Αναγνώριση προτύπων
Εργασία 4
Σάββας Λιάπης 57403

Άσκηση 4.1

Η σ.π.π. μιας τυχαίας μεταβλητής δίνεται από την σχέση:

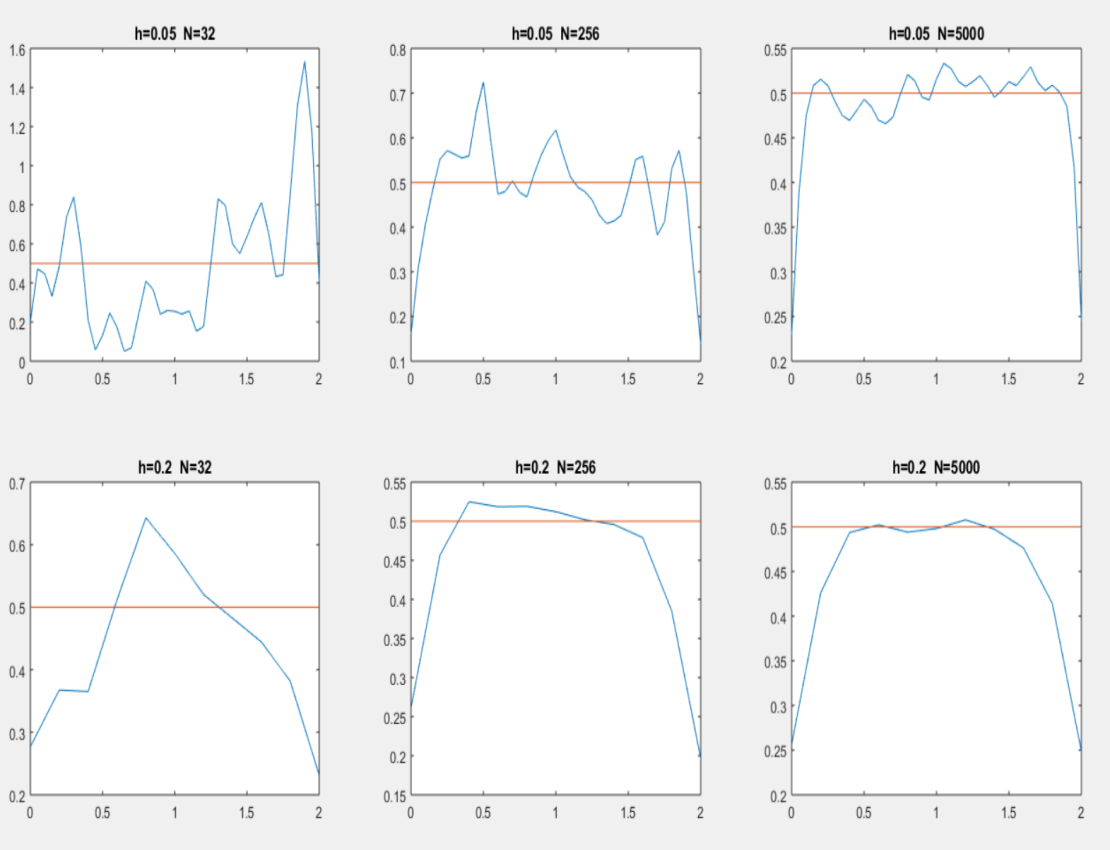
$$p(x) = \begin{cases} \frac{1}{2}, & 0 < x < 2 \\ 0, & αλλού \end{cases}$$

α. Χρησιμοποιείστε την μέθοδο παραθύρων Parzen για να την προσεγγίσετε χρησιμοποιώντας ως συνάρτηση πυρήνα την Gaussian N(0,1). Επιλέξτε την παράμετρο λείανσης να είναι h=0.05 και h=0.2 . Για κάθε περίπτωση, σχεδιάστε την εκτίμηση βασιζόμενοι σε N= 32, N=256 και N=5000 σημεία, τα οποία παράγονται από μία γεννήτρια ψευδοτυχαίων αριθμών, σύμφωνα με την p(x) .

β. Επαναλάβατε το προηγούμενο ερώτημα έχοντας N=5000 σημεία και χρησιμοποιώντας την εκτίμηση k-πλησιεστέρων γειτόνων με k=32, 64 και 256 αντιστοίχως.

