Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Αναγνώριση Προτύπων

2016-2017

Γιώργος Δασούλας
03112010
Νικολάος Ζαρίφης
03112178

2η εργαστηριακή άσκηση

Προθεσμία : 5/12/2016

<u>Σκοπός</u>

Σκοπός της παρούσας εργαστηριακής άσκησης είναι η εξαγωγή χαρακτηριστικών από ακουστικά αρχεία και η κατηγοριοποίηση των δειγμάτων στα αντίστοιχα ψηφία. Τα βήματα της εξαγωγής των χαρακτηριστικών αναλύθηκαν στην προεπεξεργασία της εργαστηριακής άσκησης .
Ως χαρακτηριστικά χρησιμοποιούμε τους 13 από τους συντελεστές που προέκυψαν από τα φίλτρα **Mel ,** μέσα από τα οποία πέρασαν τα πλαίσια του κάθε δείγματος .

Για την κατηγοριοποίηση των δειγμάτων χρησιμοποιούμε στην εργαστηριακή άσκηση κρυφά μαρκοβιανά μοντέλα (HMM) . Συγκεκριμένα , εκπαιδεύουμε 9 μοντέλα (ένα για κάθε ψηφίο) με βάση το 70% των δειγμάτων (Train Set) και για κάθε ακουστικό δείγμα από το TestSet (30% των δοθέντων) υπολογίζουμε την λογαριθμική πιθανοφάνεια για κάθε μοντέλο και επιλέγουμε τη μεγαλύτερη .

<u>Βήμα 10 : Αρχικοποίηση κρυφών μαρκοβιανών μοντέλων</u>

Σε αυτό το μοντέλο αρχικοποιήσαμε τα 9 μας κρυφά μοντέλα Markov μέσω εργαλείων των βιβλιοθηκών **HMM, KPMtools, KPMstats, netlab, PRTools** . Χρησιμοποιήσαμε 6 καταστάσεις για κάθε μοντέλο και 2 γκαουσιανές για κάθε χαρακτηριστικό . Για την αρχικοποίηση των κατανομών χρησιμοποιήσαμε τη συνάρτηση mixgauss_init , με την οποία παρήγαμε τους πίνακες μέσων τιμών και συνδιασποράς των mixture γκαουσιανών κατανομών . Επίσης , χρησιμοποιήσαμε για αρχικό πίνακα μεταβάσεων θέσαμε τυχαίες τιμές στις θέσεις : $a_{i,i}$, $a_{i,i+1}$, οι οποίες όμως να συμβαδίζουν με την στοχαστικότητα του πίνακα (όλες οι γραμμές να αθροίζουν στο 1) . Τέλος , θέσαμε ώς αρχική κατανομή ένα διάνυσμα , μήκους ίσου με τις καταστάσεις , με μηδενικά σε όλες τις θέσεις εκτός της αρχικής , που είναι ίση με 1 .

Λόγω του ότι έχουμε συνεχείς μεταβολές των παρατηρήσεων , χρειαζόμαστε gaussians distributions για να περάσουμε από τις μη-παρατηρήσιμες στις παρατηρήσιμες καταστάσεις .

Βήμα 11 : Χρήση αλγορίθμου Expectation-Maximization

Στο βήμα αυτό εκπαιδεύουμε 9 κρυφά μαρκοβιανά μοντέλα , ένα για κάθε ψηφίο μέσω του αλγορίθμου Expectation - Maximization . Για το σκοπό αυτό χρησιμοποιούμε τη συνάρτηση mhmm_em για την εκπαίδευση των μοντέλων . Για κάθε μοντέλο - ψηφίο , χρησιμοποίησαμε κάθε δείγμα-εκφώνηση του αντίστοιχου ψηφίου ως είσοδο στον αλγόριθμο του Expectation-Maximization . Επίσης , τονίζουμε πως οι παράμετροι που θέσαμε στον κώδικα είναι :

- Iterations = 10
- Mixtures of gaussian distributions = 2
- Number of States = 6

Γενικά , για την εκτέλεση του κώδικα δοκιμάσαμε διαφορετικές τιμές στις παραμέτρους . Όπως θα δούμε και στο Confusion Matrix , οι καταστάσεις , οι οποίες αντιπροσωπεύουν τα φωνήματα (phonemes) , καθορίζουν και την απόδοση του ταξινομητή , σε σχέση με τα φωνήματα που αναγνωρίζει .

