«ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ»

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ &
ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Αναγνώριση Προτύπων με Έμφαση στην Αναγνώριση Φωνής

2η Εργαστηριακή Άσκηση

9ο Εξάμηνο

Αθανασίου Νικόλαος

03112074

Σκοπός της 2ης εργαστηριακής άσκησης είναι να δημιουργήσουμε ένα σύστημα αναγνώρισης ψηφίων από ακουστικά δεδομένα.Στη συνέχεια της άσκησης επιχειρούμε από τους συντελέστες που εξάγαμε για κάθε πλαίσιο κάθε σήματος να κτασκευάσουμε ένα μοντέλο για αναγνώριση φωνής με τη βοήθεια κρυφών μαρκοβιανών μοντέλων τα οποία αρχικά εκπαιδεύουμε με το 70% των δεδομένων, ελέγχουμε τα αποτελέσματα της εκπαίδευσης (30% των δεδομένων) και τέλος επιδεικνύουμε για κάθε ψηφίο τα πιθανότερα μονοπάτια που ακολούθησε -ακολουθία καταστάσεων κρυφού μαρκοβιανού- με τη βοήθεια του αλγόριθμου viterbi .

Βήμα 10

Στο βήμα αυτό αρχικοποιούμε τα κρυφά μαρκοβιανά μοντέλα που θα χρησιμοποιήσουμε.Ορίζουμε τις καταστάσεις οι οποίες τοποθετούνται στο εύρος [5,9] .Επιλέγεται ο αριθμός τους να είναι 5 καθώς παρήγαγε τα καλύτερα αποτελέσματα που είναι λογικό αφού οι εναλλαγές σε κάθε φωνητικό σήμα είναι λίγες και εντοπίζονται κυρίως στην αρχή τη μέση και το τέλος κάθε φωνήματος άρα και μικρότερος να ήταν ο αριθμός δεν θα είχαμε αρνητικές μεταβολές στα αποτελέσματα.Ο πίνακας μεταβάσεων παίρνει αρχικά τυχαίες τιμές(*rand()*) στη συνέχεια μηδενίζονται όλα τα στοιχεία του πέρα από την διαγώνιο(παραμονή στην ίδια κατάσταση) και την πάνω απ' αυτή ψευδοδιαγώνιο(μετάβαση σε διαδοχική μόνο) και τέλος με τη βοήθεια της *mk_stohastic* κανονικοποιείται ώστε να γίνει στοχαστικός δηλαδή $0 \leq a_{ij} \leq 1$ και $\sum_{j=1}^{\#\,states} a_{ij}\ , \forall\, i : i \in \{1, .., \#states\}$.Ξεκινάμε πάντα από την "αρχική" κατάσταση($P(q_0) = 1\ P(q_i) = 0\ \forall i \setminus \{0\}$).Οι παρατηρήσεις του μοντέλου – αφού πρόκειται για κρυφό μαρκοβιανό- είναι οι 13 συντελέστες που εξάγαμε από κάθε σήμα φωνής κατά την προπαρασκευή της άσκησης των οποίων μοντελοποιούμε την πιθανότητα με μείγμα γκαουσσιανών με τη βοήθεια της *mixgauss_init()* με τον αλγόριθμο **kmeans**.

Βήμα 11

Στη συνέχεια χρησιμοποιώντας τη συνάρτηση *mhmm_em()* εκπαιδεύουμε το μοντέλο μας με τη βοήθεια του αλγορίθμου EM δίνοντας τους ως παραμέτρους αυτά που υπολογίστηκαν στο Βήμα 10 και αρχικές εκτιμήσεις για τη μέση τιμή και τη διασπορά κάθε ψηφίου.Αξίζει να σημειωθεί ότι τα αποτελέσματα του συγκεκριμένου και των παρακάτω βημάτων ενδέχεται να ποικίλλουν αφού στο βήμα 10 οι αρχικοποιήσεις του πίνακα μεταβάσεων και του πίνακα *mixmat()* που δίνονται ως παράμετροι στην *mhmm_em()* γίνονται

