Αναγνώριση Προτύπων

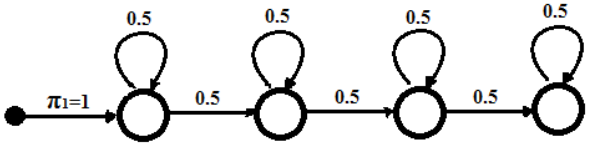
Θέμα: Κρυφά Μαρκοβιανά Μοντέλα και Αναγνώριση Φωνής

Εισαγωγή

Σ'αυτήν την εργασία βασικό αντικείμενο ήταν η εκπαίδευση κρυφών μαρκοβιανών μοντέλων προς δημιουργία ενός συστήματος αναγνώρισης μεμονωμένων ψηφίων. Σαν δεδομένα χρησιμοποιήθηκαν τα χαρακτηριστικά που εξήχθησαν απο την 1η εργαστηριακή άσκηση και συγκεκριμένα για κάθε εκφωνητή οι 13 πρώτοι mel frequency cepstrum coefficients που εξήχθησαν απο επικαλυπτόμενα κατα 10msec χρονικά παράθυρα των 25msec το καθένα. Τα δεδομένα αυτά αποθηκεύτηκαν σε ενα 1x9 cell στοιχείο DATA.mat το οποίο σε κάθε cell περιέχει επιπλέον 14 κελιά, καθένα απο τα οποία έχει υπο την μορφή πινάκων Tx13 τους mfccs του αντίστοιχου ομιλητή , οπού T το πλήθος των χρονικών παραθύρων κατα την ανάλυση του σήματος. Αναφέρουμε εδώ οτι παρόλο που για κάποια ψηφία είχαμε 15 ομιλητές, σε κάποια άλλα είχαμε 14. Έτσι για την μοντελοποίηση και έλεγχο του συστήματος χρησιμοποιήθηκαν τυχαία μόνο οι 14 για να αποφευχθεί πιθανό bias. Τα δεδομένα αυτά διαμοιράζονται τυχαία σε training set και testing set με αναλογία 70-30 ενώ δίνεται η δυνατότητα να επιλεγεί το πλήθος των επαναλήψεων του πειράματος (training - testing) λαμβάνοντας τυχαία διαφορετικά δεδομένα εκπαίδευσης και ελέγχου. Επίσης δίνεται η δυνατότητα επιλογής του πλήθους των καταστάσεων κάθε μοντέλου που εκφράζει ένα ψηφίο (NOS) καθώς και το πλήθος των gaussians που περιγράφει κάθε κατάσταση (NOG) όπως επίσης και το μέγιστο πλήθος επαναλήψεων του αλγορίθμου Expectation Maximization κατα την εκπαίδευση των μοντέλων (Niter). Τέλος ηχογραφήθηκαν τα ψηφία απο 1-9 απο εμένα τον ίδιο και αφού εξήχθησαν τα χαρακτηριστικά και αποθηκέυθηκαν στο 1x9 cell στοιχείο MGV.mat έγινε testing και εξήχθησαν τα αντίστοιχα αποτελέσματα.

Αρχικοποίηση-Εκπαίδευση μοντέλων και Δεδομένα εκπαίδευσης-αξιολόγησης

Τα μοντέλα που χρησιμοποιήθηκαν ήταν αρχικά της μορφής left-right ωστόσο έγιναν και δοκιμές με τυχαία τοπολογία μοντέλων για σύγκριση των αποτελεσμάτων. Συγκεκριμένα σε πρώτη φάση ο πίνακας ματάβασης καταστάσεων επιλέχθηκε να έχει στην κύρια και τη δευτερεύουσα διαγώνιο στοιχεία με τιμή 0.5 και μηδενικά αλλού. Επίσης οι αρχικές πιθανότητες ήταν όλες μηδενικές εκτός απο την πιθανότητα να βρεθούμε στην πρώτη κατάσταση η οποία τέθηκε ίση με τη μονάδα. Στο σχήμα 1 φαίνεται η τοπολογία των μοντέλων για 4 καταστάσεις.



Σχήμα 1 : Τοπολογία επιλεγμένων μοντέλων

Όπως αναφέρθηκε και στην εισαγωγή τα κύρια δεδομένα βρίσκονται εντός του cell στοιχείου DATA.mat. Για να διαχωριστούν με τυχαίο τρόπο σε δεδομένα εκπαίδευσης και ελέγχου χρησιμοποιήσαμε την εξής λογική: Διάλεξε ένα ψηφίο (digit) προς διαχωρισμό των δεδομένων σε testing και training set. Στη συνέχεια (ωστε το πρόγραμμα να είναι robust) βρές πόσοι είναι οι ομιλητές αυτού του ψηφίου (στην προκειμένη περίπτωση 14) και με χρήση της συνάρτησης ***randperm*** δημιούργησε ένα διάνυσμα με τυχαία σειρά απο 1 εως το πλήθος των ομιλητών (άρα ανακατεμένοι οι αριθμοί απο 1 εως 14). Επίσης επέλεξε ένα ποσσοστό για training (π.χ 0.7) και υπολόγισε το πλήθος των ομιλητών που πρέπει να αντιστοιχεί στο training set. Η παραπάνω προεπεξεργασία φαίνεται στο παρακάτω τμήμα προγράμματος.

```
NUMBER = DATA{digit};                                %κραταω τις εκφωνησεις του ψηφιου digit
no_of_speakers = size(NUMBER,2);                     %το συνολο των εκφωνητων
permutation = randperm(no_of_speakers);              %μπερδεμενοι αριθμοι απο το 1 εως τον αριθμο των εκφωνησεων
percent_for_train=0.7;                               %θα κρατησουμε το 70% για training
keep_for_train=ceil(percent_for_train * no_of_speakers); %πλήθος ομιλητών για training
```