Παρατηρήσαμε πως για κάθε εκτέλεση του κώδικα (εκπαίδευση του μοντέλου και κατηγοριοποίηση των test data) είχαμε διαφορετικά αποτελέσματα ακρίβειας (που κυμαίνονταν μεταξύ 75% - 83% .
Για το λόγο αυτό το επόμενο βήμα στον κώδικα είναι να εκτελέσουμε - εκπαιδεύσουμε κάθε μοντέλο 10 φορές . Στη συνέχεια , για τους 10 κατηγοριοποιητές που προκέκυψαν , κρατάμε αυτόν με το μεγαλύτερο ποσοστό επιτυχίας **συνολικής αναγνώρισης .**

Η διαφορά ανάμεσα στη 1 και τις πολλές εκτελέσεις φαίνεται στις παρακάτω τιμές :
Για μία εκτέλεση είχαμε διακύμανση απόδοσης ανάμεσα στις τιμές : 75%-83% , ενώ για 10 εκτελέσεις και κρατώντας τον καλύτερο ταξινομητή είχαμε απόδοση :

```
Best accuracy rate from 10 classifiers: 86.666667%  for classifier 8
```

Διερεύνηση :

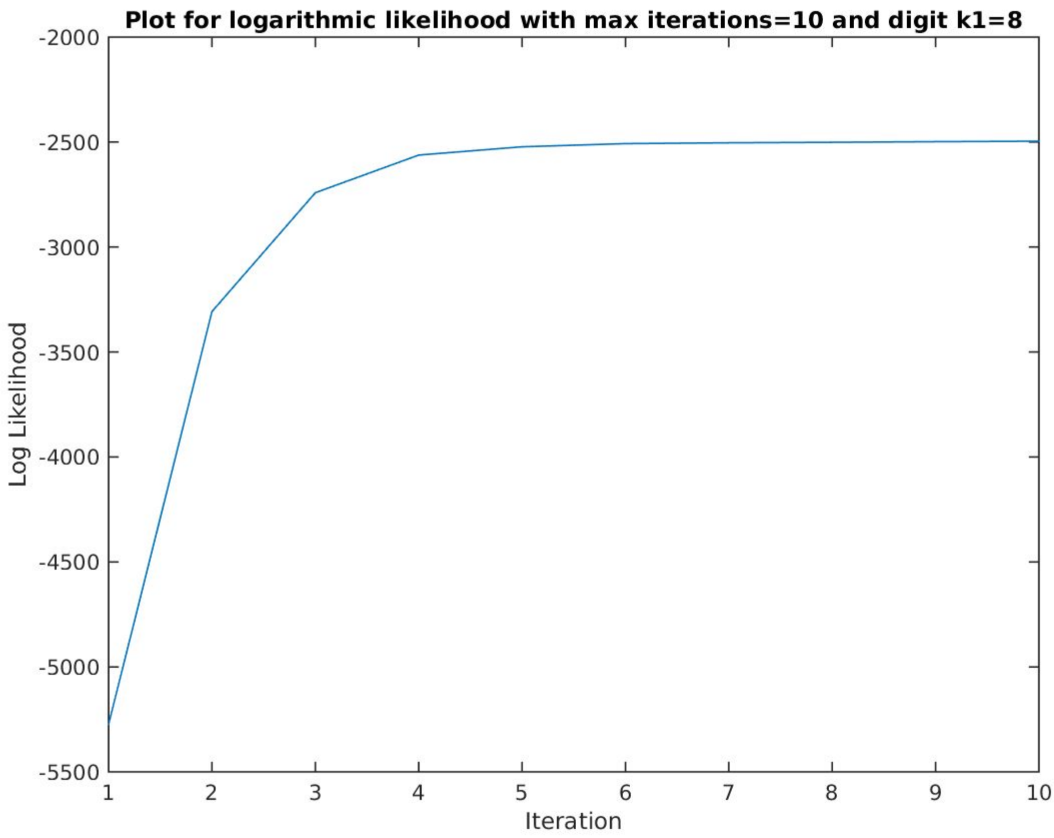
Όπως είδαμε παραπάνω λόγω της διακύμανσης απόδοσης του ταξινομητή , χρειάστηκε να εκπαιδεύσουμε παραπάνω από μία φορές τα μοντέλα μας .
Για την αύξηση της ακρίβειας του κατηγοριοποιητή μας , εκπαιδεύσαμε τα μοντέλα μας 10 φορές και με **πλειοψηφικό** κανόνα για κάθε δείγμα , πήραμε την τελική απόφαση . Αυτή η τεχνική μας έδωσε αρκετά μεγαλύτερο accuracy , όπως φαίνεται παρακάτω :

```
Accuracy rate: 93.333333%
```

Προφανώς , αυτή η τεχνική διαφέρει πλέον από το κλασσικό κρυφό μαρκοβιανό μοντέλο , αλλά αποτελεί μια καλή παραλλαγή με πολύ καλύτερα αποτελέσματα . Αυτή η απόδοση φαίνεται και στο παρακάτω ερώτημα με το Confusion Matrix , όπου έχουμε πολύ λιγότερες συγχύσεις του ταξινομητή .

