠΡΟΤΥΠΩΝ

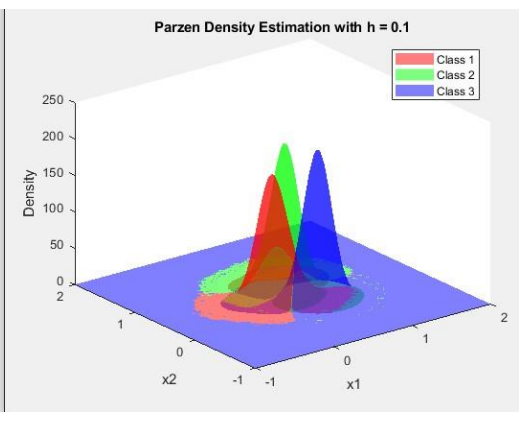
ΑΣΚΗΣΗ 1

Σκοπός αυτής της άσκησης είναι η εκτίμηση δεσμευμένων κατανομών πιθανότητας με χρήση μη-παραμετρικών τεχνικών που με τη βοήθειά τους, σε συνδυασμό με τον κανόνα Bayes, θα οριστούν οι περιοχές απόφασης του προβλήματος , για διαφορετικές παραμέτρους κάθε φορά.

Στο 1^{ο} ερώτημα βάση αυτών που ζητούνται, υπολογίζονται οι εκτιμήσεις των δεσμευμένων πυκνοτήτων πιθανοτήτων $p(x|\omega_i)$ με χρήση παραθύρων Parzen.





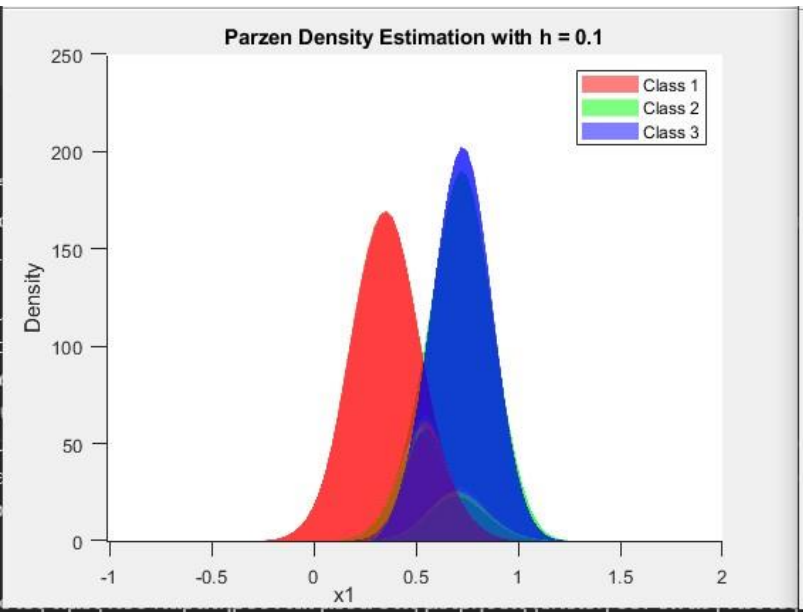


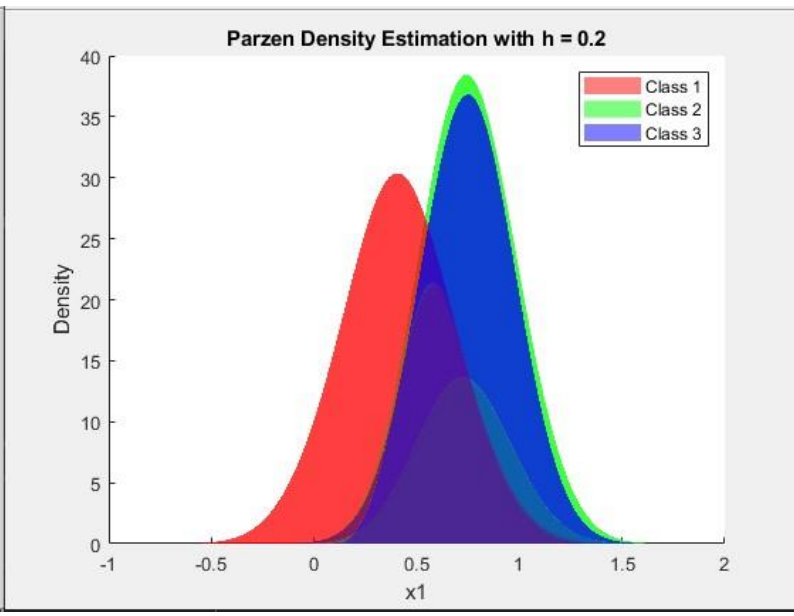
Αυτή είναι η γραφική αναπαράσταση από τις εκτιμητέες PDFs για διαφορετικές τιμές h που ζητούνταν, που στην ουσία καθορίζει κατά πόσο κάθε δείγμα επηρεάζει τη τιμή της $p(x|\omega_i)$, σε όλο τον χώρο.

