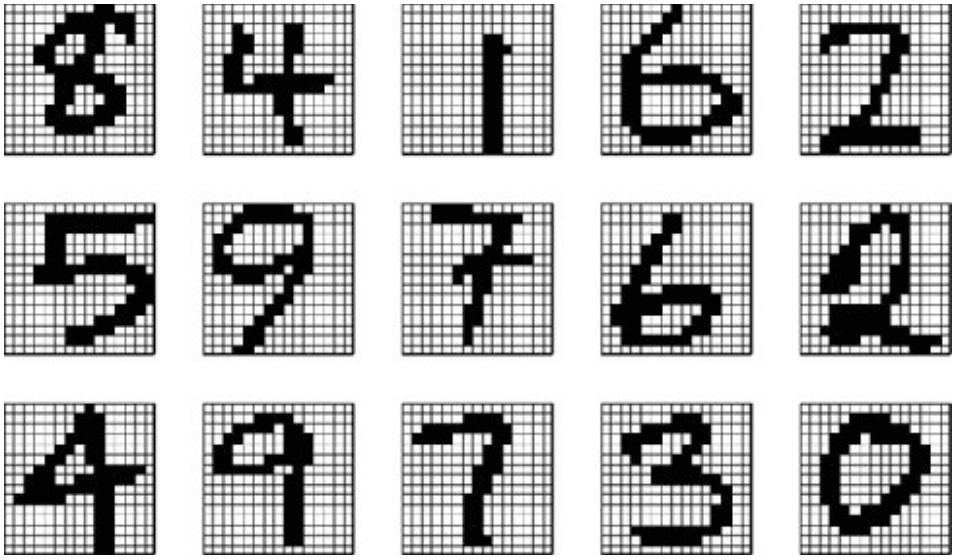
ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

2η Εργαστηριακή Άσκηση

Εξαγωγή χαρακτηριστικών απο φωνή για χρήση σε εφαρμογή αναγνώρισης

Μάθημα : Αναγνώριση Προτύπων



Ροή Σ

Συνεργάτες :

- Βαβουλιώτης Γεώργιος (**Α.Μ. : 03112083**)
- Σταυρακάκης Δημήτριος (**Α.Μ. : 03112017**)

Σκοπός: Σκοπός της άσκησης αυτής είναι η υλοποίηση ενός συστήματος επεξεργασίας και αναγνώρισης φωνής, με εφαρμογή σε αναγνώριση μεμονωμένων λέξεων. Αρχικά αυτό που θα κάνουμε είναι εξαγωγή κατάλληλων ακουστικών χαρακτηριστικών από φωνητικά δεδομένα. Τα εν λόγω χαρακτηριστικά είναι στην ουσία ένας αριθμός συντελεστών cepstrum που εξάγονται μετά από ανάλυση των σημάτων με μια ειδικά σχεδιασμένη συστοιχία φίλτρων (filterbank). Συγκεκριμένα μας δίνονται 133 .wav αρχεία(9 digits και 15 speakers per digit), μόνο που λείπουν 2 διότι θεωρήθηκαν προβληματικά. Επίσης θα πρέπει να επισημάνουμε οτι η διάρκεια των σημάτων αυτών διαφέρει γεγονός το οποίο έχει ληφθεί υπόψην στην υλοποίηση της άσκησης με την βοήθεια του Matlab.

Εκτέλεση Άσκησης

Θεωρώντας δεδομένα όλα όσα υλοποιήθηκαν και εξηγήθηκαν στην προπαρασκευή αυτού του εργαστηρίου θα συνεχίσουμε την επεξήγηση όσων κάναμε απο το Βήμα 10 μέχρι και το τέλος.

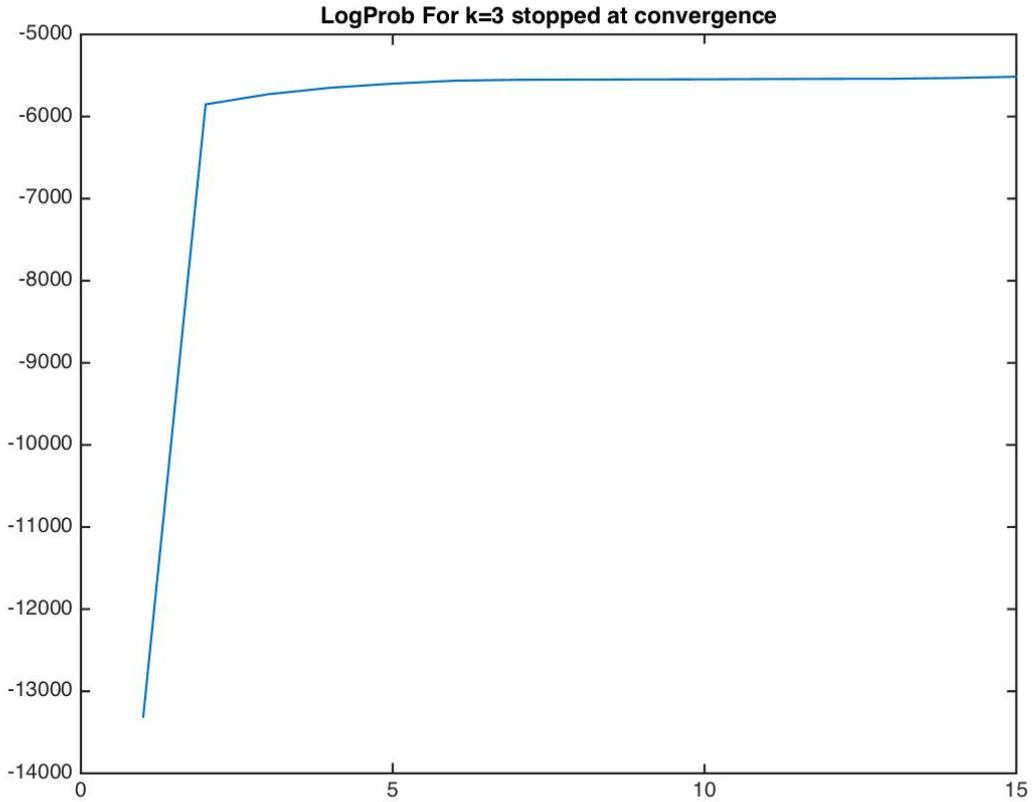
Βήμα 10 : Στο βήμα αυτό θα κάνουμε αρχικοποίηση των κρυφών Μαρκοβιανών μοντέλων. Αυτό που κάνουμε είναι να ορίσουμε ένα left-right μοντέλο για κάθε digit, το οποίο έχει Ns το πλήθος καταστάσεις, όπου Ns ανήκει στο διάστημα [5,9], όπως αναφέρεται και στην εκφώνηση της άσκησης. Επίσης ορίζουμε τον πίνακα μεταβάσεων A όπως ζητείται, δηλαδή αν A={aij}, τότε aij = 0 για j<i και j>i+1. Θα πρέπει να επισημάνουμε οτι οι τιμές του πίνακα A οι οποίες δεν είναι μηδενικές αρχικοποιούνται χρησιμοποιώντας την συνάρτηση mk_stohastic() η οποία κάνει τον πίνακα A στοχαστικό(το άθροισμα των στοιχείων κάθε γραμμής δίνει 1). Για να καταφέρουμε να έχουμε πάντα μια αρχική κατάσταση στην οποία θα μπορούμε να παραμείνουμε είτε να πάμε σε μια επόμενη κατάσταση, ορίζουμε τις αρχικές πιθανότητες των καταστάσεων μηδέν εκτός απο την αρχική κατάσταση.Τα διανύσματα ακουστικών χαρακτηριστικών που εξάγαμε στην προπαρασκευή της άσκησης θα τα χρησιμοποιήσουμε εδω σαν παρατηρήσεις. Για κάθε πλαίσιο φωνής θα έχουμε ενα διάνυσμα Ci(j), όπου j ανήκει στο {1,2,…,13} και αντιπροσωπεύει τους MFCC συντελεστές. Για την μοντελοποιήση των πιθανοτήτων των συντελεστών χρησιμοποιούμε κατανομές gauss αφού είναι επιτρεπτές οι συνεχείς μεταβολές(έχω Nm το πλήθος κατανομές για κάθε χαρακτηριστικό για καθεμία απο τις Ns καταστάσεις). Η αρχικοποίηση των κατανομών έγινε με χρήση της mixgauss_init().