Απάντηση 4.1(α):

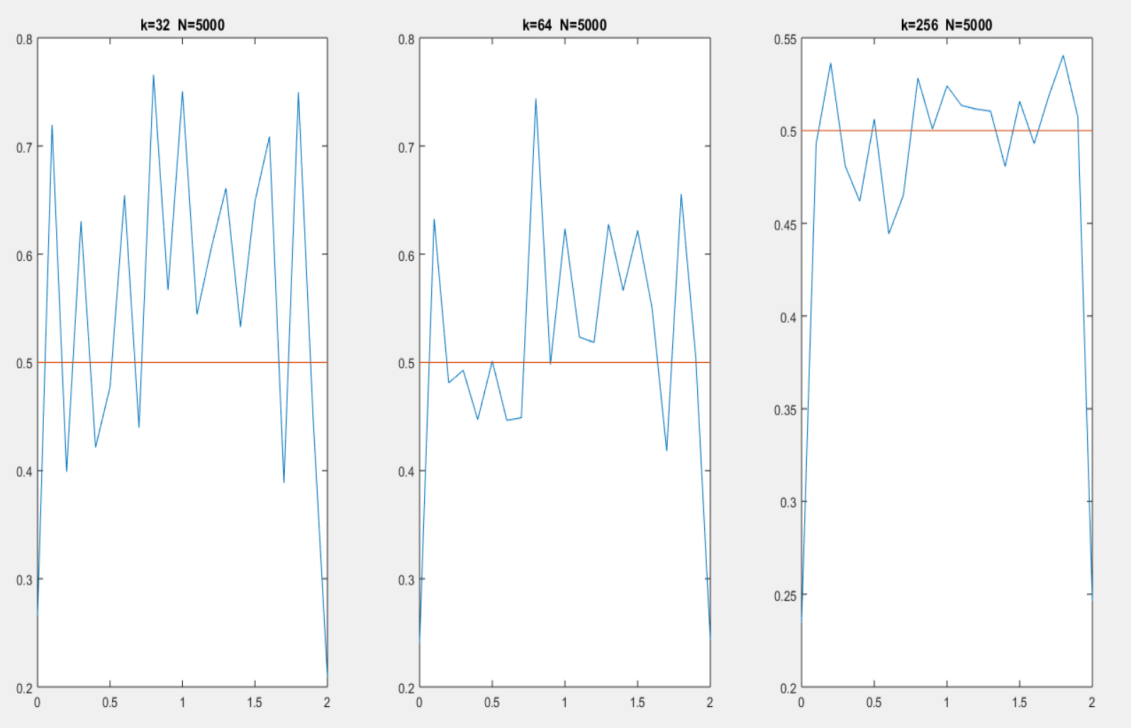
Δημιουργούμε την επιθυμητή θεωρητική pdf (ώστε να υπάρχει στο plot σαν μέτρο σύγκρισης) . Έπειτα, βασιζόμενη σε διαφορετικό αριθμό δειγμάτων και διαφορετικό εύρος παραθύρων κάθε φορά δημιουργούμε τις προσεγγίσεις της αρχικής pdf :



Παρατηρώντας τις γραφικές παραστάσεις μπορούμε εύκολα να καταλήξουμε στο συμπέρασμα ότι για h=0.2 και N=5000 παίρνουμε μια γραφική που μοιάζει σε μεγαλύτερο βαθμό με την ομοιόμορφη κατανομή. Αυτό περιμέναμε να δούμε γιατί γνωρίζουμε ότι όσο περισσότερα τα σημεία (N) τόσο περισσότερο η γραφική μοιάζει με την επιθυμητή γραφική. Επίσης βλέπουμε ότι για h=0.2 παίρνουμε ένα πιο επιθυμητό αποτέλεσμα από το h=0.05 και αυτό επιβεβαιώνει πάλι τις θεωρητικές μας προσδοκίες εφόσον δεν μελετάμε μία κατανομή με πολλές λεπτομέρειες δουλεύει καλύτερα η επιλογή μεγαλύτερων τιμών h (αν έχουμε μικρότερα h η προσέγγιση είναι πολύ θορυβώδης και χρειάζεται πολύ μεγάλο αριθμό N για να επιτύχουμε το επιθυμητό αποτέλεσμα)

Απάντηση 4.1(β):

Παρακάτω βλέπουμε τις προσεγγίσεις της pdf για διαφορετικούς αριθμούς knn γειτόνων .