Πλέον εφόσον γνωρίζουμε το πλήθος των ομιλητών που θέλουμε να κρατήσουμε για training, και προφανώς τους υπόλοιπους για testing , και έστω αυτό το πλήθος είναι 10 (που για 14 συνολικά ομιλητές είναι το 70%) τότε θα πρέπει να κρατήσουμε απο το cell στοιχείο NUMBER 10 στοιχεία τα οποία θα περιγράφονται απο τις 10 πρώτες τιμές του διανύσματος permutation. Τα επόμενα στοιχεία του permutation θα είναι οι δείκτες των στοιχείων του NUMBER cell που θα χρησιμοποιήσουμε για testing ομιλητές. Επίσης ταυτόχρονα με αυτή τη διαδικασία μπορούμε να δημιουργήσουμε έναν πίνακα concat_data οποίος θα περιέχει συνενωμένα τα δεδομένα των ομιλητών προς εκπαίδευση και θα χρησιμοποιηθεί κατα την κλήση της mixgauss_init για αρχικοποίηση των χαρακτηριστικών των γαουσιανών (μέσοι , πίνακες συμμεταβλητότητας και βάρη). Η διαδικασία περιγράφεται απο τον παρακάτω κώδικα.

```
concat_data = [];                                      %αρχικοποίηση των δεδομένων για mixgauss_init
for i=1:keep_for_train                                 %για το πλήθος των ομιλητών που θα κρατήσουμε
    concat_data = [concat_data [NUMBER{permutation(i)}]']; %concatenated δεδομενα για τη mixgauss_init
    TRAIN_DATA{i} = [NUMBER{permutation(i)}]';         %πλεον 10 τυχαιοι ομιλητες στα cells του TRAIN_DATA
end                                                    %τέλος

for i = 1:no_of_speakers-keep_for_train
    TEST_DATA{i} = [NUMBER{permutation(keep_for_train+i)}]'; %οι υπολοιποι ομιλητες για τεστ
end
```

Παρατηρούμε οτι δουλεύουμε πολύ με cells το οποίο φάνηκε πολύ βολικό λόγω της φύσης του προβλήματος καθώς τα δεδομένα έχουν κατα κύριο λόγο διαφορετική διάσταση. Στη συνέχεια μπορούμε να αρχικοποιήσουμε τις παραμέτρους των gaussian χρησιμοποιώντας τη συνάρτηση ***mixgauss_init*** με την οποία θα κάνουμε ένα απλό kmeans. Το πλήθος των γκαουσιανών προς αρχικοποίηση είναι ίσο με το γινόμενο των καταστάσεων που θέλουμε να έχει το κάθε μοντέλο (NOS-Number Of States) επι το πλήθος των gaussian που περιγράφουν την κάθε κατάσταση (NOG-Number Of Gaussians). Στη συνέχεια πρέπει να προσέξουμε ωστε οι πίνακες των μέσων, συμμεταβλητότητας και βαρών των γκαουσιανών να έχουν σωστή διάσταση, ομαδοποιώντας ουσιαστικά τις γκαουσιανές ανα κατάσταση, καθώς η mixgauss_init δεν λαμβάνει υπ'οψιν κάτι τέτοιο. Τέλος αρχικοποιούμε πλέον και τις παραμέτρους της τοπολογίας του μοντέλου όπως περιγράφηκε στην αρχή της ενότητας.

```
[mu0, Sigma0, weights0] = mixgauss_init(NOS*NOG, concat_data, 'diag','kmeans'); %για αρχικοποίηση

Sigma0 = reshape(Sigma0, [size(Sigma0,1) size(Sigma0,2) NOS NOG]);              %σωστή διάσταση των πινάκων
mu0 = reshape(mu0, [size(mu0,1) NOS NOG]);
mixmat0 = reshape(weights0, [NOS NOG]);

T = diag(ones(NOS,1)/2,0)+diag(ones(NOS-1,1)/2,1);                              %Τοπολογία μοντέλου
P=zeros(1,NOS); P(1)=1;
```

Έπειτα μπορούμε με βάση αυτές τις παραμέτρους να καλέσουμε τη συνάρτηση ***mhmm_em*** έχοντας επιλέξει μέγιστο αριθμό επαναλήψεων που καθορίζεται απο τη μεταβλητή Niter. Βασικό εδώ είναι να αναφέρουμε οτι παρουσιάστηκαν πολλές φορές προβλήματα με τη συνάρτηση ***assert.m*** όταν αυτή καλείται απο τη συνάρτηση ***fwdback.m.*** Επειδή αυτό διέκοπτε το πρόγραμμα κατα τη διάρκεια εκαπαίδευσης κατι που δεν επέτρεπε να καταφέρουμε να φτιάξουμε ένα αυτοματοποιημένο σύστημα αξιολόγησης όλων των πιθανών συνδιασμών καθώς, έγινε τροποποίηση της fwdback.m κατα την οποία αποφεύγουμε την κλήση της assert.m αλλα παράλληλα αναγνωρίζουμε το αν κάτι τέτοιο θα είχε συμβεί απο την τιμή της Log Likelihood. Συγκεκριμένα αν η Log Likelihood (LL) καταλήξει να έχει την τιμή -Inf αυτό σημαίνει οτι θα έχει παρουσιαστεί κάποιο σφάλμα και οφείλουμε να επαναλάβουμε τη διαδικασία απο το αμέσως προηγούμενο στάδιο.

```
[LL, prior, transmat, mu, Sigma, mixmat] = ...
     mhmm_em(TRAIN_DATA, P, T, mu0, Sigma0, mixmat0,'max_iter',Niter);

 if LL(end)==-Inf    %Για να αποφυγουμε το προβλημα της assert
     MODEL;          %Επανάληψη για το συγκεκριμένο digit
 end
```

Τελικά επαναλαμβάνοντας τη διαδικασία για κάθε ψηφίο αποθηκεύουμε τα χαρακτηριστικά του κάθε μοντέλου σε ένα γενικότερο cell ωστε να μπορέσουμε να τα ανακαλέσουμε κατα τη διάρκεια του testing.