Για μια μικρή τιμή μήκους παραθύρου h, το εύρος επιρροής κάθε δείγματος στη συνολική κατανομή είναι περιορισμένο και το τελικό αποτέλεσμα της εκτιμητέας κατανομής είναι «αγκαθωτό» και ασυνεχές. Για μια μεγάλη τιμή h το αποτέλεσμα είναι ακριβώς το αντίθετο, δηλαδή βλέπουμε μια κατανομή πολύ ομαλή και απλωμένη στον χώρο, πράγμα που στερεί τις ιδιαιτερότητες του αντικειμένου που μελετάται ως προς τα χαρακτηριστικά του και το υπεραπλουστεύει. Η ίδια ακριβώς φιλοσοφία και ο μαθηματικός φορμαλισμός ακολουθείται σε προβλήματα SPH.

Μια σημαντική σημείωση είναι πως τα δεδομένα έχουν κανονικοποιηθεί ως προς τις μέγιστες τιμές που παρατηρούνται μέσα στις μετρήσεις (επειδή δεν είναι γνωστές οι ιδιότητές τους για να τεθούν αντικειμενικά όρια που φράσσουν τις επιτρεπτές τιμές των διακριτών χαρακτηριστικών τους). Αυτό γίνεται με σκοπό να απαλειφθεί η πιθανή τάση του συστήματος να κατηγοριοποιεί δεδομένα με κύριο χαρακτηριστικό αυτό που έχει μεγαλύτερη τάξη μεγέθους, πράγμα που μαθηματικά εξηγείται από τη συνάρτηση παραθύρου ϕ() που κάνει χρήση της ευκλείδειας απόστασης, μιας μετρικής που δεν είναι ανεξάρτητη της κλίμακας των μεγεθών (δεν είναι scale-invariant δηλαδή).

Σύμφωνα με τη θεωρητική ανάλυση που κάναμε ειδικά για το μήκος παραθύρου, είναι λογική ρύθμιση, σε περίπτωση που αντιμετωπίζεται μικρότερο dataset, να αυξηθεί η τιμή του h για να επιτευχθεί όμοια ομαλότητα με πριν. Αφού το h εμφανίζεται μέσα σε εκθετική συνάρτηση με μορφή τετραγώνου και υφίσταται υποτετραπλασιασμός των δεδομένων θα θεωρηθεί εμπειρικά μια τιμή διπλάσια της αρχικής π.χ για h=0.1 τώρα θα θεωρηθεί h=0.2. Ωστόσο πειραματικά διαπιστώνεται κάτι διαφορετικό.

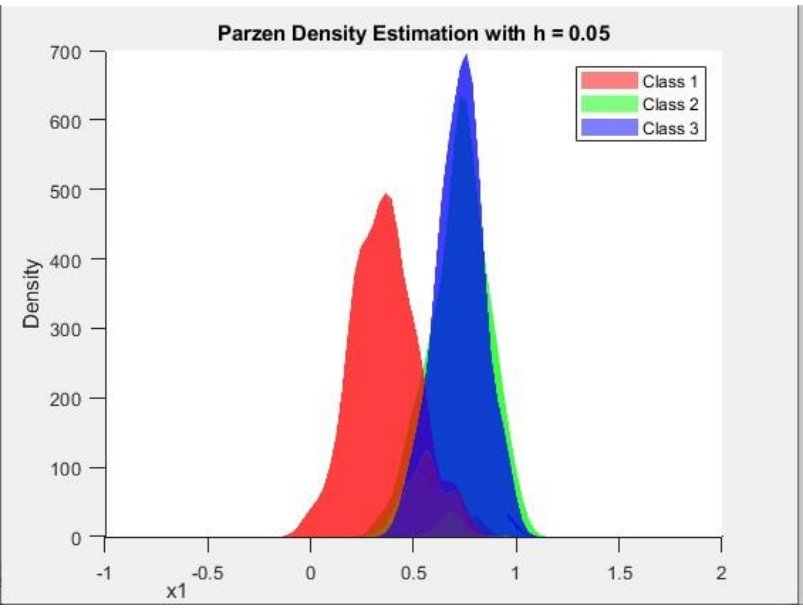


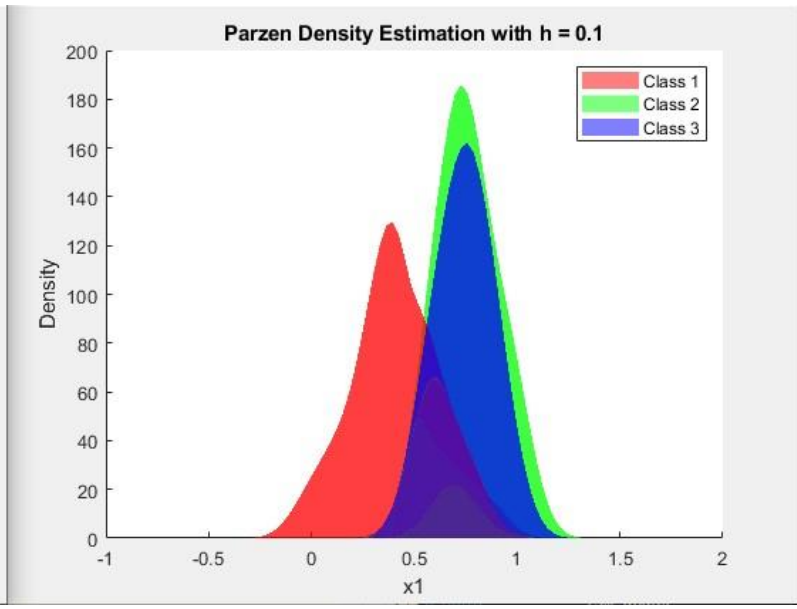


Αριστερά για το σύνολο του dataset τίθεται h = 0.1. Δεξιά για το 25% του dataset τίθεται το διπλάσιο h.