Με την αύξηση του κ βλέπουμε ότι η προσέγγιση πλησιάζει την επιθυμητή pdf και επιβεβαιώνει τις θεωρητικές μας γνώσεις όπου θα περιμέναμε ότι για να πάρω καλύτερη προσέγγιση θα πρέπει να πάρω μεγαλύτερο πλήθος σημείων (επίσης για καλύτερη προσέγγιση πρέπει να μειώσουμε και το εμβαδόν της περιοχής που «σπάμε» για να μελετήσουμε την συνολική pdf (στον κώδικα είναι η μεταβλητή step)).

Άσκηση 4.2

Έστω πρόβλημα με 3 κλάσεις και με Priors:

- $P(x|\omega_1) = N(2, \quad 0.5), \quad P(\omega_1) = 0.5$
- $P(x|\omega_2) = N(1, \quad 1), \quad P(\omega_2) = 0.3$
- $P(x|\omega_3) = N(3, \quad 1.2), \quad P(\omega_3) = 0.2$

Α. Δημιουργήστε τυχαίο δείγμα $\{x_i\}$ 100 δειγμάτων (προτύπων) που να ακολουθεί τις παραπάνω κατανομές, σύμφωνα με τις δεδομένες εκ των προτέρων πιθανότητες. Χρησιμοποιήστε αυτό το δείγμα ως «σύνολο εκμάθησης». Δημιουργήστε τυχαίο δείγμα $\{y_i\}$ 1000 προτύπων που να ακολουθεί τις παραπάνω κατανομές, σύμφωνα με τις δεδομένες εκ των προτέρων πιθανότητες. Χρησιμοποιείστε αυτό το σύνολο ως σύνολο δοκιμής.

Β. Ταξινομήστε τα δείγματα $\{y_i\}$ χρησιμοποιώντας τον αλγόριθμο k-nearest-neighbor και υπολογίστε την πιθανότητα λάθους. Υλοποιήστε τον αλγόριθμο για k=1,2,3. Σχολιάστε τα αποτελέσματα σας. Συγκρίνετε την πιθανότητα λάθους με αυτή του Bayesian ταξινομητή.

- Προαιρετικό: Πως μπορείτε να υπολογίσετε ένα βέλτιστο k για το σύνολο $\{x_i\}$; Ποιο είναι αυτό;

Γ. Χρησιμοποιήστε τα παραπάνω δείγματα για κατηγοριοποίηση με Parzen Windows. Χρησιμοποιήστε τουλάχιστον 4 διαφορετικές τιμές της παραμέτρου λείανσης $h_n = \sigma$ ((spread) και διαλέξτε εκείνη που δίνει τα καλύτερα αποτελέσματα. Σχολιάστε το αποτέλεσμα.

- Προαιρετικό: Πως μπορείτε να υπολογίσετε ένα βέλτιστο h_n για το σύνολο $\{x_i\}$; Ποιό είναι αυτό;

Δ. Χρησιμοποιήστε τα παραπάνω δείγματα για κατηγοριοποίηση με Parzen Windows/ Probabilistic Neural Networks. Χρησιμοποιήστε 4 διαφορετικές τιμές του $h_n = \sigma$ (spread). Χρησιμοποιήστε την σχετική συνάρτηση του Neural Network Toolbox του Matlab® ή άλλη της επιλογής σας (στο λογισμικό που διανεμήθηκε υπάρχει και ο φάκελος parzenPNN με τα προγράμματα που περιγράφονται στις διαφάνειες 5b). Σχολιάστε τα αποτελέσματα. Συγκρίνετε με τον k-NN.

Απάντηση 4.2(Α):

Δημιουργούμε το σύνολο εκμάθησης x και το σύνολο δοκιμής y με τις παρακάτω εντολές :

```
traindata=100;
testdata=1000;
%the vectors containing the mean covariance and prior
avg=[2 1 3];
S=[0.5 1 1.2];
prior=[0.5 0.3 0.2];

%making the gauss distributions according to the data given
[x,trainclass]=generate_gauss_classes(avg,S,prior,traindata);
[y,testclass]=generate_gauss_classes(avg,S,prior,testdata);
```

Η συνάρτηση `generate_gauss_classes` παίρνει σαν ορίσματα τις μέσες τιμές, τις τιμές σ^2 ,τις prior πιθανότητες καθώς και το σύνολο των δειγμάτων που θέλουμε να πραχτούν και επιστρέφει ένα διάνυσμα με τις τιμές των δειγμάτων που έχουν παραχθεί και ένα διάνυσμα με την κλάση που ανήκει το κάθε δείγμα.