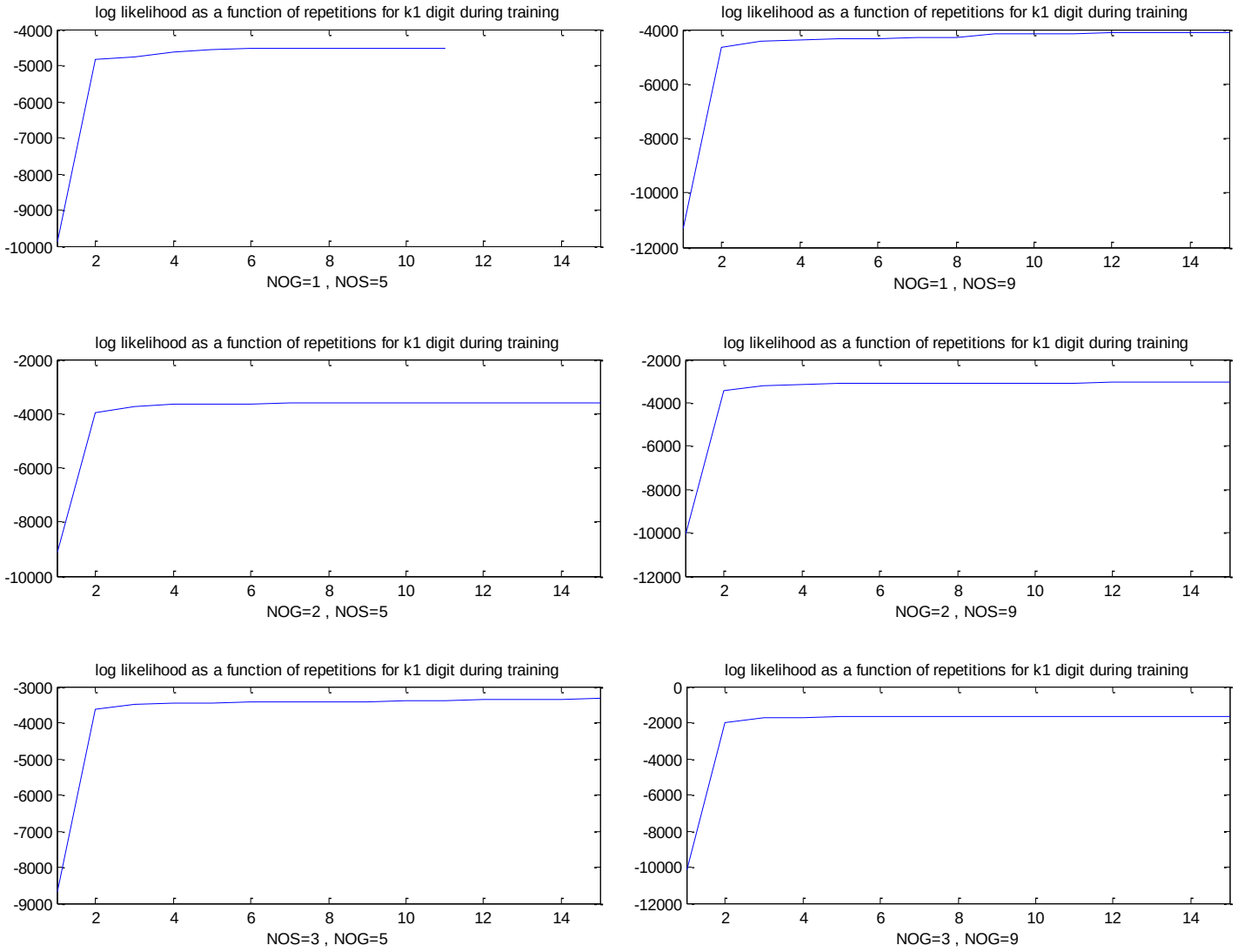
```
LLL{digit} = LL;                          %Πλέον σε κάθε στοιχειο των διπλανών
PRIOR{digit} = prior;                     %1x9 cells θα βρίσκονται τα
TRANSMAT{digit} = transmat;               %χαρακτηριστικά του μοντέλου
MU{digit} = mu;                           %του ψηφίου digit
SIGMA{digit} = Sigma;
MIXMAT{digit} = mixmat;
TEST{digit} = TEST_DATA;
```

Στο τέλος στο 1x9 cell TEST κάθε στοιχείο του θα είναι ένα επιπλέον 1x4 cell οπου το καθένα θα αντιστοιχεί πλέον στα χαρακτηριστικά ενός ομιλητή του digit που θα χρησιμοποιήσουμε για testing.

Παραδοτέα-Ερωτήματα

1. Σ'αυτό το σημείο να αναφέρουμε πως ηχογραφήθηκαν και ελέχθησαν τα δεδομένα όπως περιγράφεται ρητά απο την εκφώνηση της άσκησης. Αυτά βρίσκονται εντός του 1x9 cell στοιχείου MGV.mat οπου το i-οστό στοιχείο (π.χ MGV{i}) περιέχει τα MFCC χαρακτηριστικά του ψηφίου i που εκφωνήθηκε προσωπικά απο εμένα. Βέβαια επειδή κάτι η δημιουργία ενος γενικότερου συνόλου δεδομένων προς έλεγχο δεν είναι υποχρεωτική, όπως είπαμε έγινε διαχωρισμός των ήδη εξαγόμενων χαρακτηριστικών απο την 1η εργαστηριακή άσκηση σε training και testing δεδομένα.