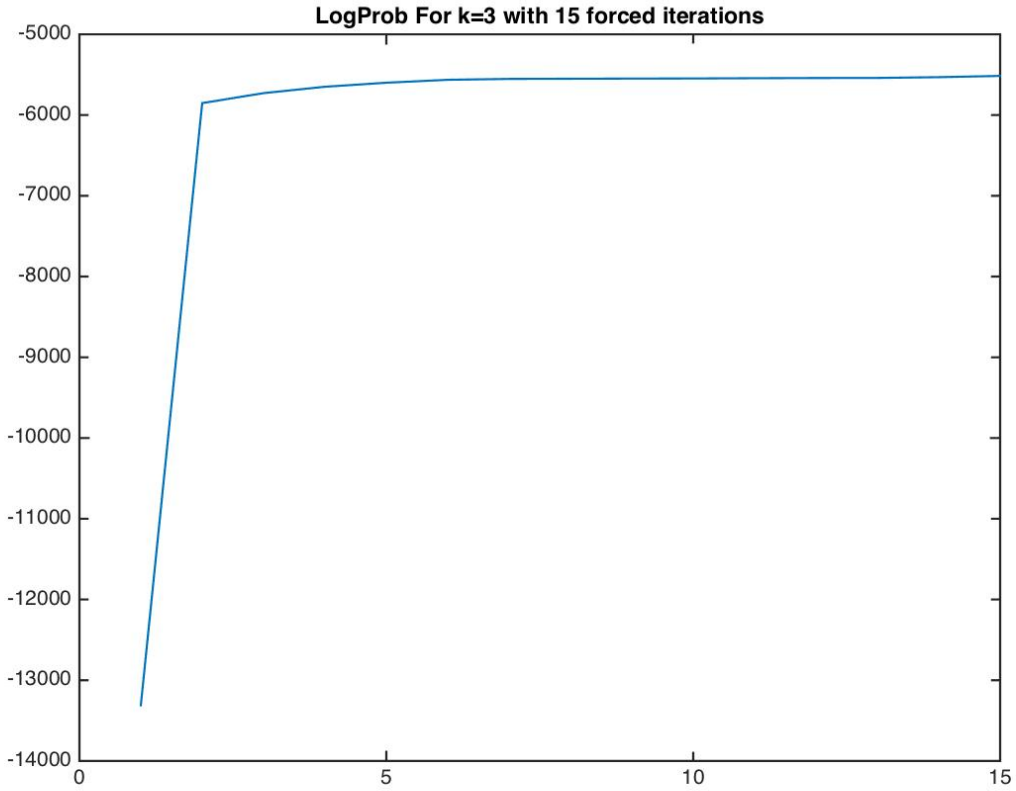
Βήμα 11 : Στο βήμα αυτό κάνουμε εκπαίδευση των 9 μοντέλων με χρήση του αλγορίθμου EM. Η Η σύγκλιση ελέγχεται μέσω της μεταβολής του log likelihood με μέγιστο πλήθος 10 επαναλήψεων. Είναι πολύ σημαντικό να αναφέρουμε οτι λόγω της τυχαίας αρχικοποίησης των μοντέλων η εκπαίδευση θα έδινε διαφορετικά αποτελέσματα αν γινόταν παραπάνω απο μια φορές. Αυτός είναι και ο λόγος για τον οποίο είναι πιθανό να πάρουμε κάποια άσχημα αποτελέσματα, αν γίνει μια κακή αρχικοποίηση. Ως δεδομένα εκπαίδευσης χρησιμοποιήθηκε το 70% των διαθέσιμων δεδομένων.