Απάντηση 4.2(Β):

Για να ταξινομήσουμε τα δείγματα χρησιμοποιούμε την συνάρτηση `k_nn_classifier`, που δίνεται στο eclass η οποία παίρνει σαν ορίσματα το σύνολο εκμάθησης x , την κλάση στην οποία αντιστοιχεί το κάθε στοιχείο και το σύνολο δοκιμής y που θέλουμε να ταξινομήσουμε με την μέθοδο knn.

```
errork1 =

    0.4680

errork2 =

    0.4010

errork3 =

    0.4350
```

Εφαρμόζοντας αυτή την μέθοδο για κ=1 κ=2 και κ=3 παίρνουμε το εξής αποτέλεσμα όσον αφορά το σφάλμα από την πραγματική ταξινόμηση. Δηλαδή για :

κ=1 : σφάλμα =0.468
κ=2 : σφάλμα =0.401
κ=3 : σφάλμα =0.435

```
bayeserror =

    0.3300
```

επίσης το σφάλμα ταξινόμησης κατά bayes είναι : 0.33

βλέπουμε ότι με κ=2 παίρνουμε μικρότερο σφάλμα από ότι κ=1και αυτό συμβαίνει γιατι μεγαλώνοντας το κ παίρνουμε καλύτερη εκτίμηση. Ωστόσο στην δική μας περίπτωση βλέπουμε ότι και κ=3 το σφάλμα είναι μεγαλύτερο από ότι για κ=2, παρόλα αυτά κ=2 και κ=3 είναι πολύ κοντινές τιμές και αυτό το αποτέλεσμα μπορεί να οφείλεται στα συγκεκριμένα δείγματα που πήραμε . Αν τρέξω τον κώδικα χωρίς την εντολή `randn('seed',0)` τις περισσότερες φορές παίρνω ότι το σφάλμα για κ=3 είναι μικρότερο από το σφάλμα για κ=2.

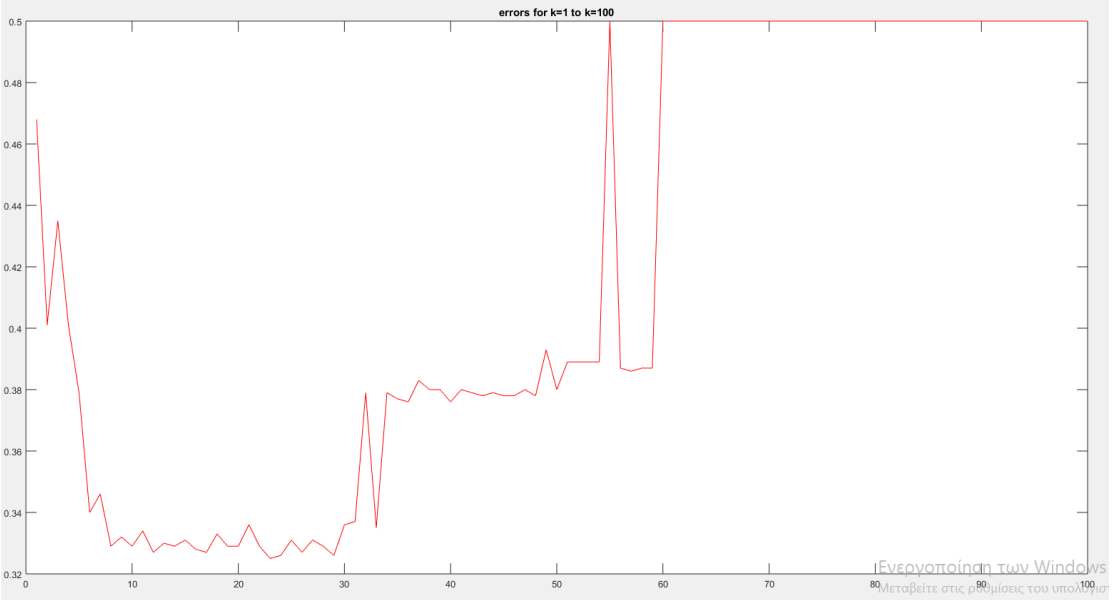
Απάντηση 4.2(Β(προαιρετικό)):