Παρόλαυτα βλέπουμε πως δεν είναι όμοιες οι κατανομές και η επίδραση διπλασιασμού του h είναι σαν να έχει εφαρμοστεί στο πρόβλημα που γίνεται χρήση του πλήρους dataset. Ο λόγος είναι επειδή τα δεδομένα είναι πυκνά διεσπαρμένα (συγκριτικά με τα h που χρησιμοποιούνται)και η κατανομή τους είναι προσεγγιστικά σαν μιας κανονικής κατανομής, για αυτό και με μικρότερο dataset μπορεί να απεικονιστεί με ίδιες παραμέτρους και ίδια ευκρίνεια τις εκτιμήσεις. Αν γινόταν χρήση άλλης συνάρτησης παραθύρου ίσως να μην παρατηρούνταν το ίδιο.

Για δοκιμές με μικρότερα h επαληθεύονται όλα τα παραπάνω και η αρχική θεωρητική κρίση.



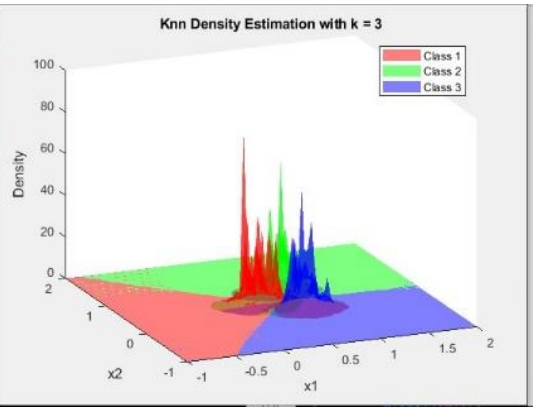


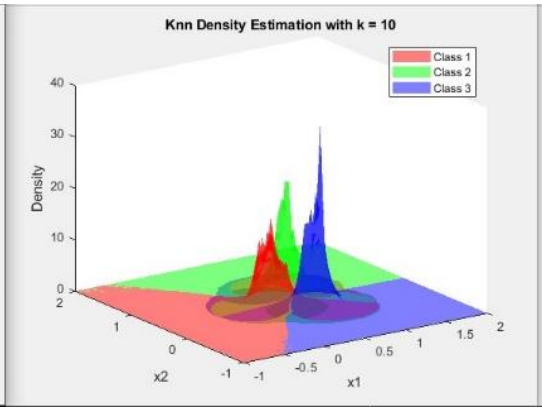
Τώρα γίνεται εκτίμηση των πυκνοτήτων πιθανοτήτων με τη χρήση του εκτιμητή KNN.

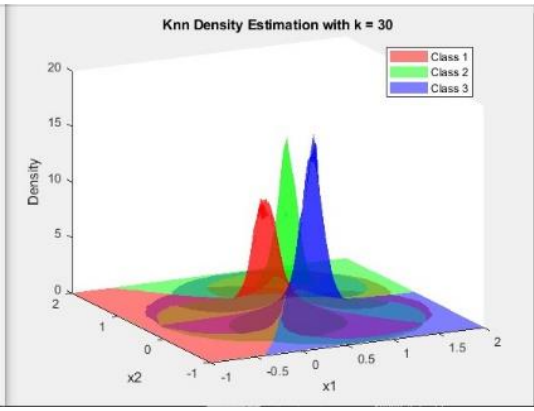
Η διαφορά με τη προηγούμενη τεχνική είναι πως τώρα αντί να υπάρχει προβληματισμός για την επιλογή καταλληλότερης συνάρτηση παραθύρου και το μήκος παραθύρου, έχουμε όγκους υπερσφαιρών με κέντρο το σημείο x στο οποίο γίνεται η προσέγγιση. Οι όγκοι αυτοί ορίζονται από τον αριθμό των δεδομένων που είναι επιθυμητό να επηρεάζουν την εκτίμηση. Επειδή τα δεδομένα είναι τυχαία κατανεμημένα ο όγκος των υπερσφαιρών μεταβάλλεται, ανάλογα τη διασπορά των διαθέσιμων δεδομένων περί τυχαίου σημείου x, άρα κατά κάποιον τρόπο ο όγκος V υπακούει σε συνάρτηση πλέον υπό συνθήκες (για να μπορεί ο εκτιμητής να συγκλίνει για άπειρα δεδομένα στη πραγματική κατανομή).