2.Το επιλεγμένο ψηφίο είναι k1=4 εφόσον (ΑΜ:03105644) αν και έχει ενσωματωθεί στην αρχή του κώδικα η δυνατότητα επιλογής επιθυμητού ψηφίου για εμφάνιση των χαρακτηριστικών. Εφόσον οι το πλήθος επαναλήψεων της mhmm_em.m εξαρτάται άμεσα απο το πλήθος των καταστάσεων και των γκαουσιανών σε κάθε κατάσταση , επιλέγουμε να δώσουμε σαν όριο επαναλήψεων το μέγιστο που ανεφέρεται στην εκφώνηση (Niter=15) και εμφανίζουμε τα αποτελέσματα για διαφορετικές τιμές NOS και NOG όπως φαίνεται και στο σχήμα 2.



Σχήμα 2 : Λογαριθμική πιθανοφάνεια για διαφορετικά μοντέλα

3. Σ'αυτό το μέρος για την εξαγωγή των αποτελεσμάτων απο τους ομιλητές των οποίων τα δεδομένα χρησιμοποιήθηκαν ως testing κατασκευάστηκε ένας 9x9 πίνακας Confusion Matrix. Συγκεκριμένα ο πίνακας ξεκινάει με όλα τα στοιχεία του μηδενικά και κάθε φορά που η εκφώνηση ενός ψηφίου *i* αναγνωρίζεται ώς ψηφίο *j* τότε το στοιχείο *CM(i,j)* του πίνακα αυξάνεται κατα "ένα". Έτσι ένα "τέλειο" μοντέλο θα μας έδινε έναν διαγώνιο πίνακα *CM.* Στην αρχή του κώδικα υπάρχει η δυνατότητα να επιλέξουμε πόσσες φορές θέλουμε να επαναληφθεί το πείραμα οπου σε κάθε επανάληψη γίνεται ένας νέος τυχαίος διαχωρισμός των δεδομένων training και testing set χωρίς ωστόσο να μηδενίζεται ο *CM* πίνακας σε κάθε επανάληψη του πειράματος. Στο τέλος λοιπόν για έναν ικανοποιητικό αριθμό επαναλήψεων (περίπου 10) μπορούμε να εξάγουμε ασφαλή και γενικευμένα αποτελέσματα.

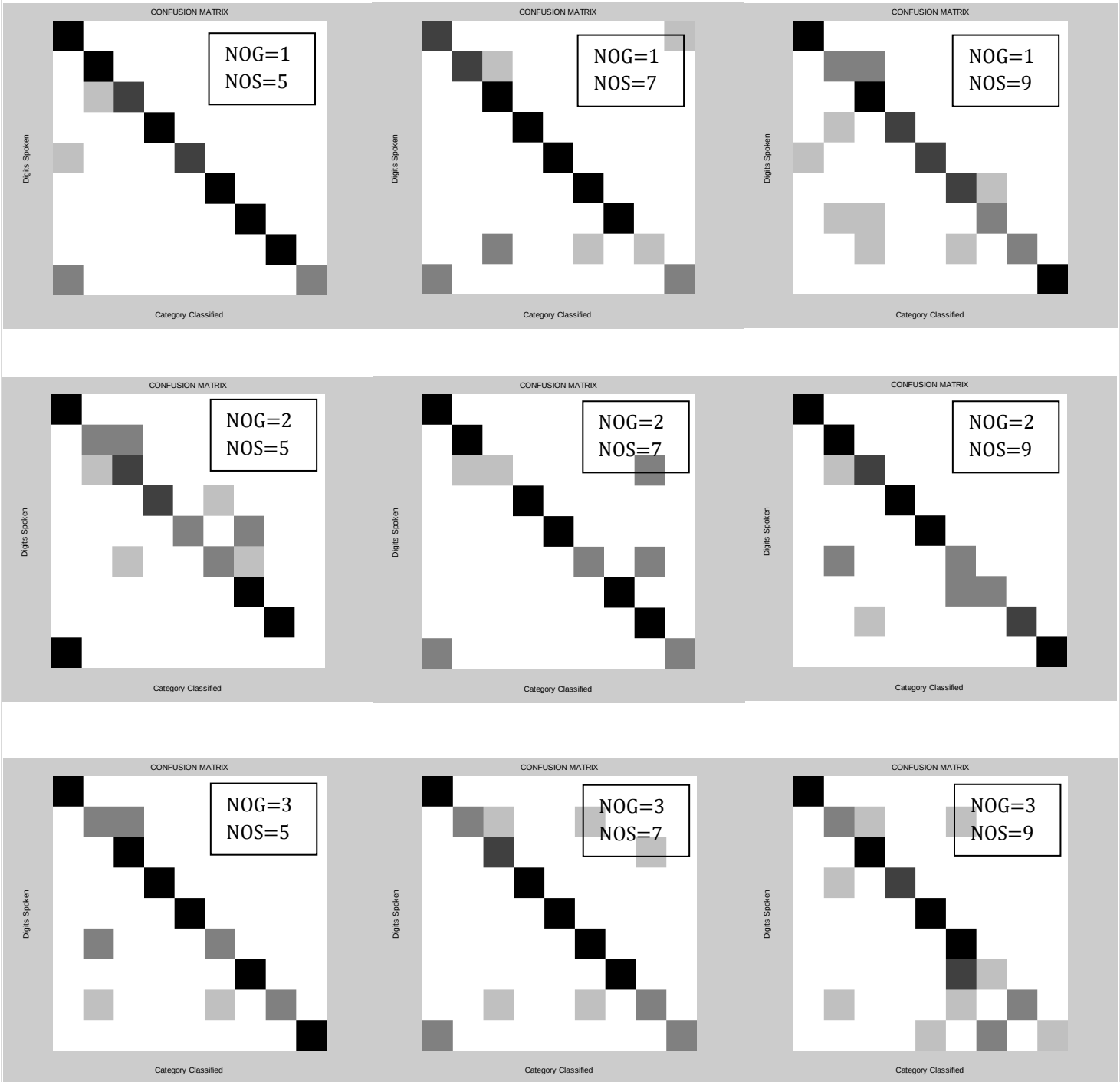
Η λογική με την οποία κατηγοριοποιούμε την εκφώνηση του *i* ψηφίου ως *k* είναι η εξής: Έχοντας τα mfcc δεδομένα της εκφώνησης του *i* ψηφίου τότε για κάθε μοντέλο *k* υπολογίζουμε με χρήση της συνάρτησης ***mhmm_logprob*** τη λογαριθμική πιθανοφάνεια του κάθε μοντέλου ως προς αυτή την εκφώνηση.

```
loglik(k) = mhmm_logprob(data, PRIOR{k}, TRANSMAT{k}, MU{k}, SIGMA{k}, MIXMAT{k});
```