<u>Βήμα 13 : Log Likelihood Plot</u>

Στο συγκεκριμένο βήμα πλοτάρουμε την τιμή της λογαριθμικής πιθανοφάνειας για κάθε βήμα επανάληψης . Για το παρακάτω διάγραμμα χρησιμοποιήσαμε το μοντέλο που αντιστοιχεί στο ψηφίο $k_1 = 8$ και τις τιμές πιθανοφάνειας από τα διανύσματα που προέκυψαν κατά την εκπαίδευση του μοντέλου . Για την εκπαίδευση του μοντέλου χρησιμοποιήσαμε max iterations = 10 . Όπως είδαμε με διαφορετικά τρεξίματα , αρκεί η εκπαίδευση μέχρι 7-8 μέγιστες επαναλήψεις . Από το παρακάτω διάγραμμα φαίνεται , πως αρχίζουμε να βλέπουμε σύγκλιση από το 5ο-6ο βήμα επανάληψης και μετά . Αυτό μας δείχνει πως αυτός ο αριθμός των επαναλήψεων μπορεί να δώσει δυνατότητα **γενίκευσης** στο κρυφό μαρκοβιανό μοντέλο για το συγκεκριμένο TrainSet . Επίσης , όπως ήταν αναμενόμενο η λογαριθμική πιθανοφάνεια αυξάνεται με το πλήθος των επαναλήψεων , καθώς ο αλγόριθμος μεγιστοποίησης της πιθανοφάνειας είναι αυξητικός.



<u>Βήμα 14 : Confusion Matrix</u>

Με μία εκτέλεση του κώδικα :

	1	2	3	4	5	6	7	8	9
1	5	0	0	0	0	0	0	0	0
2	0	2	3	0	0	0	0	0	0
3	0	0	5	0	0	0	0	0	0
4	0	0	0	5	0	0	0	0	0
5	0	0	0	0	5	0	0	0	0
6	0	0	1	0	0	4	0	0	0
7	0	0	1	0	0	0	4	0	0
8	0	0	2	0	0	0	0	3	0
9	2	0	1	0	0	0	0	0	2

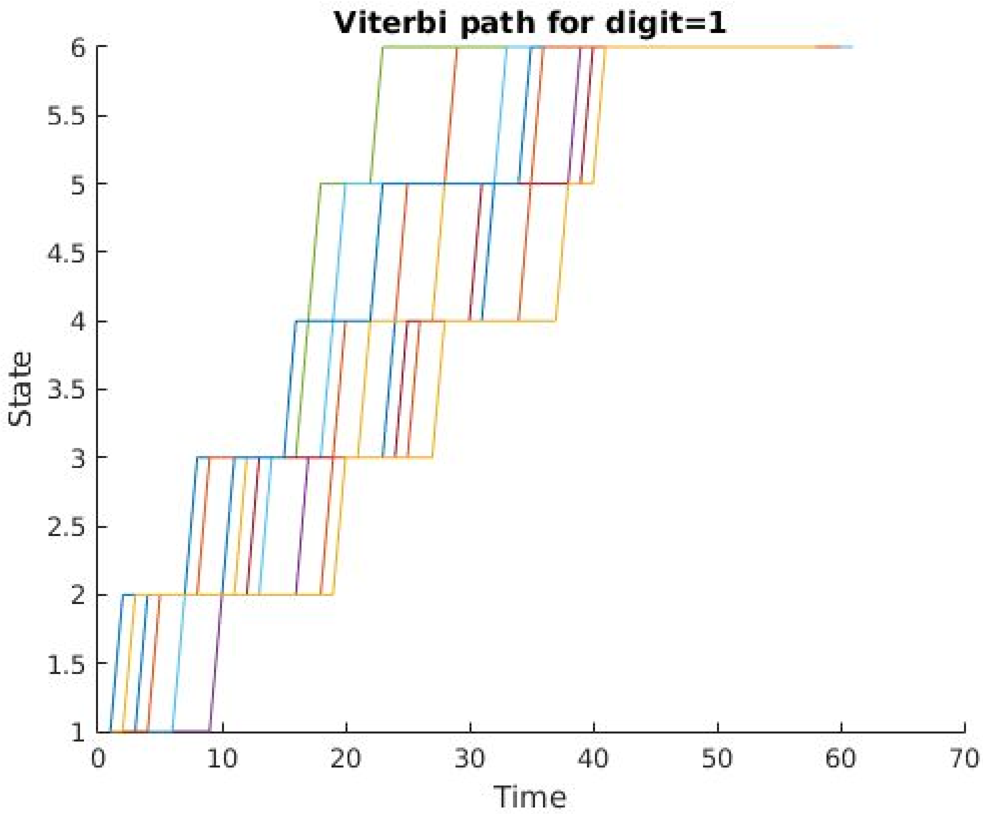
Με πολλές εκτελέσεις και πλειοψηφική επιλογή