Βλέπουμε πως οι εκτιμήσεις είναι πιο αγκαθωτές και με μικρότερη διακύμανση σε σύγκριση με τον προηγούμενο εκτιμητή. Πάλι, αυξάνοντας το k, δηλαδή τον αριθμό των δεδομένων που εμπεριέχονται σε έναν όγκο με κέντρο το σημείο x στον δισδιάστατο χώρο, έχουμε μια

ομαλή επιφάνεια, η οποία είναι όμοια της κανονικής κατανομής. Αυτό υποδεικνύει πως και η αρχική εκτίμηση για τη συνάρτηση παραθύρου στον εκτιμητή Parzen ίσως να ήταν επιτυχημένη (δηλαδή θα επιτύγχανε γρήγορη σύγκλιση λόγω της φύσεως των δεδομένων).

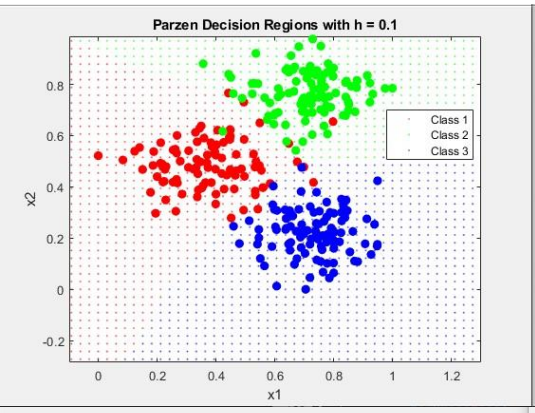


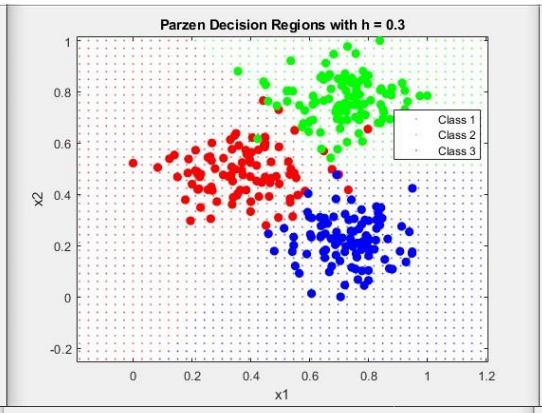


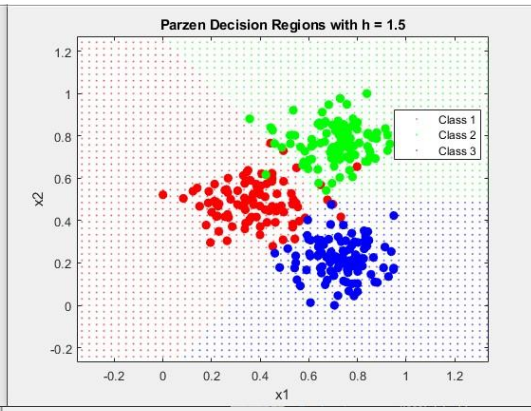


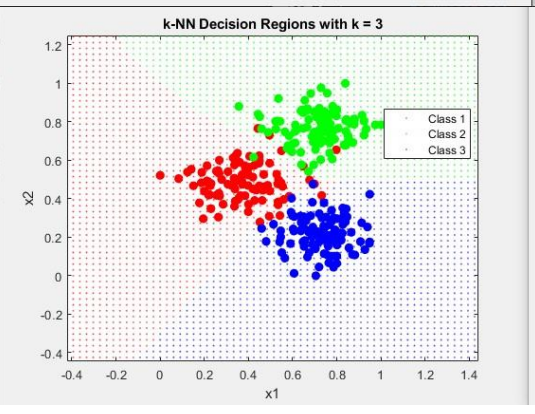
Για τη ταξινόμηση των δεδομένων, αφού είναι γνωστή η πληροφορία πως οι a priori πιθανότητες είναι ίσες, αρκεί να συγκριθούν οι τιμές των πυκνοτήτων που υπολογίστηκαν στα παραπάνω ερωτήματα. Για τη περίπτωση του KNN δεν είναι απαραίτητη η πληροφορία των a priori πιθανοτήτων διότι ο μαθηματικός φορμαλισμός του υποδεικνύει πως υπολογίζεται η από κοινού πιθανότητα όπου ισχύει $p(x,\omega_i) = p(x|\omega_i)P(\omega_i)$

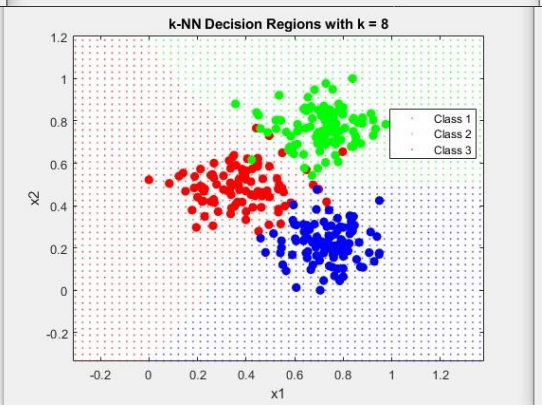
Παρακάτω παρατίθενται οι περιοχές απόφασης και για τους 2 εκτιμητές-ταξινομητές χρωματισμένες με το χρώμα της κλάσης τους.