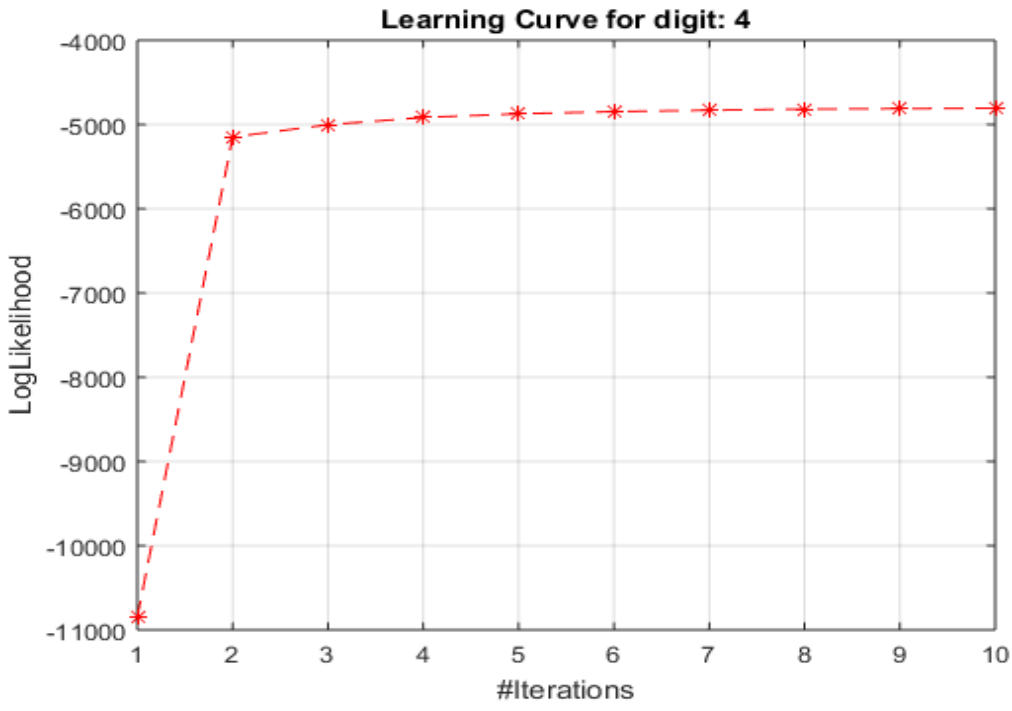
τυχαία.Τέλος ως αποτέλεσμα της παραπάνω συνάρτηση έχουμε 9 εκτιμήσεις μία για κάθε ψηφίο.Γίνεται reshape σε κάποια αποτελέσματα λόγω των απαιτήσεων της *em_hmm()* στο πως δέχεται και επεξεργάζεται τις παραμέτρους.

Βήμα 12

Στο συγκεκριμένο βήμα επιλέγουμε το 30% των δεδομένων δηλαδή 4 ομιλητές για κάθε ψηφίο ώστε να ελέγξουμε(testing) τα αποτελέσματα της εκπάιδευσης.Αρχικά εξάγουμε τα χαρακτηριστικά για καθένα από τους 4 ομιλητές για κάθε ψηφίο (απλώς γίνεται load του πίνακα C της προπαρασκευής και υπάρχουν ήδη αλλά παρατίθεται σε σχόλια ο απαραίτητος κώδικας).Έπειτα δίνουμε στη συνάρτηση *mhmm_logprob()* για κάθε ψηφίο και test εκφώνηση τα δεδομένα αυτά μαζί με τα υπόλοιπα αποτελέσματα της *mhhm _em ()* για κάθε ψηφίο και συγκεκριμένο ομιλήτη (εκτός της Ε.Μ.Π. που μας επέστρεψε για τους "train" ομιλητές σε κάθε ψηφίο) και εκείνη μας επιστέφει μια εκτίμηση για κάθε ψηφίο από τις οποίες επιλέγουμε τη μεγαλύτερη αφού αυτή είναι η καλύτερη εκτίμηση για το που ανήκει το συγκεκριμένο ψηφίο.Τέλος, με την παραπάνω μέθοδο κατατάσσουμε τις εκφωνήσεις των test δεδομένων σε μία από τις 9 κατηγορίες.