Έτσι βρίσκουμε τη *j* στήλη στην οποία βρίσκεται η μέγιστη τιμή του 1x9 array loglik και κατηγοριοποιούμε την εκφώνηση του ψηφίου στην κατηγορία *j* αυξάνοντας το στοιχείο *CM(i,j)* κατα ένα. Να πούμε εδώ οτι για την καλύτερη οπτικοποίηση των αποτελεσμάτων κατασκευάζουμε μια grayscale εικόνα *A* απο τον παρακάτω κώδικα.

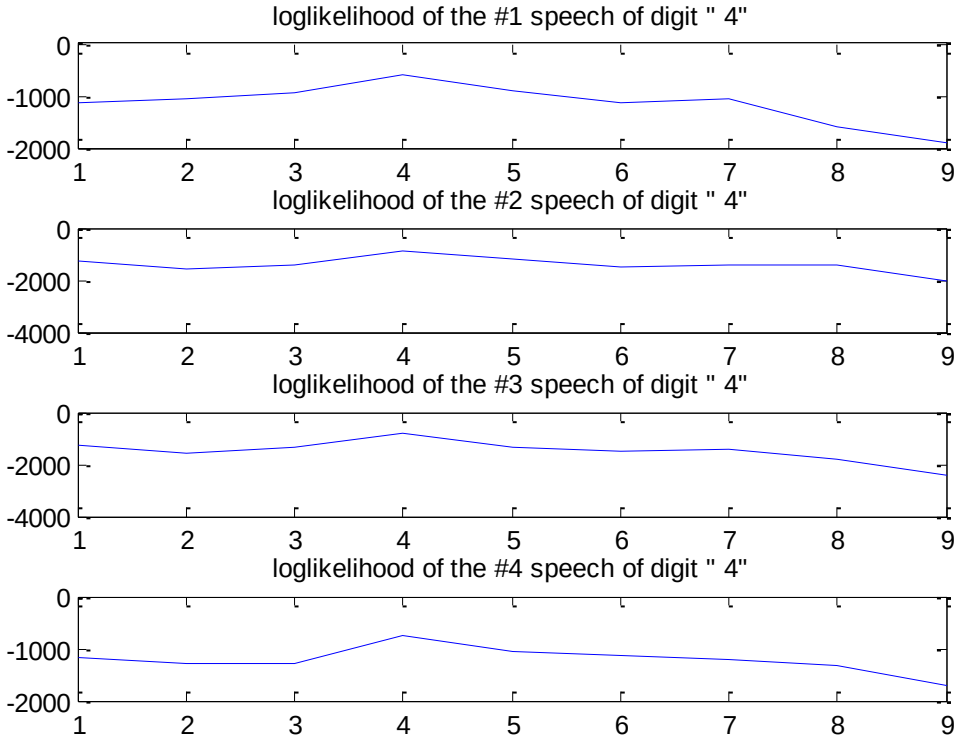
```
A = CM/max(CM(:));        %make values from zero to one
A = 1-A;                  %take the complement for better view
```

Πλέον με χρήση της ***imshow*** (και της ***imresize*** με τη μέθοδο 'nearest') μπορούμε να δούμε τα αποτελέσματα ώς μια εικόνα όπου ιδανικά θα θέλαμε όλα τα διαγώνια μπλόκ στοιχεία να είναι μαύρα και όλα τα υπόλοιπα λευκά. Οι παρακάτω πίνακες του σχήματος 3 εξήχθησαν για 1 επανάληψη του κάθε πειράματος και με διαφορετικό πλήθος καταστάσεων και γκαουσιανών που περιγράφουν την κάθε κατάσταση σε κάθε πείραμα. Σε κάθε περίπτωση το Niter της εκπαιδευσης επιλέχθηκε "15".



Σχήμα 3 : Confusion Matrixes για διαφορετικές τιμές καταστάσεων και gaussians ανα κατάσταση

Επίσης για το ψηφίο k1=4 και κάθε εκφώνηση αυτού στο testing set μπορούμε να παρατηρήσουμε το 1x9 array loglik βάσει του οποίου όπως αναφέραμε έγινε η κατηγοριοποίηση. Τυχαία επιλέγουμε για την απεικόνιση NOG=1 και NOS=5.