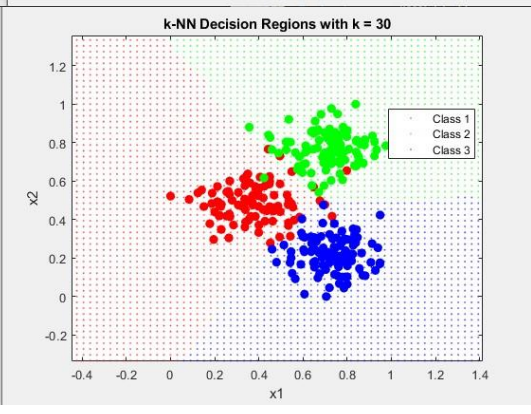












Για τον εκτιμητή Parzen, όπως ειπώθηκε και για το ερώτημα του μικρού dataset, διαπιστώθηκε πως και για μικρά h οι κατανομές μπορούν να προσεγγιστούν με αρκετά καλή ακρίβεια. Για αυτό τον λόγο φαίνεται και στα τρία διαγράμματα να είναι επαρκέστατες οι περιοχές απόφασης, ειδικά στις δύο τελευταίες που είναι σχεδόν πανομοιότυπες. Αυτό κρίνεται κυρίως παρατηρώντας τα όρια μεταξύ του κόκκινου-πράσινου και κόκκινου-μπλε ορίου που υπάρχουν μικρές επικαλύψεις. Ακόμα και για μικρά h φαίνεται αυτές οι επικαλύψεις να διευθετούνται επαρκώς.

Για τον εκτιμητή κ-NN φαίνεται για το k = 30 να διευθετούνται τα παραπάνω όρια το ίδιο καλά με τον προηγούμενο εκτιμητή. Αυτό ίσως να ήταν αναμενόμενο, με τη σκέψη πως μόνο για k=30 είχαμε όμοιες εκτιμήσεις για τις πυκνότητες πιθανότητας με τον εκτιμητή Parzen.

Όσον αφορά την σύγκριση των δύο μεθόδων, οι εκτιμητές k-NN και Parzen έχουν τα δικά τους πλεονεκτήματα και μειονεκτήματα. Ο k-NN είναι απλός και ευέλικτος, προσαρμόζεται καλά σε μη παραμετρικές κατανομές δεδομένων, καθώς δεν απαιτεί μοντελοποίηση ολόκληρης της πυκνότητας αλλά μόνο τις κοντινότερες γειτονιές του σημείου. Ωστόσο, χρειάζεται προσεκτική επιλογή του αριθμού γειτόνων k, είναι ευαίσθητος στον θόρυβο και τα δεδομένα με ανισόρροπες κλάσεις, και είναι απαιτητικός υπολογιστικά για μεγάλα σύνολα δεδομένων. Από την άλλη, η μέθοδος Parzen προσφέρει ακριβή εκτίμηση πυκνότητας για περίπλοκες κατανομές και η ποιότητα της εκτίμησης μπορεί να βελτιωθεί προσαρμόζοντας το πλάτος h, όπως και η ταχύτητα να εκτιμηθεί με χρήση του compact support. Παρόλα αυτά, είναι υπολογιστικά δαπανηρή, ειδικά για μεγάλα h και ορισμένες συναρτήσεις παραθύρου και απαιτεί προσεκτική επιλογή του h για να αποφευχθεί η υπερ- ή υπο-προσαρμογή της εκτίμησης. Συνολικά, ο k-NN είναι πιο άμεσος και απαιτεί λιγότερη ρύθμιση, ενώ ο Parzen είναι πιο ακριβής αλλά και πιο περίπλοκος σε προβλήματα μεγάλης κλίμακας.

Σε σύγκριση με γεωμετρικές τεχνικές όπως οι γραμμικοί διαχωριστές και τα SVMs, οι μέθοδοι Parzen και k-NN είναι πιο ευέλικτες και αποτελεσματικές σε περίπλοκες κατανομές. Ωστόσο, οι γεωμετρικές τεχνικές έχουν τα πλεονεκτήματα της ταχύτητας και της ανθεκτικότητας σε υπερπροσαρμογή (overfitting) , ειδικά όταν τα δεδομένα είναι γραμμικά ή περίπου γραμμικά διαχωρίσιμα. Αν και τα SVMs είναι πολύ αποδοτικά για δυαδικά προβλήματα, απαιτούν πρόσθετες στρατηγικές για προβλήματα πολλών κλάσεων, γεγονός που προσθέτει πολυπλοκότητα.

ΑΣΚΗΣΗ 2

Έχουμε ένα πρόβλημα δυαδικής ταξινόμησης και σκοπός είναι να χρησιμοποιηθούν τρεις γραμμικοί ταξινομητές, ο αλγόριθμος Batch Perceptron, Ho-Kashyap και SVM για την διεργασία αυτή.