Βήμα 13

Η λογαριθμική πιθανοφάνεια για το ψηφίο 4 όπως υπολογίστηκε παρατίθεται στη συνέχεια:

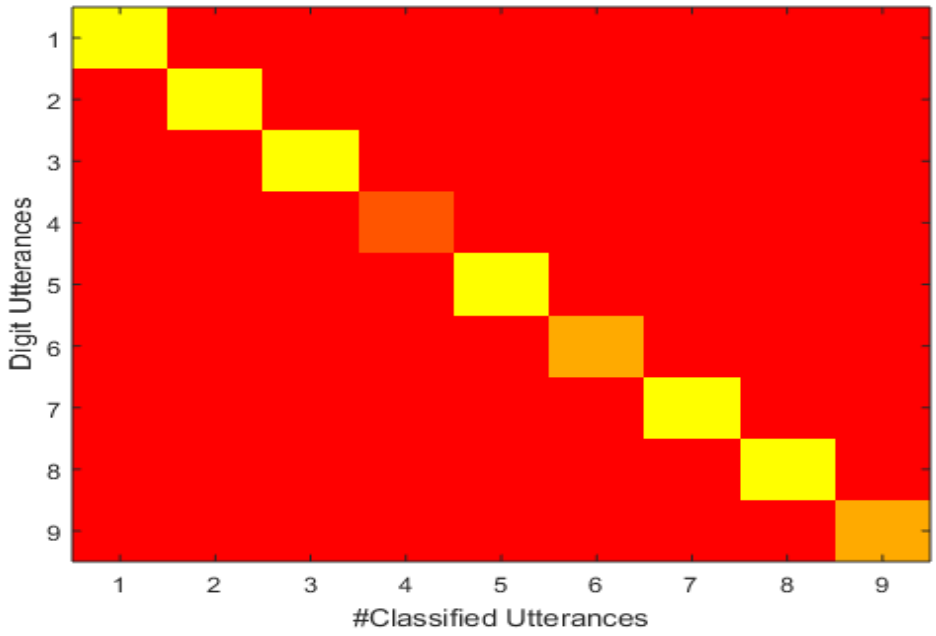


Βήμα 14

Από το αποτέλεσμα του βήματος 13 προκύπτει ένα πίνακας 9x(#test data=4) που το i,j στοιχείο του περιέχει την κατηγορία που εντάχθηκε η j-οστή εκφώνηση του i-οστού ψηφίου.Από τον πίνακα αυτόν σχηματίζουμε τον Confusion Matrix ο οποιος είναι ένα 9x9 πίνακας που στο i,j κελί του περιέχει πόσες φορές το ψηφίο i μπήκε στην κατηγορία j.Στην περίπτωση δηλαδή της ιδανικής κατηγοριοποίησης ο πίνακας αυτό προκύπτει διαγώνιος. Τέλος,υπολογίζεται το ποσοστό επιτυχίας με τον τρόπο που υποδεικνύεται στην εκφώνηση μέσω του Confusion Matrix και προκύπτει:

Success Rate: 91.43 %

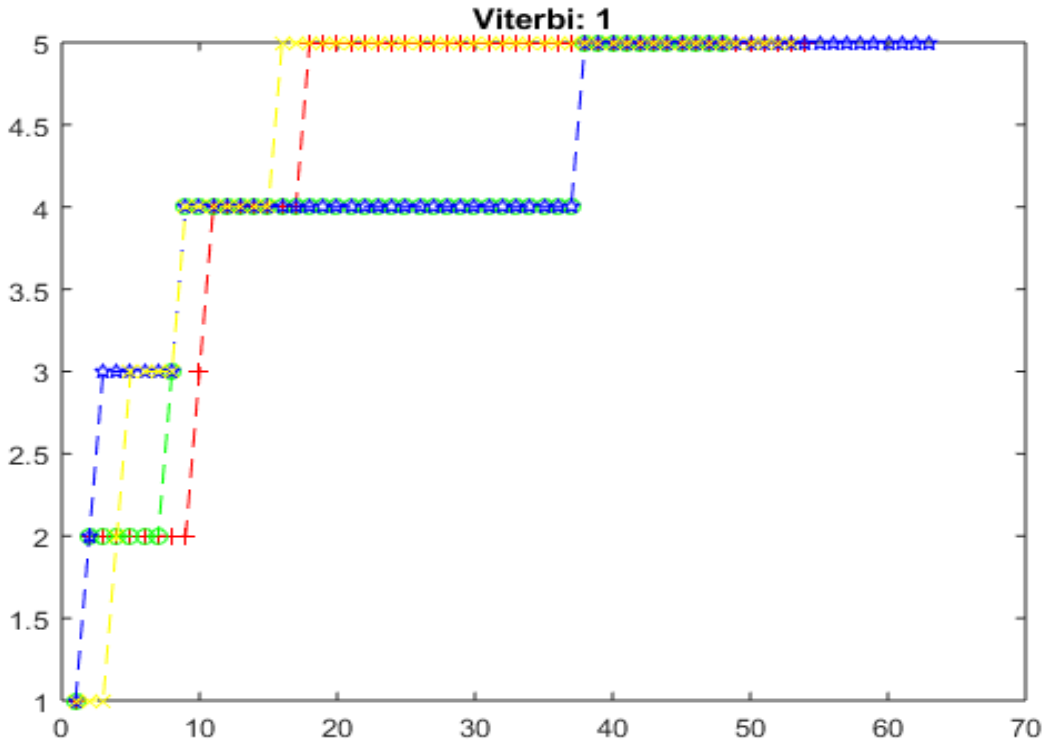
Ενδεικτικά ο Confusion Matrix:

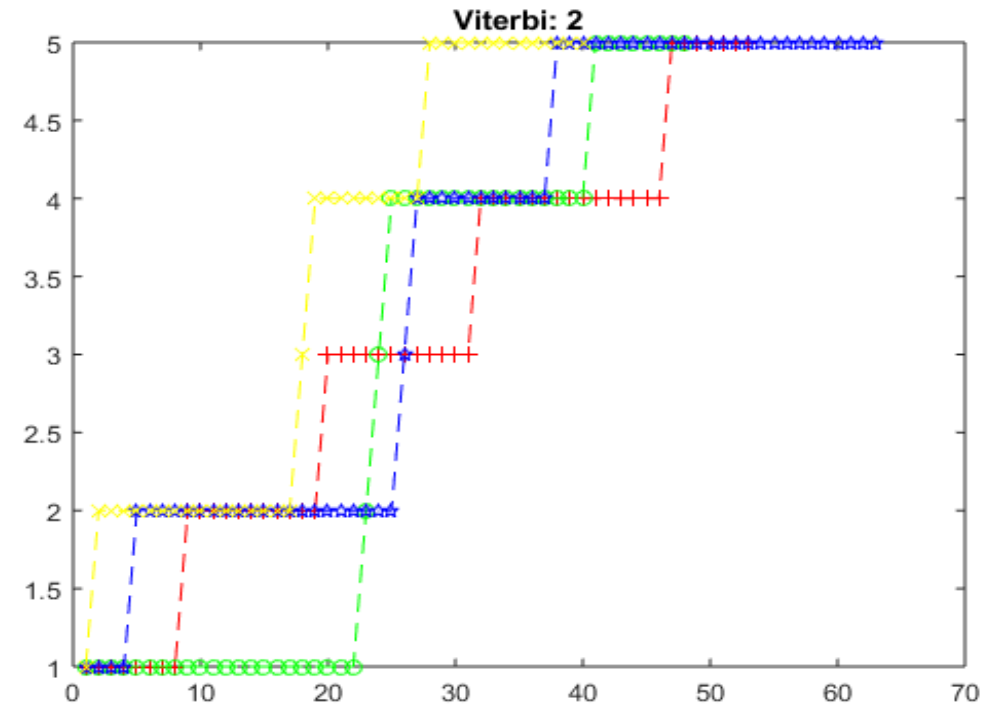


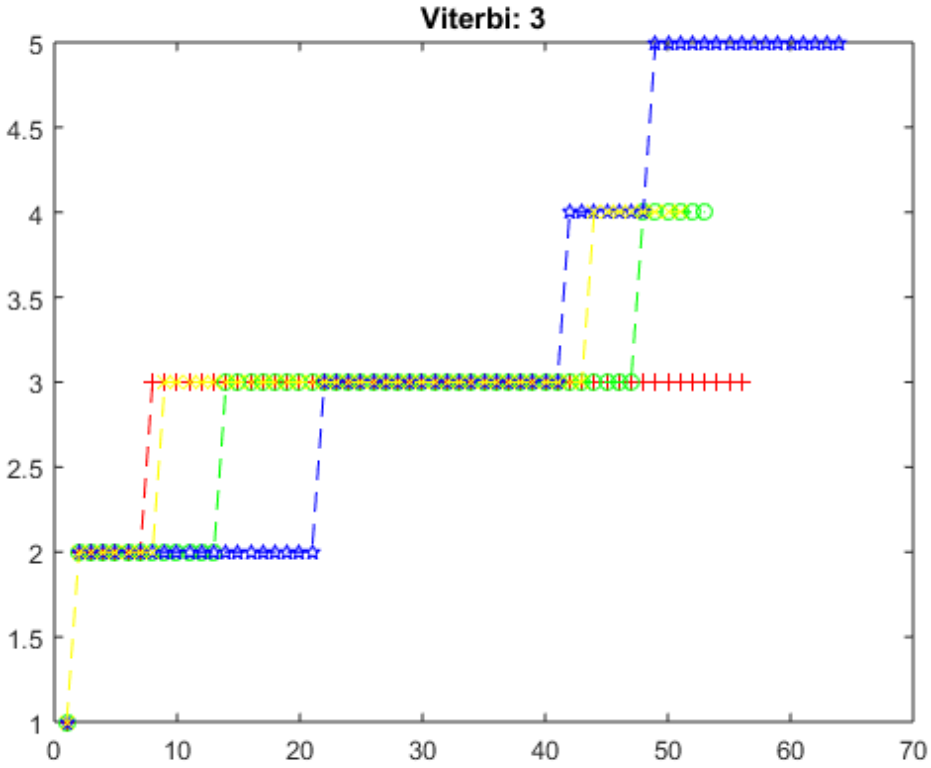
Εδώ θα πρέπει να τονίσουμε ότι αυτά είναι τα αποτελέσματα μιας εκτέλεσης.Τα αποτελέσματα διαφοροποιούνται κάθε φορά που εκτελούμε τον κώδικα λόγω της τυχαιότητας κάποιων στοιχείων του μοντέλου μας όπως προαναφέρθηκε που πάιζουν σημαντικό ρόλο στην αναγνώριση όπως ο πίνακας μεταβάσεων.Έτσι το ποσοστό ενδέχεται να πέσει μέχρι και 5-6 % η να ανέβει ακόμη παραπάνω.