Σχήμα 4 : Λογαριθμική πιθανοφάνεια των εκφωνήσεν του ψηφίου "4" για κάθε μοντέλο

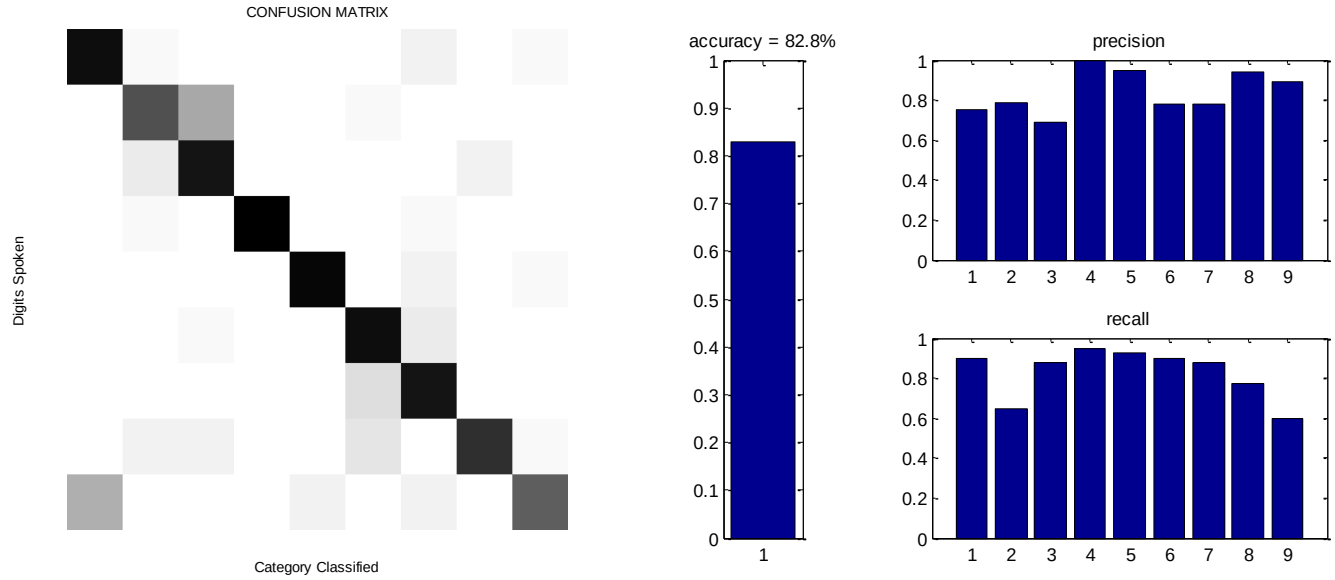
Απο τα παραπάνω διαγράμματα παρατηρούμε οτι πράγματι το μοντέλο #4 δίνει μεγαλύτερο "σκόρ" και μάλιστα στη συγκεκριμένη περίπτωση όπως παρατηρήθηκε και απο τον Confusion Matrix του αντίστοιχου πειράματος στο σχήμα 3, όλες οι εκφωνήσεις κατηγοριοποιήθηκαν σωστά. Βλέπουμε όμως οτι και άλλα μοντέλα δίνουν υψηλές τιμές για κάποιες εκφωνήσεις. Για παράδειγμα στη δεύτερη και στην τρίτη εκφώνηση του μοντέλο του ψηφίου "1" δίνει αρκετά υψηλή πιθανοφάνεια και θα μπορούσε να παρουσιαστεί "σύγχυση" σε περίπτωση που το "σκόρ" ήταν λίγο μεγαλύτερο.

Απο τους πίνακες του σχήματος 3 μπορούμε να παρατηρήσουμε οτι για μια μόνο επανάληψη του πειράματος δεν μπορούμε να βγάλουμε ασφαλή συμπεράσματα για το ποιές επιλογές παραμέτρων NOS και NOG είναι οι βέλτιστες άν και φαίνεται οτι το μοντέλο με NOG=1 και NOS=5 δεν δίνει άσχημα αποτελέσματα. Για να μπορούμε να έχουμε πιο ασφαλή συπεράσματα επαναλαμβάνουμε το πείραμα πολλές φορές και παρουσιάζουμε κάποιους απο τους συνολικούς Confusion Matrixes για διαφορετικές παραμέτρους. Επίσης υπολογίζουμε για κάθε περίπτωση την ακρίβεια (accuracy) ως το πηλίκο των σωστά κατηγοριοποιημένων δειγμάτων προς τα συνολικά δείγματα αλλα και δύο άλλες παραμέτρους οι οποίες έχουν σημαντικό ρόλο σε προβλήματα αναγνώρισης , precision και recall οπου:

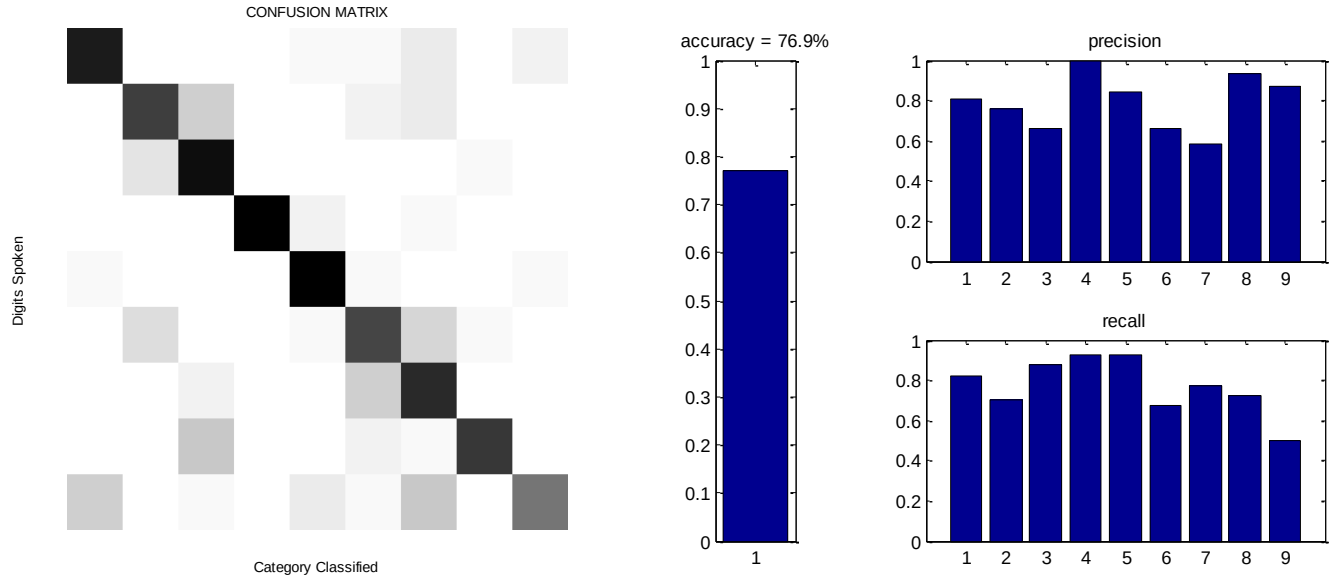
$$precision(i) = \frac{\Sigma\omega\sigma\tau\acute{\alpha}\ \tau\alpha\xi\iota\nu o\mu\eta\mu\acute{\epsilon}\nu\alpha\ \psi\eta\phi\acute{\iota}\alpha\ \sigma\tau\eta\nu\ \kappa\alpha\tau\eta\gamma o\rho\acute{\iota}\alpha(i)}{\Sigma\acute{\upsilon}\nu o\lambda o\ \tau\alpha\xi\iota\nu o\mu\eta\mu\acute{\epsilon}\nu\omega\nu\ \psi\eta\phi\acute{\iota}\omega\nu\ \sigma\epsilon\ \alpha\upsilon\tau\acute{\eta}\nu\ \tau\eta\nu\ \kappa\alpha\tau\eta\gamma o\rho\acute{\iota}\alpha(i)}$$