Ο Batch Perceptron αξιοποιεί το Perceptron criterion function, έτσι ώστε η συνάρτηση αυτή να μην είναι ασυνεχής και μη παραγωγίσιμη (τμηματική για την πιο απλή περίπτωση που έχουμε σαν συνάρτηση κριτηρίου τον αριθμό των λάθος ταξινομημένων δεδομένων). Παραγωγίζοντας τη συνάρτηση στον χώρο, λαμβάνουμε την αναδρομική σχέση

$$\mathbf{a}(k+1) = \mathbf{a}(k) + \eta(k)\sum_{\mathbf{y}\in\mathcal{Y}_k} \mathbf{y},$$

α: Διάνυσμα με τα βάρη w

η: ρυθμός μάθησης

y : Feature vector δειγμάτων που ταξινομήθηκαν λάθος

Και τον επαναληπτικό αλγόριθμο που τερματίζεται είτε για κάποιο μέγιστο αριθμό επαναλήψεων είτε για κάποιο κατώφλι σφάλματος θ

Algorithm 3 (Batch Perceptron)

1 **begin initialize** $\mathbf{a}, \eta(\cdot), \text{criterion } \theta, k = 0$
2 **do** $k \leftarrow k + 1$
3 $\mathbf{a} \leftarrow \mathbf{a} + \eta(k) \sum_{\mathbf{y}\in\mathcal{Y}_k} \mathbf{y}$
4 **until** $\eta(k) \sum_{\mathbf{y}\in\mathcal{Y}_k} \mathbf{y} < \theta$
5 **return** a
6 **end**

Ο Ho Kashyap είναι αλγόριθμος που βασίζεται στη μέθοδο ελαχίστων τετραγώνων και μια τροποποιημένη έκδοση του gradient descend. Εδώ το b συμβολίζει το margin vector που συμβολίζει γεωμετρικά ένα περιθώριο-απόσταση μεταξύ των δειγμάτων και του διαχωριστικού υπερεπιπέδου και κάθε στοιχείο του είναι αυστηρά θετικό. Ο Y^+ είναι ο ψευδοαντίστροφος πίνακας του Y (ο Y περιέχει τα feature vectors όλων των δειγμάτων).

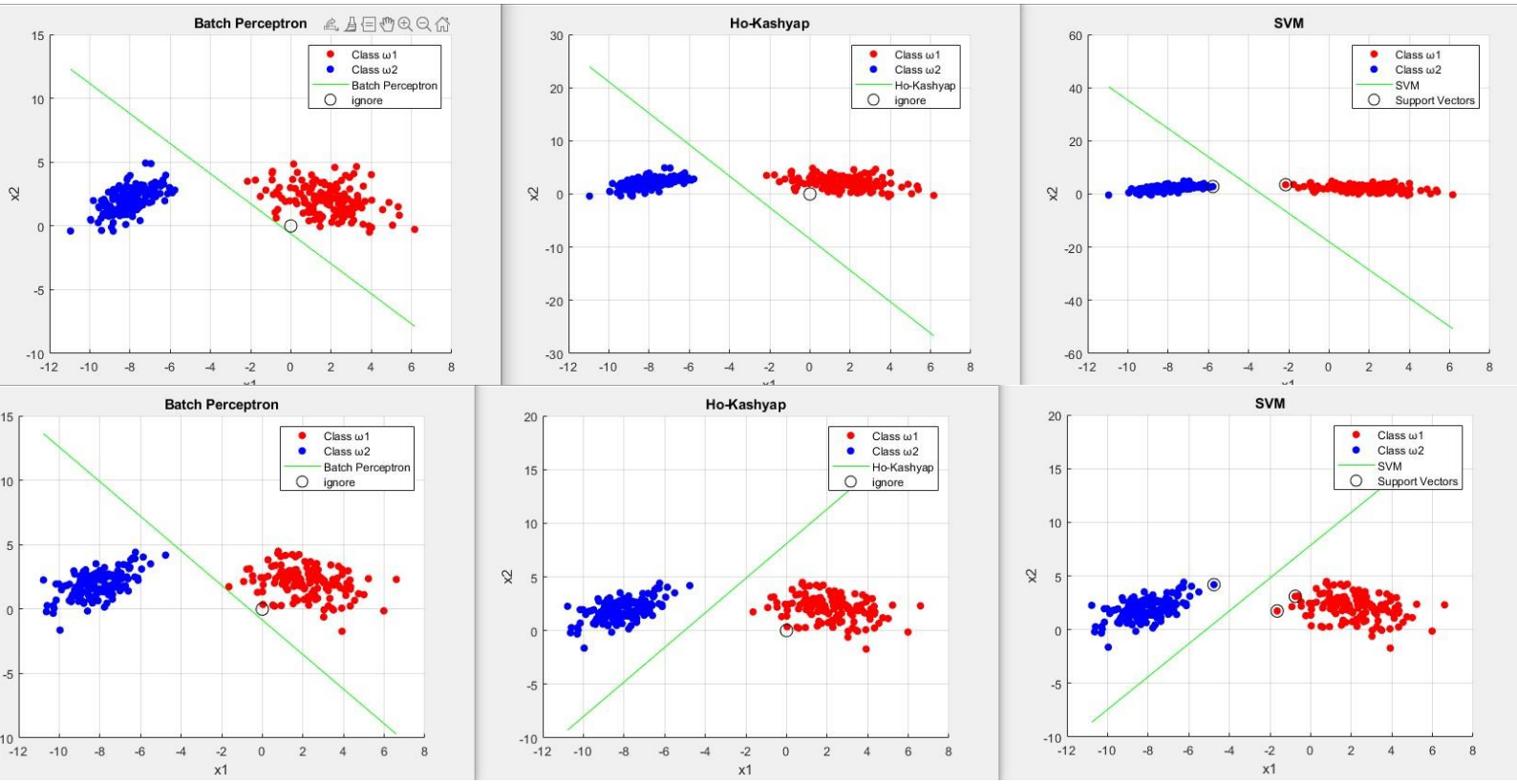
Algorithm 11 (Ho-Kashyap)