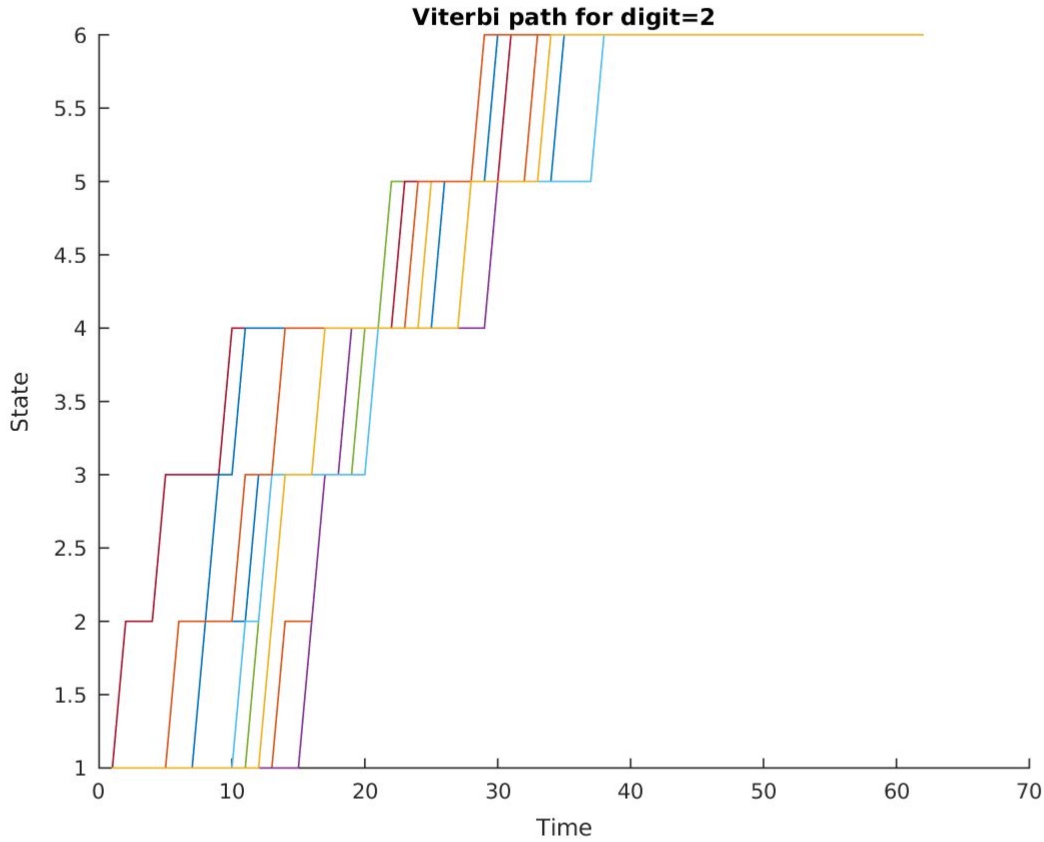
	1	2	3	4	5	6	7	8	9
1	5	0	0	0	0	0	0	0	0
2	0	5	0	0	0	0	0	0	0
3	0	0	5	0	0	0	0	0	0
4	0	0	0	5	0	0	0	0	0
5	0	0	0	0	5	0	0	0	0
6	0	0	0	0	0	4	1	0	0
7	0	0	0	0	0	1	4	0	0
8	0	0	0	0	0	0	0	5	0
9	0	0	0	0	1	0	0	0	4

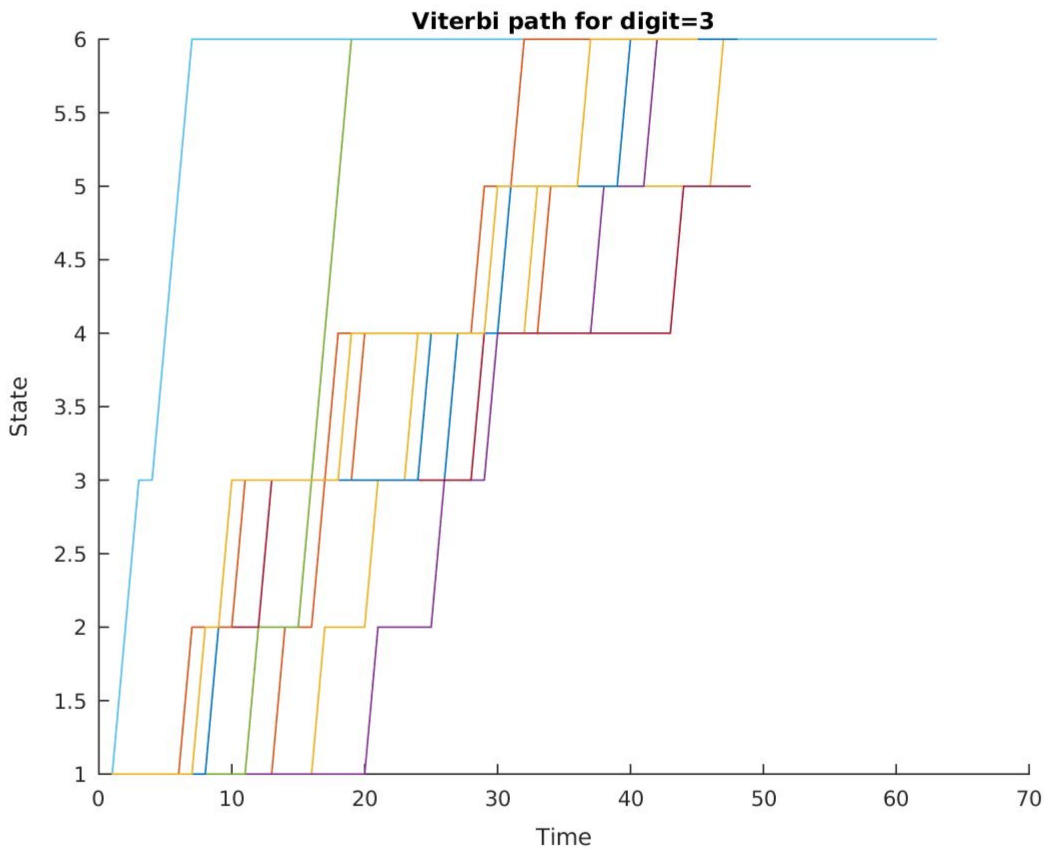
Οπως βλεπουμε απο το confusion matrix βλεπουμε σε πια ψηφια εχει σύγχυση και βλεπουμε που δεν λειτουργει καλα ο ταξινομητης.Το ποσοστο ολικης αναγνωρισης προκύπτει απο το αθροισμα της διαγωνιου δια το πλήθος των δειγμάτων, το οποίο εχουμε βρει ούτως ή άλλως στα παραπάνω ερωτήματα.