Βήμα 12 : Στο βήμα αυτό θα κάνουμε το testing, δηλαδή την αναγνώριση μεμονομένων ψηφίων. Για να το κάνουμε αυτό θα πάρουμε τους MFCC συντελεστές για τα test δεδομένα. Οι συντελεστές αυτοί χρησιμοποιούνται απο την συνάρτηση mhmm_logprob() 9 φορές, μία για κάθε μοντέλο. Η μέγιστη από τις 9 τιμές που υπολογίστηκαν αντιστοιχεί στο μοντέλο που είναι πιθανότερο να αντιπροσωπεύει το test ψηφίο, άρα το ψηφίο αυτό κατηγοριοποιείται στην κλάση του μοντέλου αυτού.

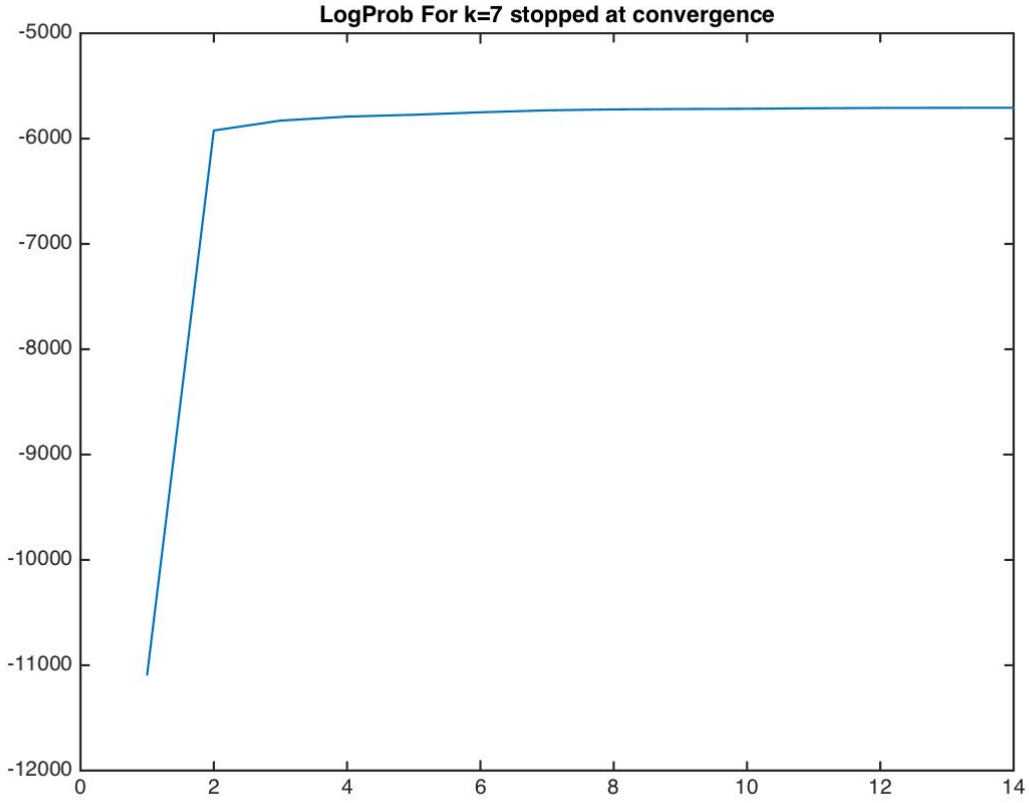
Βήμα 13 : Στο βήμα αυτό καλούμαστε να παραστήσουμε γραφικά την λογαριθμική πιθανοφάνεια ως συνάρτηση του πλήθους των επαναλήψεων. Στη συνέχεια φαίνονται τα αποτελέσματα(για κάθε K1 έχω 2 γραφήματα, το ένα αναγκάζει τον αλγόριθμο να κάνει 15 επαναλήψεις ακόμα και αν έχει πετύχει σύγκλιση και το άλλο σταματάει τον αλγόριθμο όταν πετύχουμε σύγκλιση και δεν κάνει περιττές επαναλήψεις):

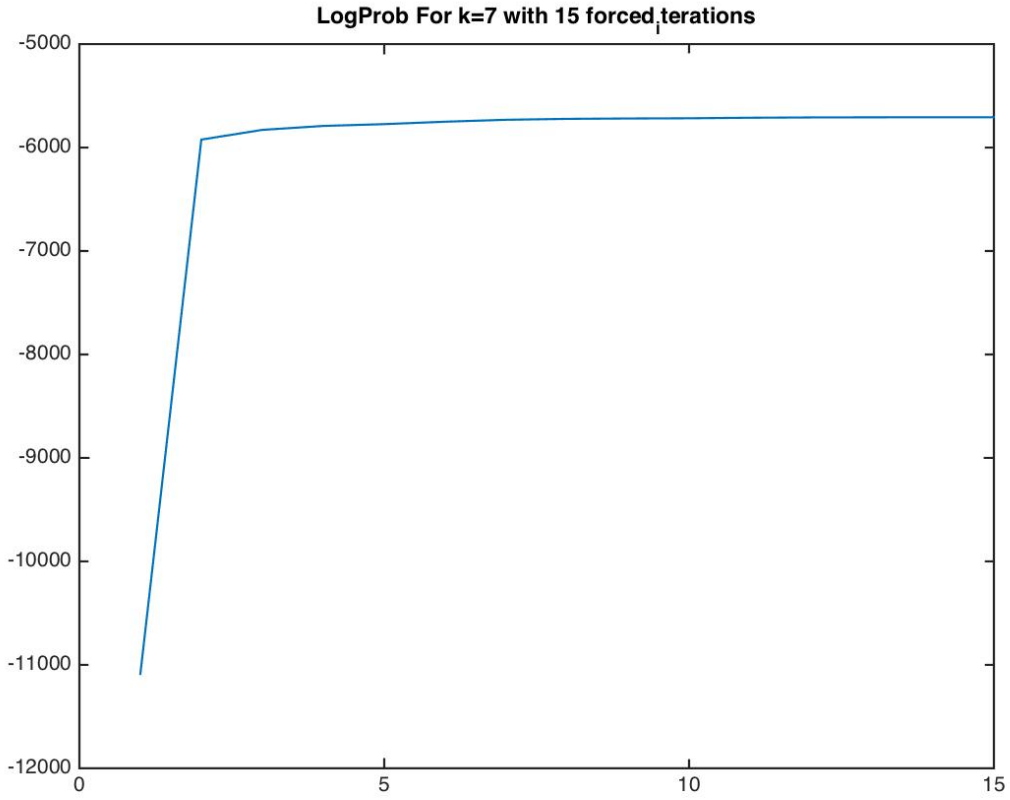
- K1 = 3 (Βαβουλιώτης ΑΜ : 03112083)