1 **begin initialize** $\mathbf{a}, \mathbf{b}, \eta(\cdot) < 1, \text{criteria } b_{min}, k_{max}$
2 **do** $k \leftarrow k + 1$
3 $\mathbf{e} \leftarrow \mathbf{Ya} - \mathbf{b}$
4 $\mathbf{e}^+ \leftarrow 1/2(\mathbf{e} + \text{Abs}[\mathbf{e}])$
5 $\mathbf{b} \leftarrow \mathbf{a} + 2\eta(k)\mathbf{e}^+$
6 $\mathbf{a} \leftarrow \mathbf{Y}^\dagger\mathbf{b}$
7 **if** $\text{Abs}[\mathbf{e}] \leq b_{min}$ **then return** $\mathbf{a}, \mathbf{b}$ *and* **exit**
8 **until** $k = k_{max}$
9 Print NO SOLUTION FOUND
10 **end**

$$\mathbf{Y}^\dagger \equiv (\mathbf{Y}^t\mathbf{Y})^{-1}\mathbf{Y}^t$$

Σημαντική σημείωση: στη γραμμή 5 του ψευδοκώδικα η σχέση είναι b(κ+1) = b(k) + 2η(κ)e^+, οπότε η φωτογραφία που είναι παρμένη από το βιβλίο του duda έχει τυπογραφικό λάθος πιθανότατα

Παρακάτω δίνονται αποτελέσματα για 2 διαφορετικά τυχαία set δεδομένων



Γενικά οι διαχωριστές SVM και Ho-Kashyap είναι αισθητά πιο απότομοι στη κλίση τους. Αυτό οφείλεται στο γεγονός ότι σαν παράμετρο λαμβάνουν το περιθώριο (margin), δηλαδή την ελάχιστη επιτρεπτή απόσταση από το κοντινότερο, σε αυτούς, σημείο. Αν και ο batch perceptron είναι ο πιο απλοϊκός αλγόριθμος και η ευθεία που δίνει να μην είναι η βέλτιστη, συγκλίνει συνήθως ταχύτερα από τα άλλα μοντέλα. Υπάρχουν φορές που ο Ho Kashyap είναι ταχύτερος αλλά αυτό εξαρτάται από τις τιμές αρχικοποίησης στις παραμέτρους του (μικρές ή μεγάλες τιμές για το b). Άρα συνοπτικά βλέπουμε ο SVM να δίνει το πιο απότομο και σταθερό-για-ταξινόμηση επίπεδο, δηλαδή με μεγαλύτερο περιθώριο, αλλά είναι πιο αργός λόγω της πολυπλοκότητάς του για την εύρεση βέλτιστου Convex. Ο Ho Kashyap δίνει ένα εξίσου σταθερό επίπεδο διαχωριστικότητας και είναι ταχύτερος, λύνοντας στη πραγματικότητα ένα σύστημα γραμμικών εξισώσεων. Ο Batch perceptron είναι ο πιο απλός αλγόριθμος στη σύλληψη και υλοποίηση του αλλά ο ασθενέστερος συγκριτικά με τους άλλους 2.

ΑΣΚΗΣΗ 3

Δίνεται ένα σύνολο δεδομένων που πρέπει να διαχωριστεί σε ποσοστά 50%, 25% και 25% σε train, validation & test set αντίστοιχα. Για να επιτευχθεί αυτό και να διατηρηθεί σε κάθε set η αναλογία του πλήθους δειγμάτων από κάθε κλάση με τυχαία επιλογή των στοιχείων κάθε κλάσης, για κάθε κλάση ορίζεται έναν πίνακας μήκους όσο και ο αριθμός των δειγμάτων της (έστω n) ο οποίος έχει κατά τυχαία σειρά μοναδικούς αριθμούς από το 1 έως και το n. Αυτός ο πίνακας θα λειτουργεί σαν μάσκα τυχαίου ανακατέματος κατά γραμμές, έτσι ώστε να πληρούνται οι παραπάνω προϋποθέσεις.