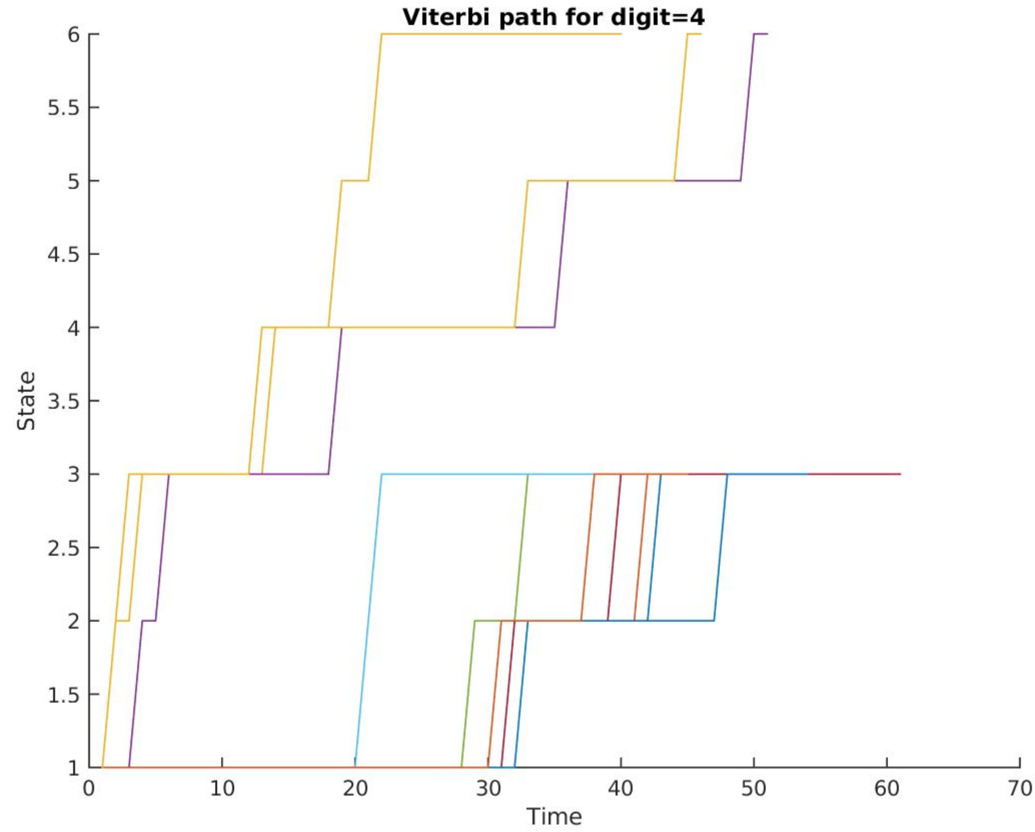
Βήμα 15 : Μονοπάτια Viterbi

Παρακάτω παραθέτουμε όλα τα διαγράμματα για τα 9 ψηφία viterbi που προέκυψαν . Από τα παρακάτω βλέπουμε τις ακολουθίες κρυφών καταστάσεων - φωνημάτων που δίνουν τα αντίστοιχα δείγματα .