- K1 = 7(Σταυρακάκης ΑΜ : 03112017)



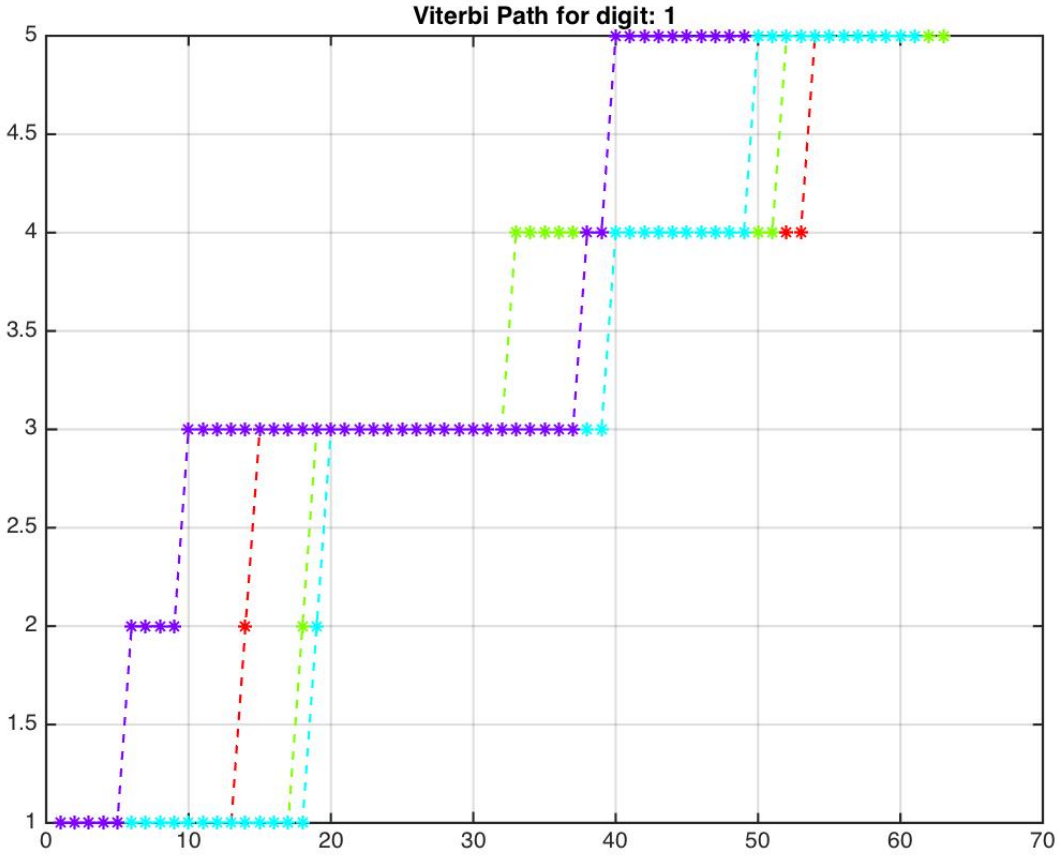


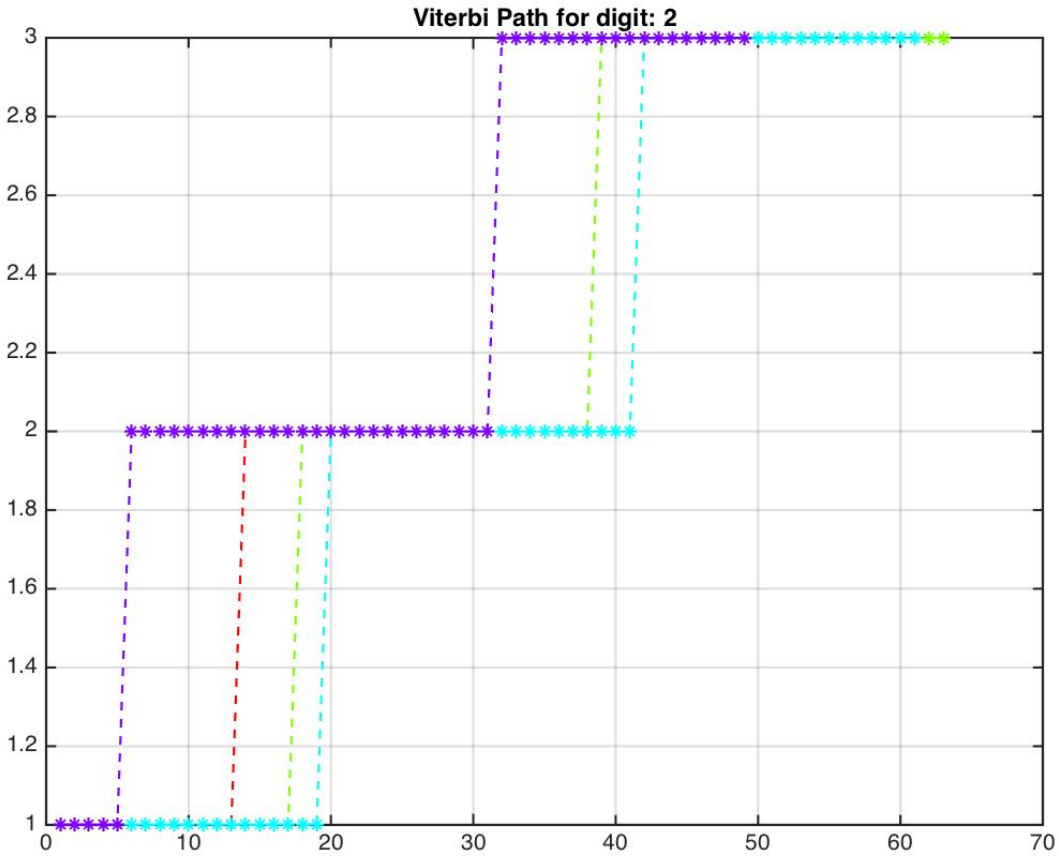
Βήμα 14 : Στο βήμα αυτό καλούμαστε να δημιουργήσουμε τον Confusion Matrix, ο οποίος θα περιέχει τα αποτελέσματα του testing. Πιο συγκεκριμένα ο πίνακας θα περιλαμβάνει στην γραμμή i και στη στήλη j άσσο αν και μόνο αν το test digit i έχει κατηγοριοποιηθεί στο μοντέλο που έχει το ψηφίο j. Προφανώς αν δεν έχει ασσο νεα κελί θα έχει μηδέν, άρα αν ο Confusion Matrix προκύψει διαγώνιος θα έχουμε πετύχει την καλύτερη δυνατή ταξινόμηση. Με την βοήθεια του Matlab πήραμε τελικά ποσοστό αναγνώρισης 80% την πρώτη φορά που το τρέξαμε και 91% περίπου τη δεύτερη φορά, τα οποία είναι ικανοποιητικά αποτελέσματα, αφού με μεγάλη πιθανότητα θα πετύχουμε σωστή κατηγοριοποίηση. Το total success ratio αλλάζει κάθε φορά που τρέχουμε τον κώδικα λόγω της τυχαιότητας που υπάρχει, εξαιτίας της αναλογίας train-test δεδομενων και των τυχαιων τιμων των πινακων του αλγοριθμου που προκύπτουν απο την rand(). Ενδεικτικά αποτελέσματα φαίνονται παρακάτω :