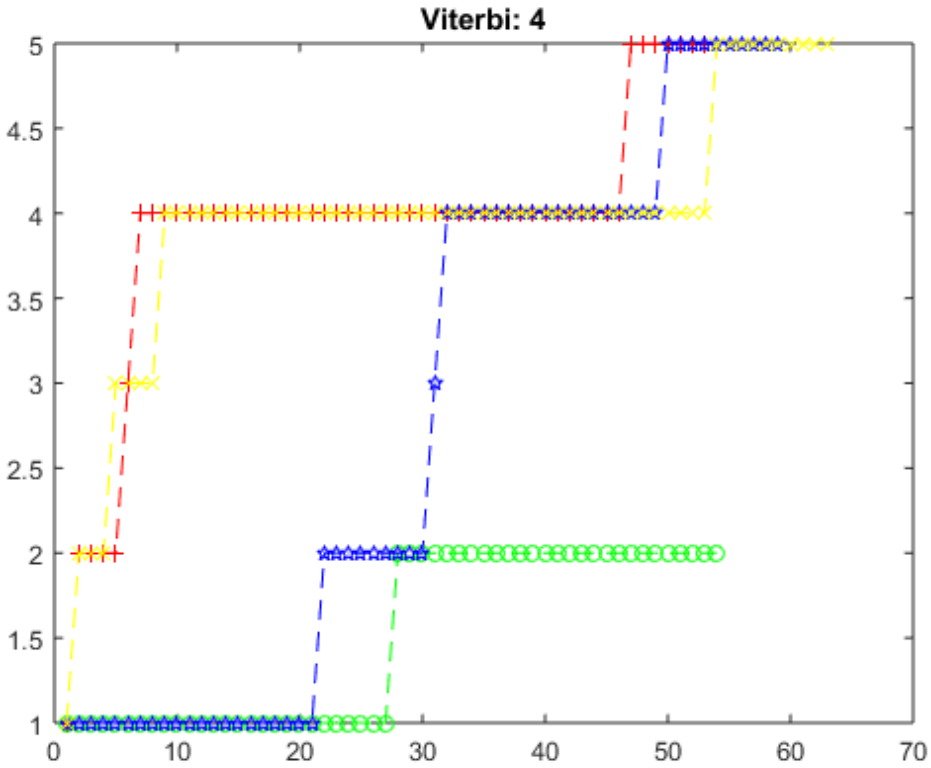
Βήμα 15

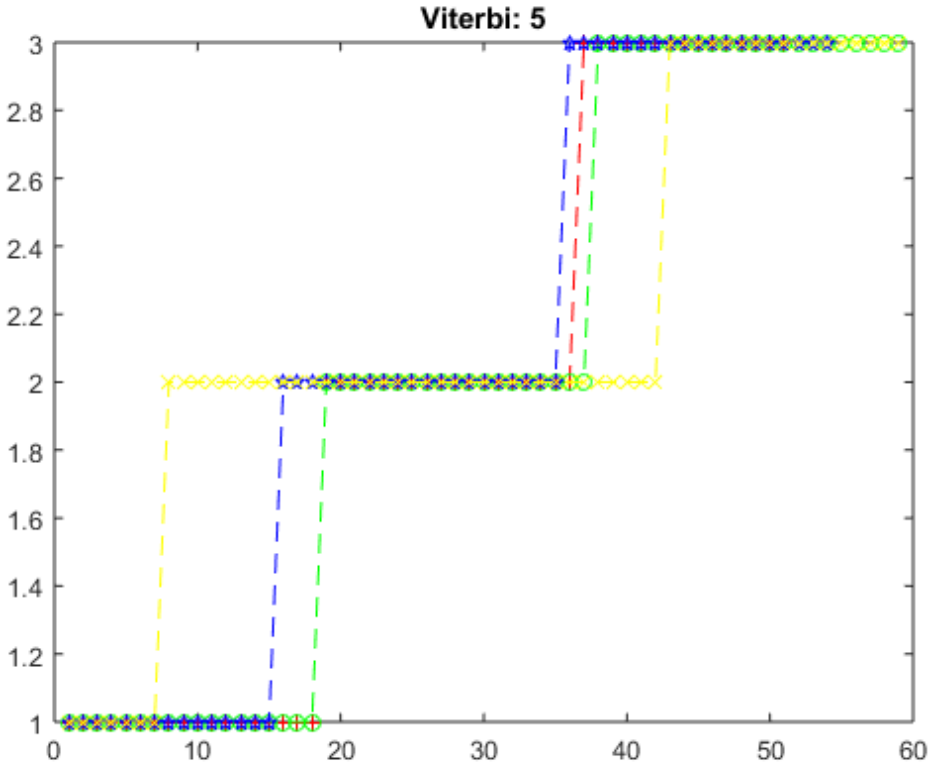
Τέλος για κάθε ψηφίο αρχικά υπολογίζουμε τις πιθανότερες παρατηρήσεις με τη βοήθεια της *mixgauss_prob()* και με χρήση της *viterbi_path()* υπολογίζουμε το πιθανότερο μονοπάτι καταστάσεων που ακολουθεί κάθε ψηφίο.Στη συνέχεια για κάθε ψηφίο παρουσιάζεται η πιθανότερη ακολουθία καταστάσεων με διαφορετικό χρώμα και σύμβολο:

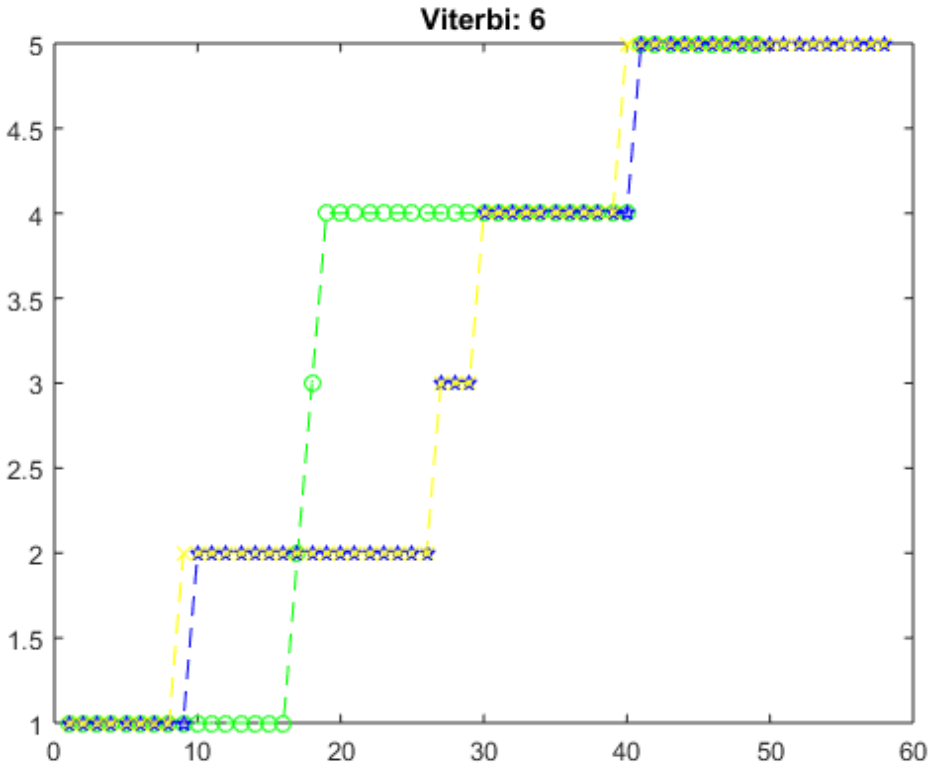


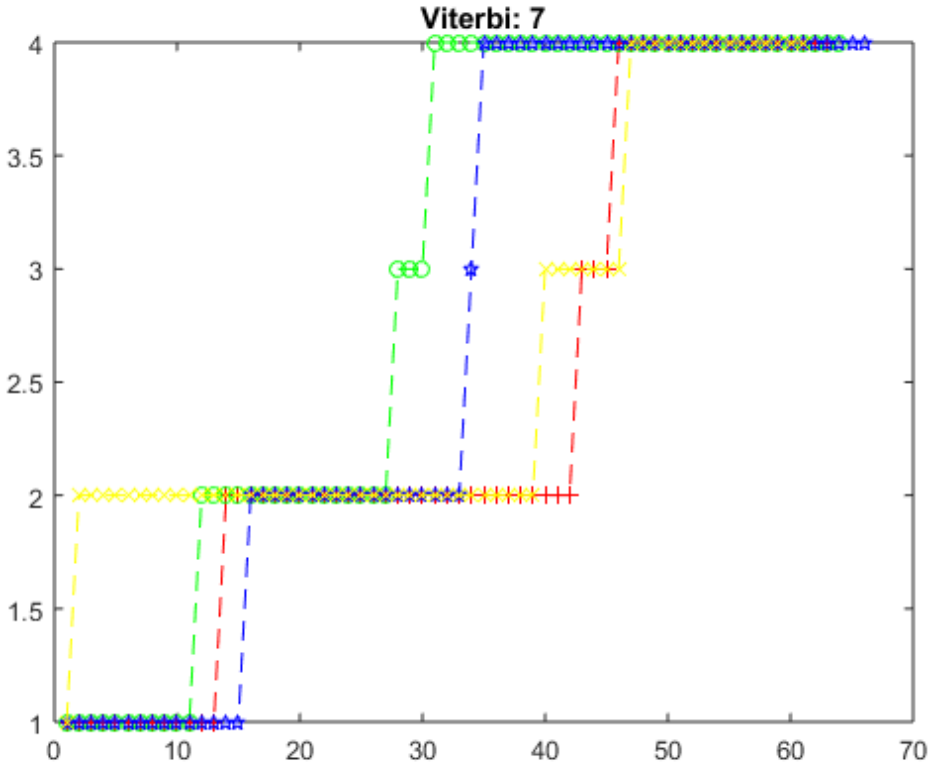


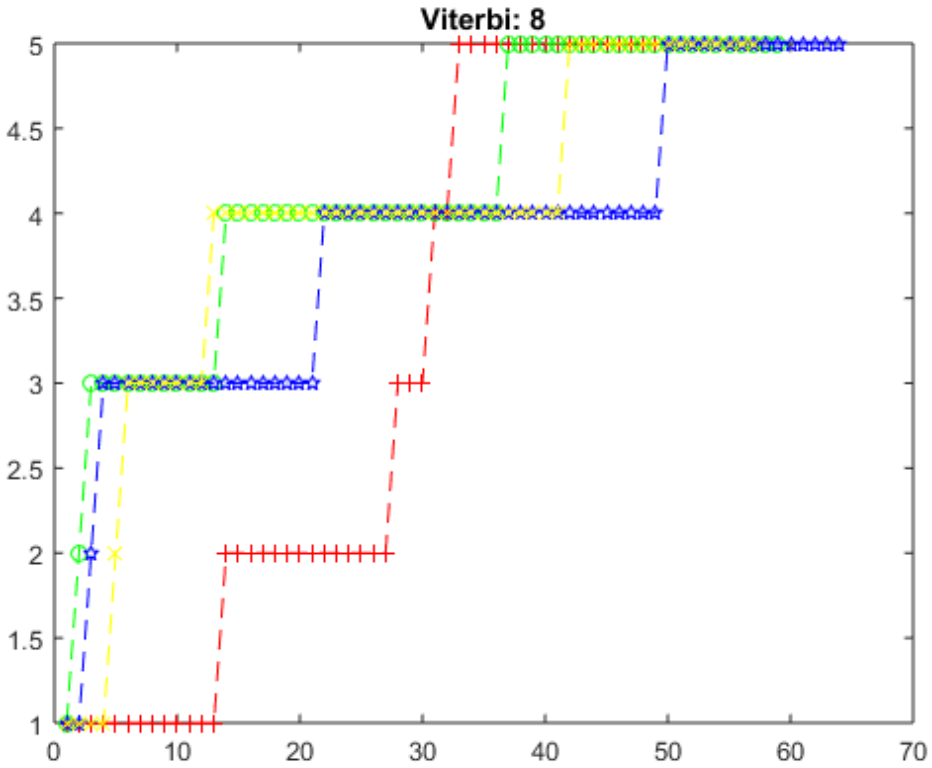

Viterbi: 3
5
4.5
4
3.5
3
2.5
2
1.5
1
0
10
20
30
40
50
60
70

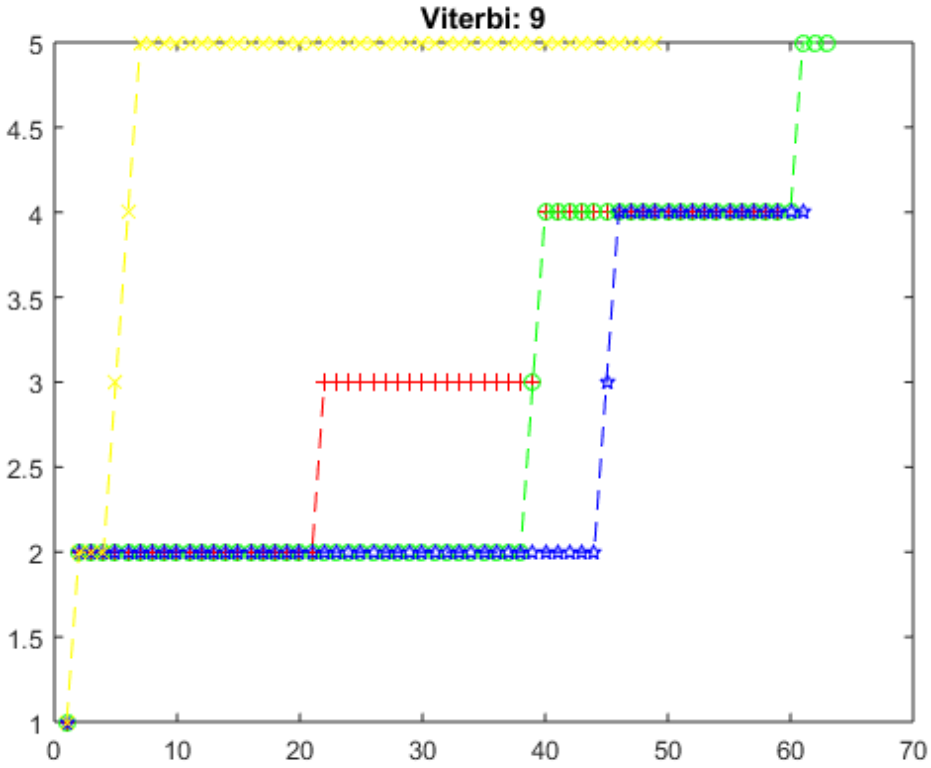

Viterbi: 4
5
4.5
4
3.5
3
2.5
2
1.5
1
0
10
20
30
40
50
60
70


Viterbi: 5
3
2.8
2.6
2.4
2.2
2
1.8
1.6
1.4
1.2
1
0
10
20
30
40
50
60


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


Viterbi: 7
4
3.5
3
2.5
2
1.5
1
0
10
20
30
40
50
60
70


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



Σημείωση:

Για το 6 παρατηρούμε ότι λείπει μια ακολουθία μονοπατιών λόγω του παραλειμένου ομιλητή από τα ίδια τα δεδομένα.