$$recall(i) = \frac{\Sigma\omega\sigma\tau\acute{\alpha}\ \tau\alpha\xi\iota\nu o\mu\eta\mu\acute{\epsilon}\nu\alpha\ \psi\eta\phi\acute{\iota}\alpha\ \sigma\tau\eta\nu\ \kappa\alpha\tau\eta\gamma o\rho\acute{\iota}\alpha(i)}{\Sigma\acute{\upsilon}\nu o\lambda o\ \epsilon\kappa\phi\omega\nu\acute{\eta}\sigma\epsilon\omega\nu\ \tau o\upsilon\ i\ \psi\eta\phi\acute{\iota}o\upsilon\ \sigma\tau o\ testing\ set}$$

Έτσι για 10 επαναλήψεις του πειράματος με τυχαίο διαχωρισμό training και testing set σε κάθε επανάληψη παρουσιάζουμε μερικούς απο τους Confusion Matrixes και τις τιμές accuracy, precision και recall.

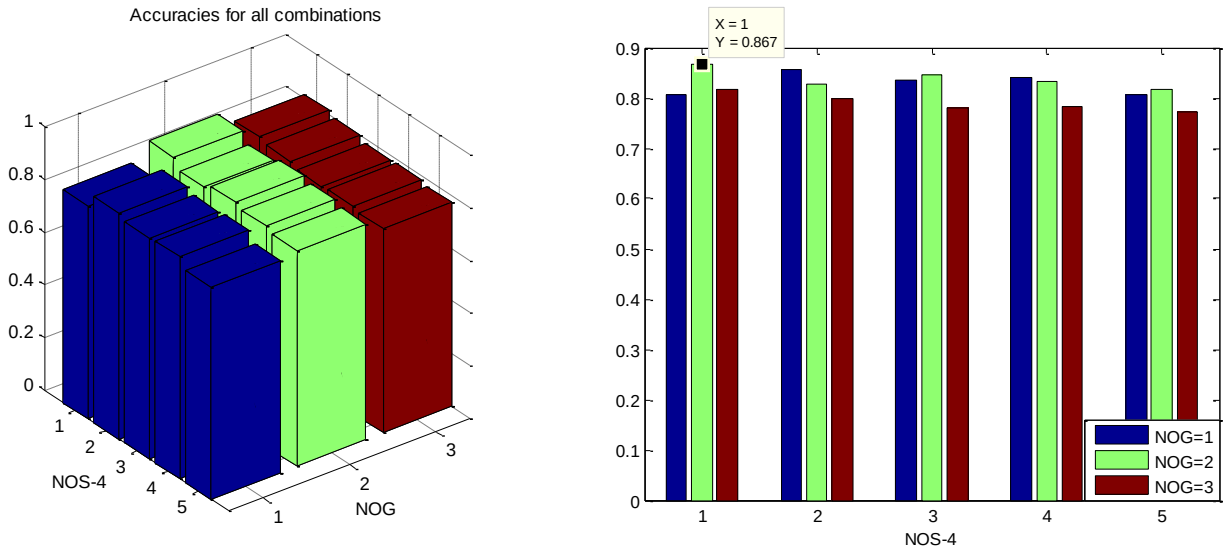


Σχήμα 5 : Αποτελέσματα για NOG=1 και NOS=5 και 10 επαναλήψεις του πειράματος



Σχήμα 6 : Αποτελέσματα για NOG=3 και NOS=9 και 10 επαναλήψεις του πειράματος

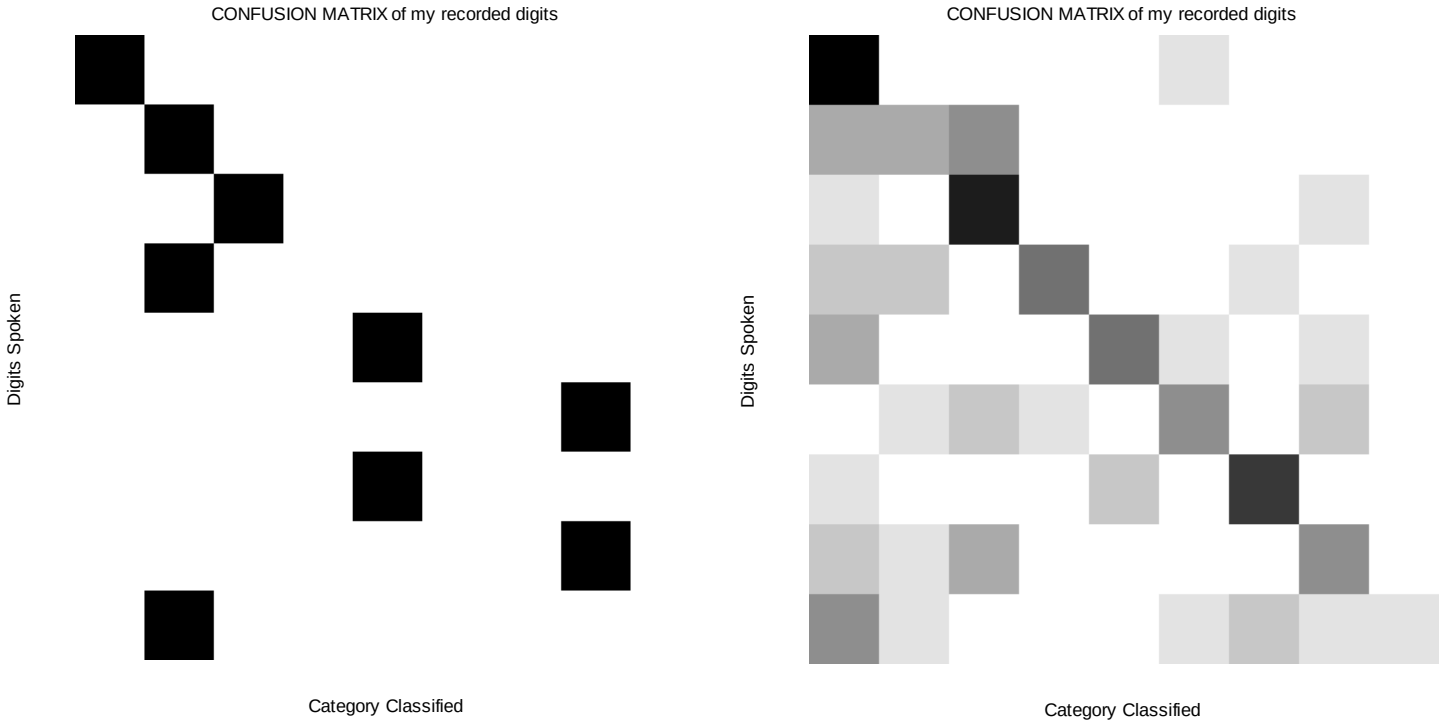