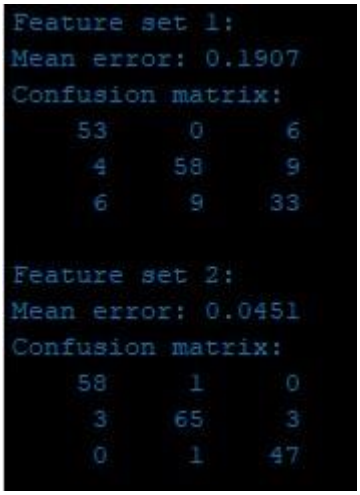
Για την εύρεση κατάλληλου αριθμού C ακολουθείται η shooting method, η οποία είναι ένας αλγόριθμος brute force για την εύρεση μιας παραμέτρου ή αρχικής συνθήκης. Σε αυτή τη περίπτωση αναζητείται η σταθερά C. Αρχικά, επειδή η επιθυμητή τάξη μεγέθους για την C ήταν άγνωστη, λήφθηκε ένα εύρος τιμών με λογαριθμικό διαμοιρασμό (logspace στο MATLAB) και από τα αποτελέσματα, έγινε αντιληπτό πως η C θα είναι ένας αριθμός το πολύ της τάξης του δεκαδικού. Έπειτα περιορίστηκε το εύρος τιμών που αναζητείται και έγινε η ίδια διαδικασία με γραμμικό διαχωρισμό. Για να γίνουν οι μετρήσεις σωστά χρησιμοποιήθηκε seed για την παραγωγή της ίδιας αλληλουχίας από ψευδοτυχαίους αριθμούς. Για διαδοχικές μετρήσεις με διαφορετικό διαμοιρασμό-ανακάτεμα απλά χρησιμοποιήθηκε διαφορετικό seed (για την m μέτρηση, seed(m)).

```
Best C: 0.117586
Test 1 Accuracy: 76.67%
Best C: 0.151758
Test 2 Accuracy: 83.33%
Best C: 0.063313
Test 3 Accuracy: 90.00%
Best C: 0.047232
Test 4 Accuracy: 80.00%
Best C: 0.023111
Test 5 Accuracy: 73.33%
Best C: 0.035172
Test 6 Accuracy: 73.33%
Mean error is: 0.205556
Standard Deviation of error is: 0.059056
```

```
Best C: 0.001000
Test 1 Accuracy: 60.00%
Best C: 0.001000
Test 2 Accuracy: 60.00%
Best C: 0.001000
Test 3 Accuracy: 60.00%
Best C: 0.001000
Test 4 Accuracy: 60.00%
Best C: 0.001000
Test 5 Accuracy: 60.00%
Best C: 0.001000
Test 6 Accuracy: 60.00%
Mean error is: 0.400000
Standard Deviation of error is: 0.000000
```

Αριστερά απεικονίζονται τα αποτελέσματα ακρίβειας για γραμμική συνάρτηση πυρήνα, ενώ δεξιά για RBF. Αν και τα αποτελέσματα για μη γραμμική συνάρτηση kernel είναι απογοητευτικά, είναι πιθανό να είναι απαραίτητος ο επαναπροσδιορισμός του εύρους τιμών για τη σταθερά C, καθώς με μια μικρή αύξηση παρατηρήθηκε σε ορισμένα tests η απόδοση να είναι καλύτερη. Πάντως από τα παραπάνω είναι εμφανές ότι το πρόβλημα προσεγγίζεται καλύτερα με τον γραμμικό SVM, καθώς με τον μη γραμμικό SVM σε περίπτωση γραμμικού προβλήματος υπάρχει αυξημένος κίνδυνος overfitting.

Για το πλήρες πρόβλημα έχουμε one vs one τεχνική και 5 cross validation. Έχουμε 2 set, training και validation. Με αυτήν την τεχνική, σε ανακατεμένο dataset, για κάθε run (έστω kth) χρησιμοποιούμε το kth μέρος του set για validation και τα υπόλοιπα δείγματα για εκπαίδευση. Όσον αφορά τη καταμέτρηση ψήφων έχουμε για το πρόβλημά μας 3 διαχωριστές (έναν για κάθε ζεύγος κλάσεων), όπου ο καθένας αποφασίζει για κάθε δείγμα σε ποια κλάση ανήκει. Η κλάση με τις περισσότερες ψήφους ενσωματώνει στη τάξη της το συγκεκριμένο δείγμα.



Κάνοντας χρήση όλων των χαρακτηριστικών ελαττώνεται κατά πολύ το σφάλμα. Λαμβάνοντας υπόψιν το πρώτο πίνακα

Όπου γίνεται χρήση των πρώτων 5 χαρακτηριστικών, βλέπουμε πιο έντονη τάση για ομοιότητα μεταξύ της 2^{ης} και 3^{ης} κλάσης αφού πιο συχνά μπερδεύονται.

Το ενδιαφέρον είναι ότι δεν είναι απόλυτα συμμετρικοί οι πίνακες και αυτό οφείλεται στο ότι κάθε ταξινομητής παρουσιάζει ανομοιότητες σε κάθε run που γίνεται στο k cross validation.