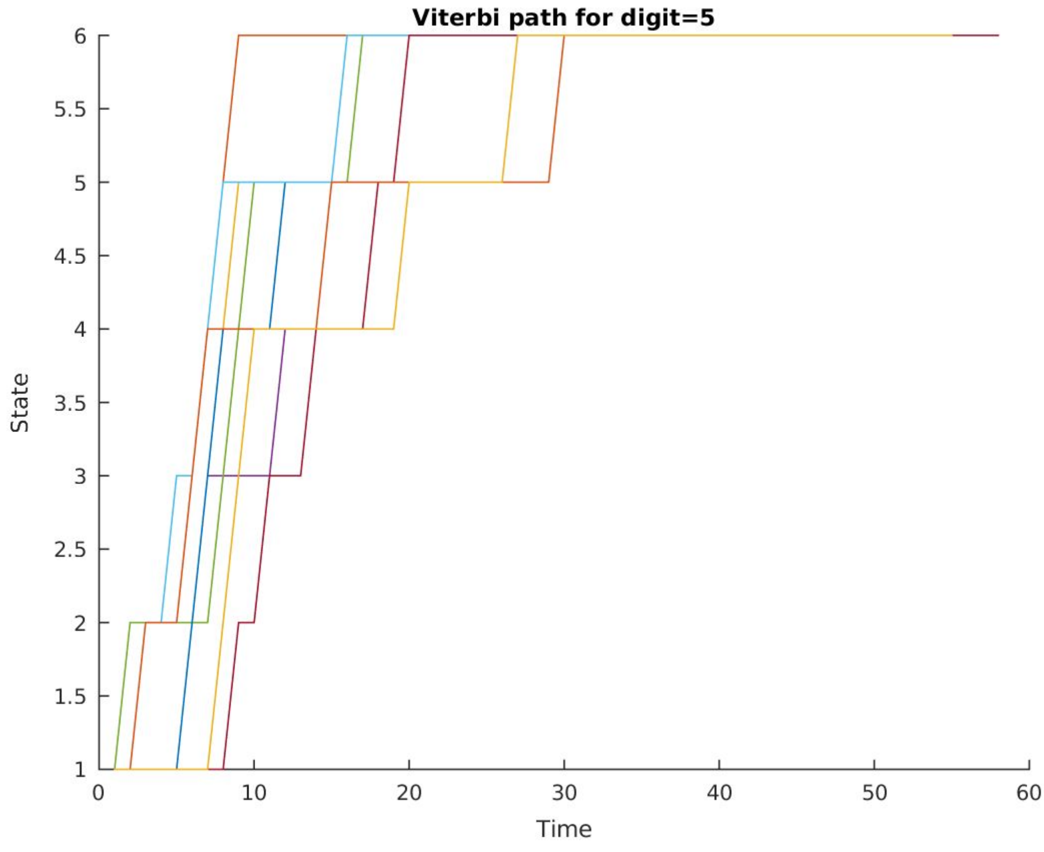



Viterbi path for digit=2
State
Time
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
0
10
20
30
40
50
60
70

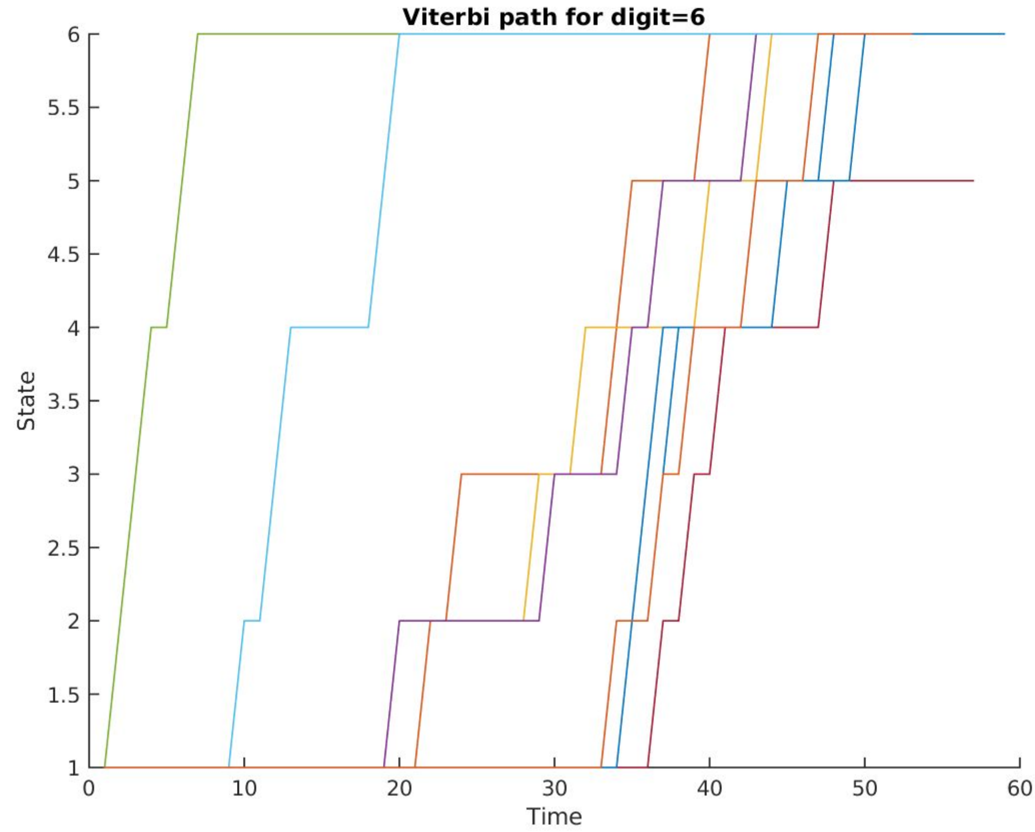

Viterbi path for digit=3
State
Time
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
0
10
20
30
40
50
60
70