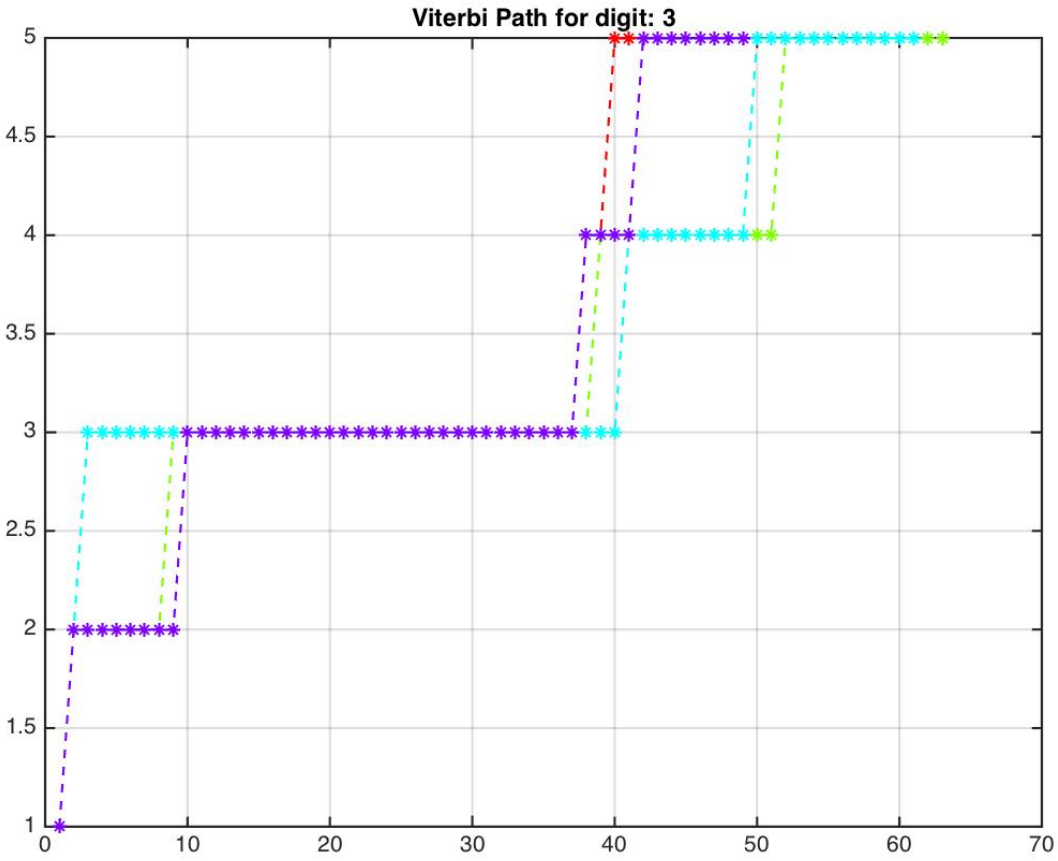
```
total_success_ratio =

    80
```

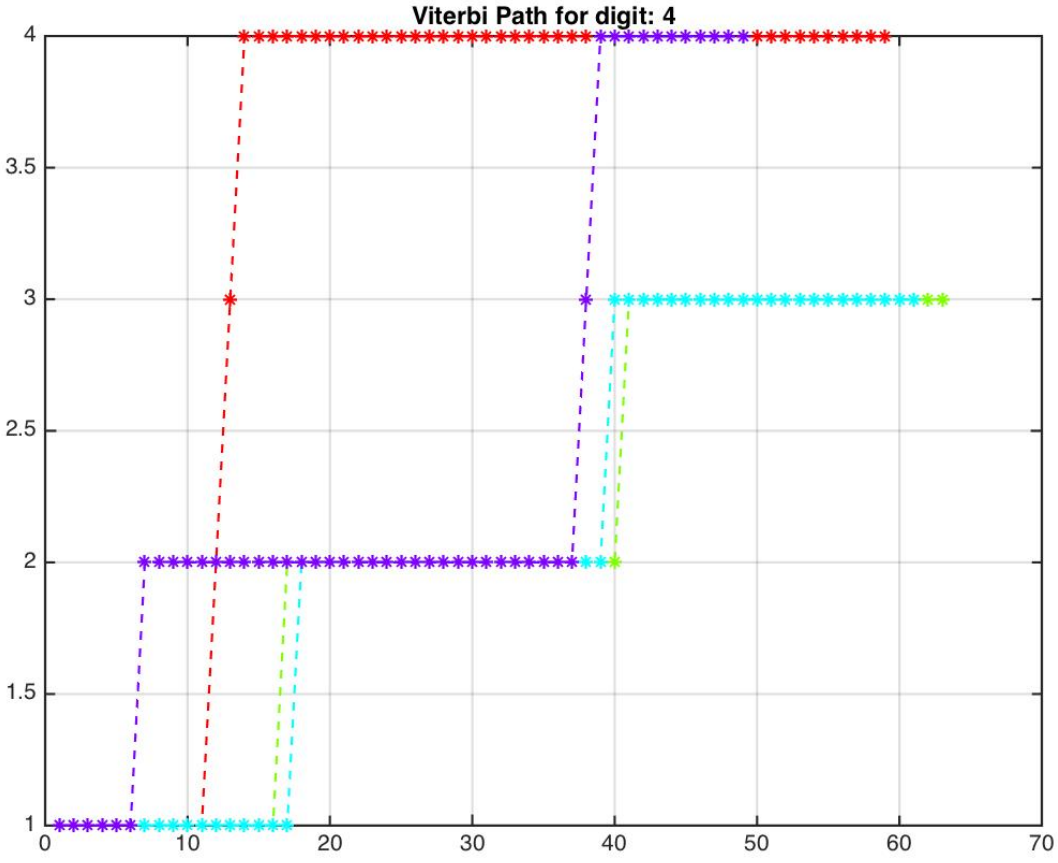
```
total_success_ratio =

   91.4286
```

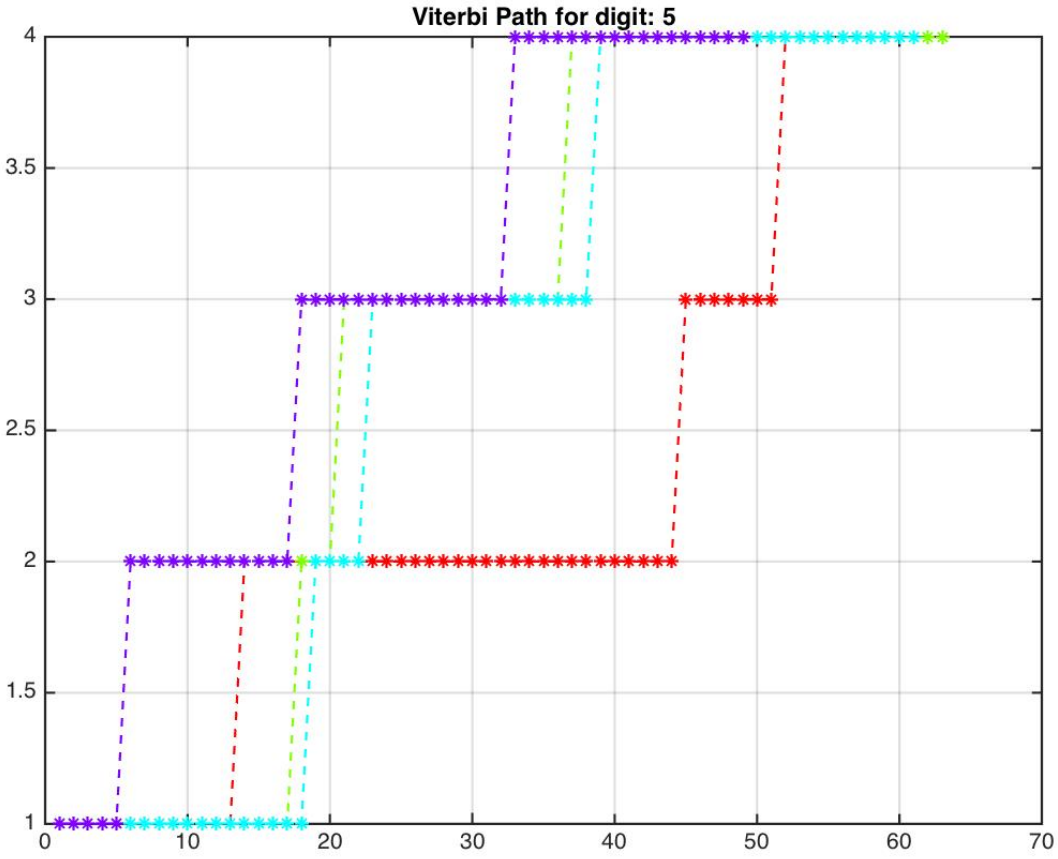
Βήμα 15 : Στο τελαυταίο βήμα αυτό που κάνουμε είναι να υπολογίσουμε τις παρατηρήσεις του πιο πιθανού μοντέλου με χρήση της συνάρτησης mixgauss_prob(). Χρησιμοποιώντας τις παρατηρήσεις αυτές, με τη βοήθεια του αλγορίθμου Viterbi υπολογίζουμε την πιο πιθανή ακολουθία καταστάσεων(με τη βοήθεια της συνάρτησης viterbi_path()). Κάθε τρέξιμο έδινε και άλλα viterbi paths που είναι λογικο λόγω της τυχαιοτητας του αλγοριθμου και ενδεικτικα παραθέτουμε τα παρακάτω αποτελέσματα:

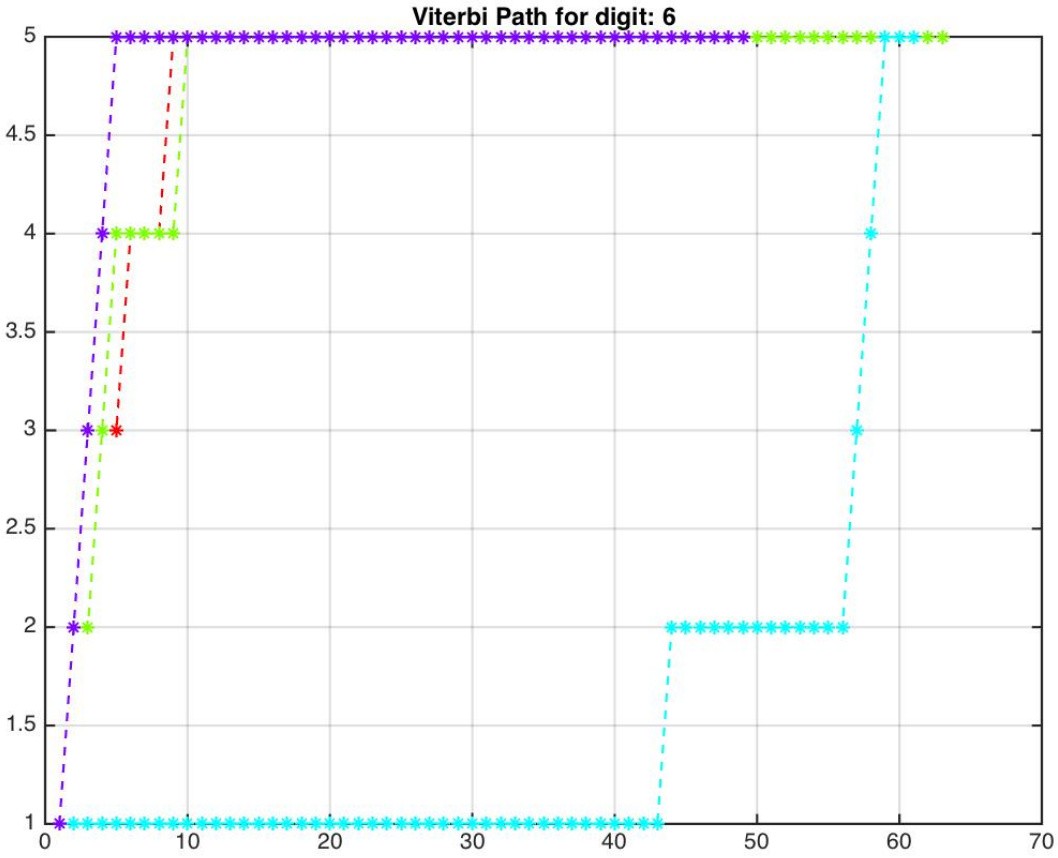


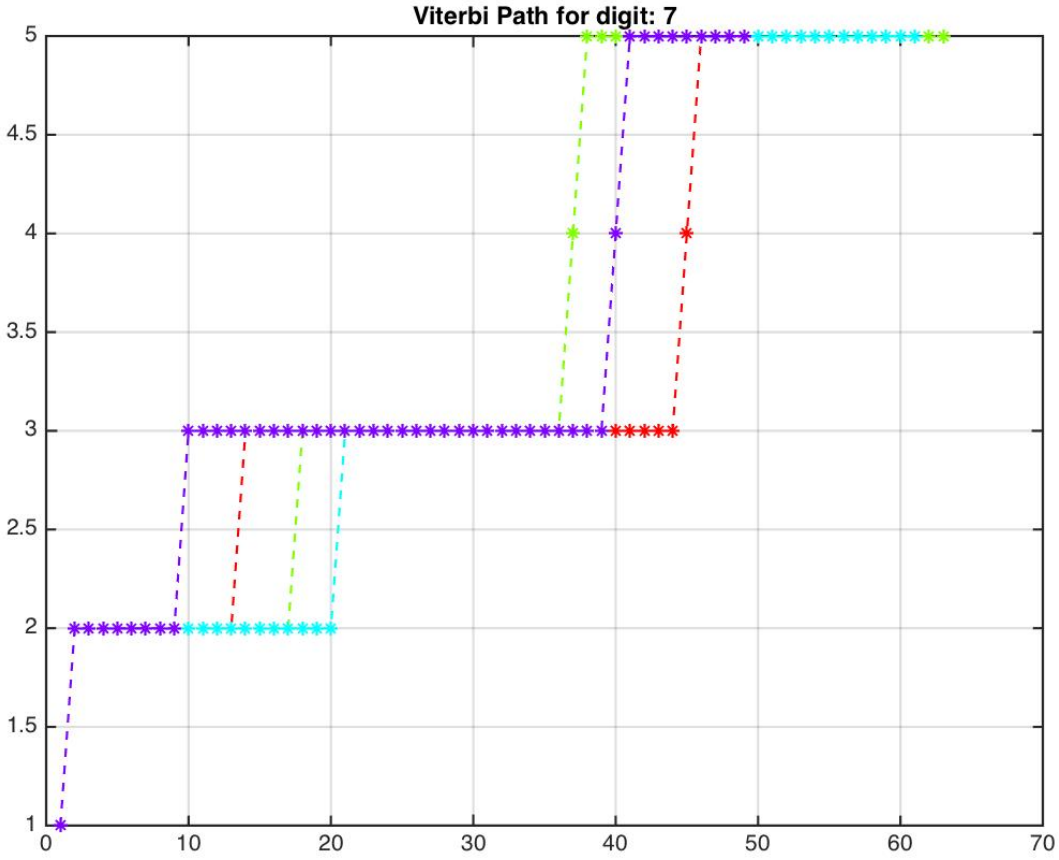


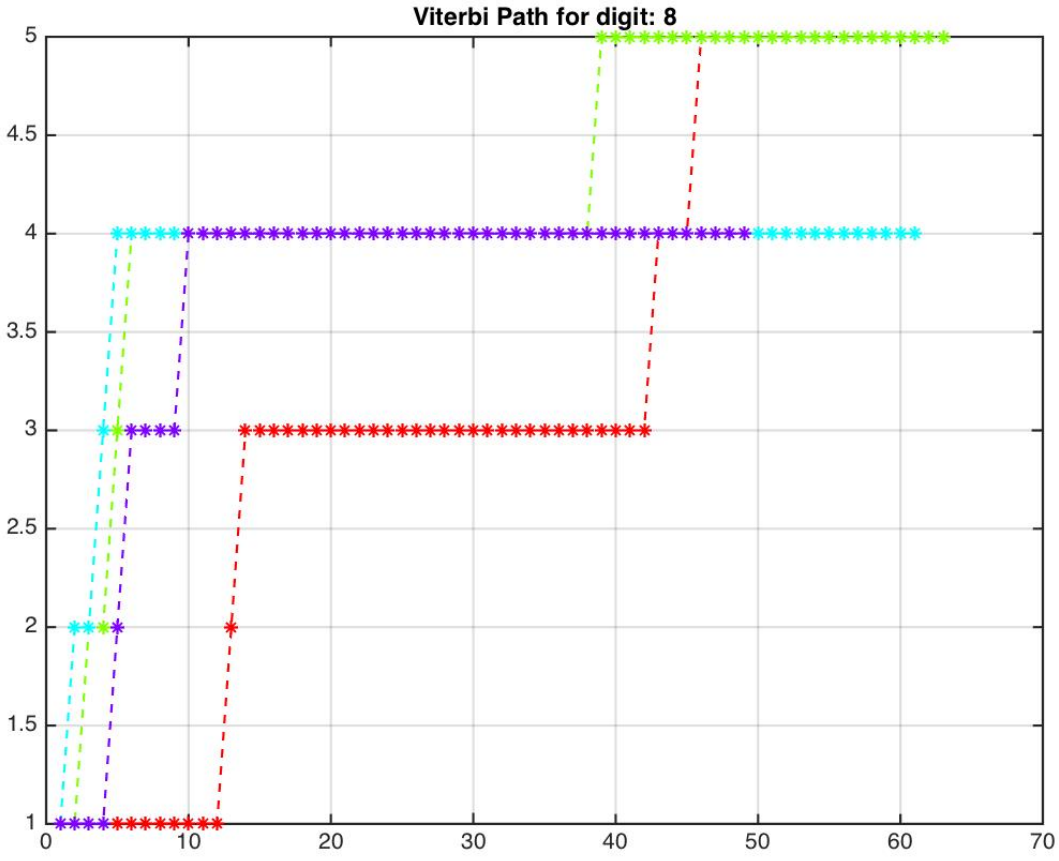

Viterbi Path for digit: 3
5
4.5
4
3.5
3
2.5
2
1.5
1
0
10
20
30
40
50
60
70

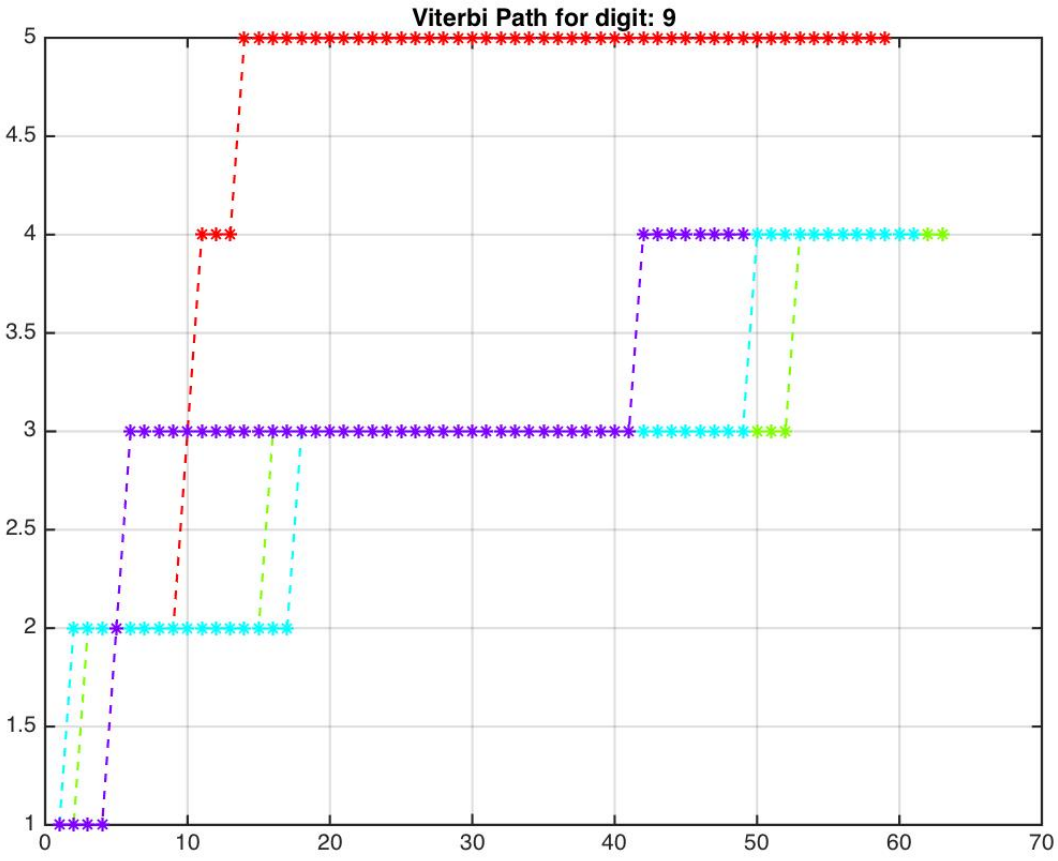

Viterbi Path for digit: 4
4
3.5
3
2.5
2
1.5
1
0
10
20
30
40
50
60
70


Viterbi Path for digit: 5
4
3.5
3
2.5
2
1.5
1
0
10
20
30
40
50
60
70


Viterbi Path for digit: 6
5
4.5
4
3.5
3
2.5
2
1.5
1
0
10
20
30
40
50
60
70


Viterbi Path for digit: 7
5
4.5
4
3.5
3
2.5
2
1.5
1
0
10
20
30
40
50
60
70


Viterbi Path for digit: 8
5
4.5
4
3.5
3
2.5
2
1.5
1
0
10
20
30
40
50
60
70


Viterbi Path for digit: 9
5
4.5
4
3.5
3
2.5
2
1.5
1
0
10
20
30
40
50
60
70