Ξέρουμε ότι όσο αυξάνεται το κ παίρνουμε ένα καλύτερο αποτέλεσμα. Μπορούμε να δώσουμε πολλές τιμές στο κ και να δούμε ποια δίνει σαν αποτέλεσμα το μικρότερο σφάλμα και να δούμε επίσης πως εξελίσσεται το σφάλμα για τις διάφορες τιμές του κ.
Στον κώδικά μας υπολογίζουμε το σφάλμα της κατηγοριοποίησης με knn αυξάνοντας τον κ σε κάθε κύκλο και παίρνουμε στο τέλος το ελάχιστο σφάλμα και το βέλτιστο κ (που μας δίνει αυτό το σφάλμα).

```
minerror =

    0.3250


koptimum =

    23
```

Ο αλγόριθμος μας δίνει ότι παίρνουμε ελάχιστο σφάλμα 0.325 για κ=23.

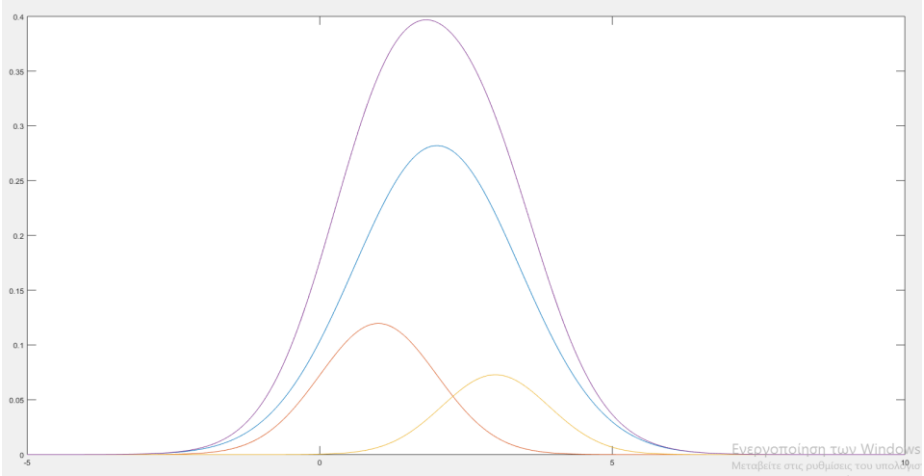
Για μικρό αριθμό κ είναι αυτονόητο ότι σίγουρα δεν μπορούμε να έχουμε αντιπροσωπευτικά αποτελέσματα για αυτό και πετυχαίνουμε μεγάλα σφάλματα. Για πολύ μεγάλα κ πάλι ο αλγόριθμος knn μπορεί να οδηγηθεί σε λανθασμένες αποφάσεις ταξινόμησης γιατί μία κλάση με μεγαλύτερο prior θα μπορεί να υπερισχύει πολύ πιο συχνά και σε περιπτώσεις που δεν πρέπει.