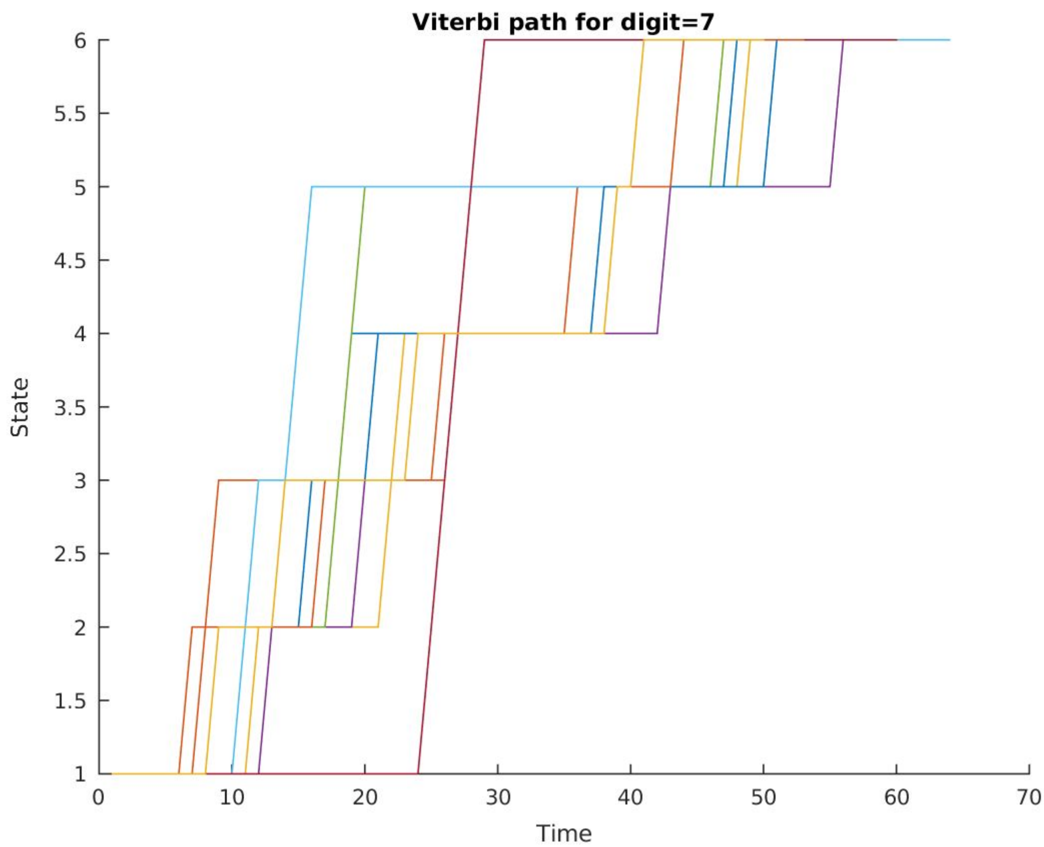


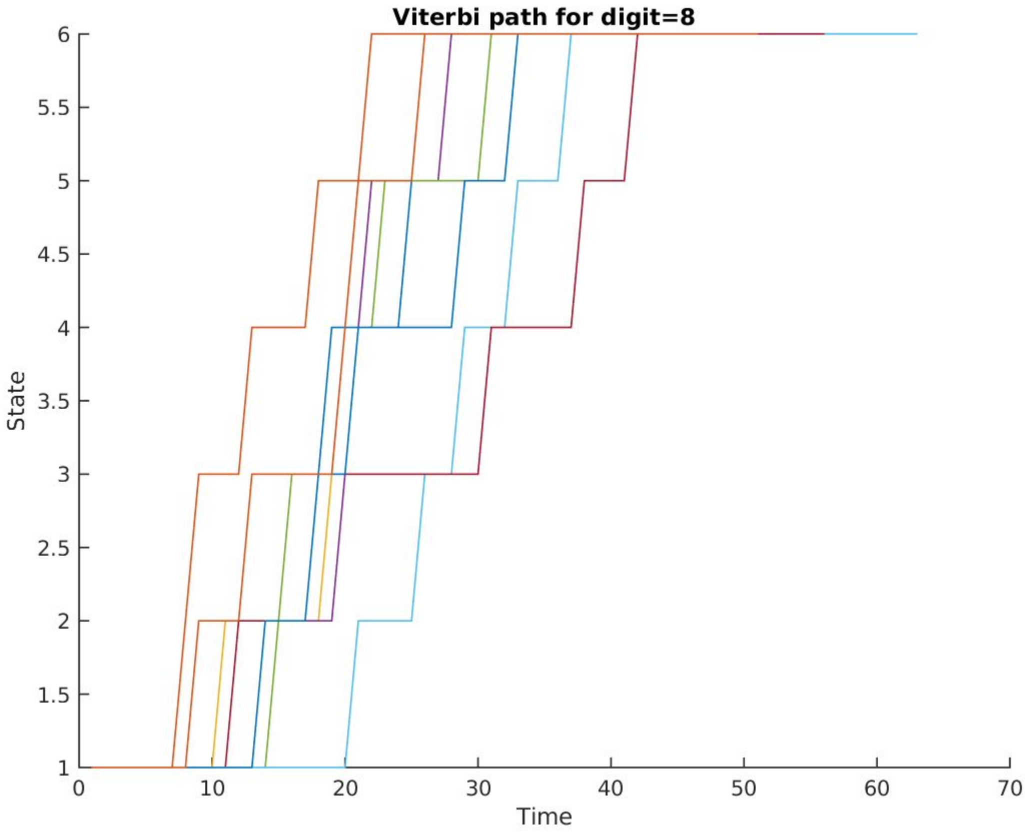
Viterbi path for digit=4
State
Time
0
10
20
30
40
50
60
70
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6