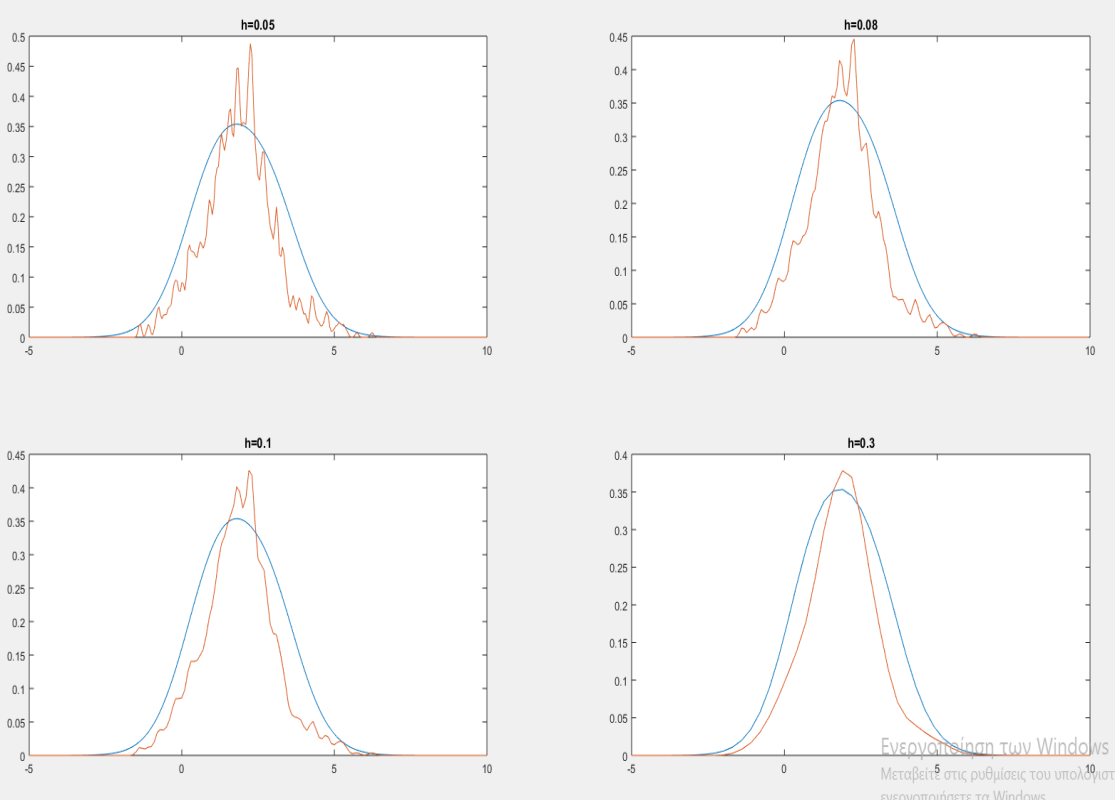
Εδώ φαίνεται πιο ξεκάθαρα αυτό που προσπαθώ να διατυπώσω με λέξεις. Όταν το κ γίνει πολύ μεγάλο ο ταξινομητής θεωρεί ότι όλα τα δείγματα ανήκουν στην κλάση 1 (στην οποία ανήκει το 50% των δειγμάτων) οπότε δίνει σφάλμα 50%.

Απάντηση 4.2(Γ):

Από τις 3 pdf παίρνουμε σαν συνισταμένη pdf την μωβ όπως βλέπουμε παρακάτω(αυτό δεν είναι απαραίτητο βήμα απλά επειδή η pdf μοιάζει σαν μία απλή κανονική κατανομή θεώρησα σκόπιμο να είναι ξεκάθαρο) :



Αν δούμε τα διαγράμματα για 4 τυχαίες τιμές του h=0.05 , h=0.08, h=0.1, h=0.3 παίρνουμε τα εξής :