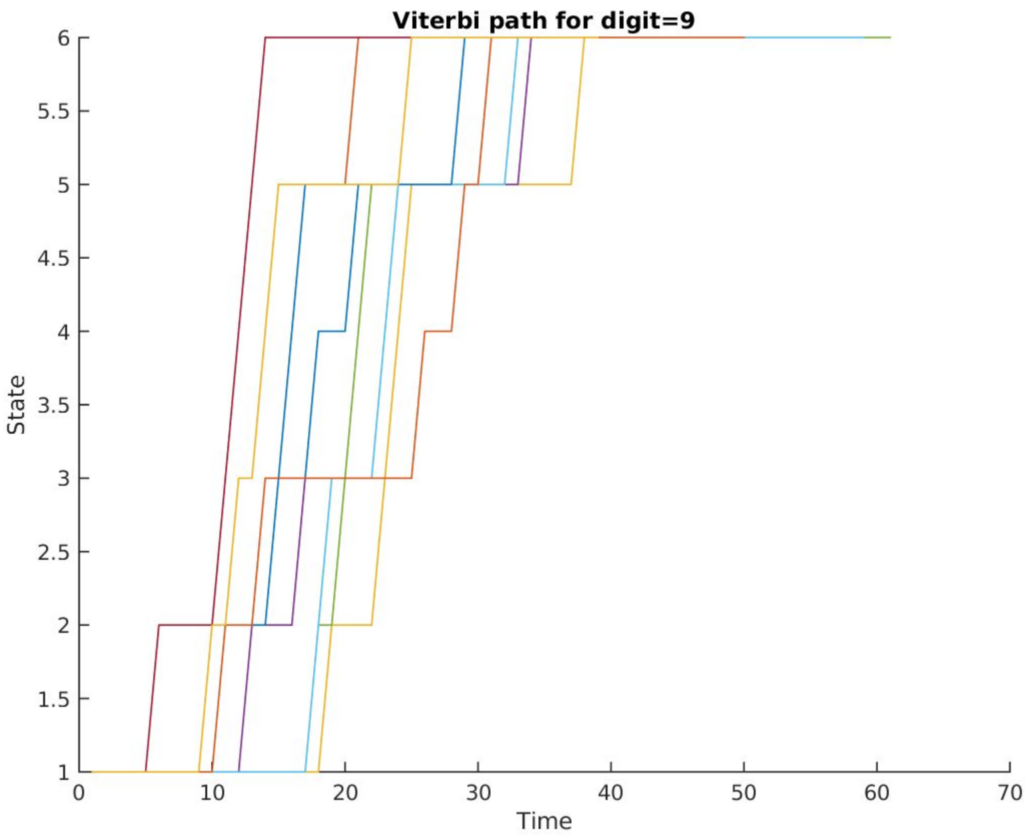
Viterbi path for digit=5
State
Time
0
10
20
30
40
50
60
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6


Viterbi path for digit=6
State
Time
6
5.5
5
4.5
4
3.5
3
2.5
2
1.5
1
0
10
20
30
40
50
60


Viterbi path for digit=7
State
Time
6
5.5
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
State
Time



Viterbi path for digit=9
State
Time

Όπως είναι λογικό όλα τα φωνήματα ξεκινάνε από κατάσταση σιγής κι καταλήγουν στην κατάσταση σιγής 6. Ανάλογα με το ψηφίο έχουμε και αντίστοιχη ταχύτητα στην κατάληξη της ταχύτητας σιγής . Βλέπουμε οτι για κάθε ψηφιο η ακολουθια φτάνει σε κάθε στάδιο σε διαφορετικό χρόνο -πλαίσιο, αυτό γιατί κάθε ψηφίο έχει διαφορετική ακουστική δομή. Επίσης ανάλογα έχουμε σε μερικά ψηφια διαφορές σε κάθε δειγμα, γιατί εξαρτάται απο το πως το προφέρεται το κάθε ψηφίο.

Τέλος , επισημαίνουμε πως αν από τους 13 συντελεστές MFCC , αφαιρούσαμε τους 2 πρώτους , θα είχαμε καλύτερα αποτελέσματα , λόγω της διαφοράς φωνημάτων .

Ο πρώτος ειδικά συντελεστής MFCC είναι αρκετά μεγαλύτερος από τους υπόλοιπους και λόγω της διαφοράς που υπάρχει , υπάρχουν και μεγαλύτερες αποκλίσεις , δίνοντας το περιθώριο για περισσότερα λάθη.