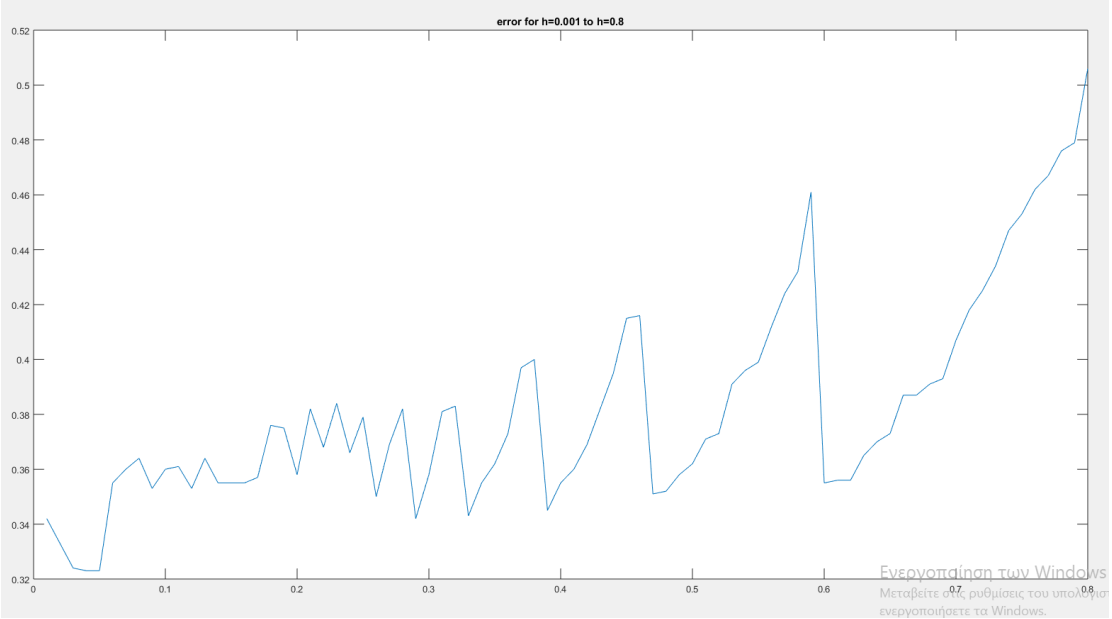
Παίρνοντας τις εικόνες για τις προσεγγιστικές pdf θα μπορούσε να μας δημιουργηθεί η εντύπωση ότι για h=0.3 παίρνουμε καλύτερα αποτελέσματα. Ωστόσο αν θυμηθούμε το πιο πάνω σχήμα όπου φαίνονται οι 3 pdf και η συνολική. Βλέπουμε ότι η κατανομή 1 καλύπτει εξ ολοκλήρου τις άλλες 2 οπότε σε σημεία οπού συνυπάρχουν 2 κλάσεις και ακόμη περισσότερο εκεί που συνυπάρχουν 3 κλάσεις (μεγάλη πυκνότητα δεδομένων και πιο μεγάλη πυκνότητα δεδομένων αντίστοιχα) θέλουμε μικρότερο παράθυρο. Οπότε το παράθυρο με h=0.05 δίνει τα καλύτερα αποτελέσματα από τις παραπάνω.

Απάντηση 4.2(Γ(προαιρετικό)):

Για αυτό το ερώτημα δημιουργήθηκαν οι pdf των επιμέρους κατανομών και έπειτα προσεγγίστηκαν με την μέθοδο parzen windows. Έπειτα από τις προσεγγίσιμες pdf εξήχθησαν οι μέσες τιμές και οι τιμές S τους. Στην συνέχεια για τις νέες μέσες τιμές πραγματοποιήθηκε ταξινόμηση κατά bayes και συγκρίνοντας την με την πραγματική ταξινόμηση καταλήγουμε στο σφάλμα. Αυτή η διαδικασία επαναλήφθηκε για 80 τιμές του h από 0.01 εώς 0.8 με βήμα 0.01. πήραμε τα εξής αποτελέσματα :

```
minerror =

    0.3230


hoptimum =

    0.0500
```

Το ελάχιστο σφάλμα είναι 0.323 και επιτυγχάνεται για h=0.05



Τώρα που ξέρουμε ότι το καλύτερο h είναι 0.05 μπορούμε να αυξήσουμε την ακρίβεια εποπτεύοντας την περιοχή μεταξύ 0.03-0.07 με βήμα 0.001 όπου παίρνουμε το αποτέλεσμα :

```
minerror =

    0.3200
```

δηλαδή ελάχιστο σφάλμα 0.32 για h=0.032

```
hoptimum =

    0.0320
```

Απάντηση 4.2(Δ):

Αυτό που κάνουμε εδώ συγκρίνουμε την είσοδο με κάθε όλα τα patterns και ψάχνουμε να βρούμε με ποιο μοιάζει (ελέγχω την διαφορά όλων των δειγμάτων με την είσοδο). Γενικά δεν συμφέρει υπολογιστικά. Το καλό με το pnn είναι ότι δεν επηρεάζεται από outliers.

```
pnnerrror =

    0.3670
```

Τρέχοντας τον αλγόριθμο καταλήγουμε σε σφάλμα : 0.3670. Αν το συγκρίνουμε με τον knn βλέπουμε ότι τα σφάλματα τους είναι πολύ κοντά (για δεδομένο dataset) με αυτό του pnn να είναι λίγο μεγαλύτερο. Σε αυτή την περίπτωση είναι προτιμότερη η μέθοδος knn γιατί όχι μόνο έχει μικρότερο σφάλμα (για κατάλληλο κ) αλλά απαιτεί και λιγότερη υπολογιστική ισχύ.