Στα παραπάνω σχήματα όπως είναι αντιληπτό παρουσιάζουμε τα αποτελέσματα για τις ακραίες τιμές των παραμέτρων. Βλέπουμε παρόλα αυτά οτι στην πρώτη περίπτωση τα αποτελέσματα είναι αισθητά πιο ικανοποιητικά. Μάλιστα το ψηφίο "4" και στις δύο περιπτώσεις είχε τέλειο precision και αρκετά υψηλο recall. Στη συνέχεια κατασκευάστηκε script το οποίο υπολόγησε την ακρίβεια που είχαμε για κάθε πιθανό συνδυασμό NOG και NOS μέσα στο πεδίο τιμών που αναφέρεται στην εκφώνηση της άσκησης. Τα αποτελέσματα αυτά αποθηκέυτηκαν στο αρχείο ACCURACY _ALL.mat και τα συνολικά αποτελέσματα παρουσιάζονται στο σχήμα 7 με χρήση της συνάρτησης ***bar3*** και ***bar***.



Σχήμα 7 : Διαγράμματα acuuracy για κάθε συνδυασμό NOS και NOG. Σημείωνεται οτι στα παραπάνω διαγράμματα ο πραγματικός αριθμός καταστάσεων είναι ο αναγραφόμενος + 4

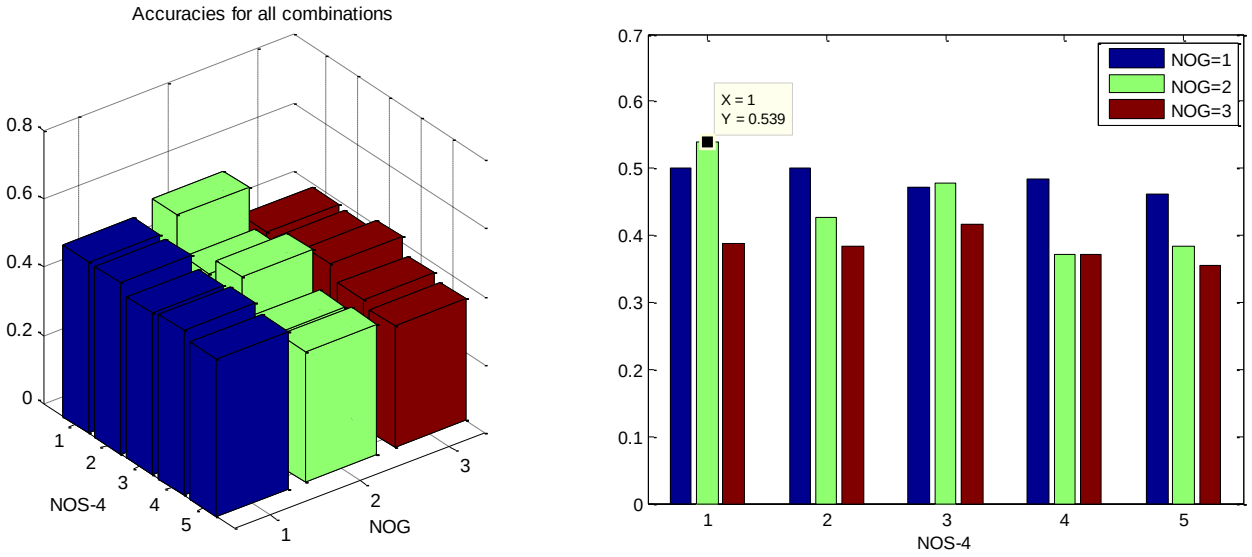
Απο τα παραπάνω διαγράμματα είναι εμφανές οτι δεν υπάρχουν πολύ μεγάλες αποκλίσεις μεταξύ των μοντέλων ωστόσο η μέση ακρίβεια 86.7% που επιτεύχθηκε απο μοντέλο με 5 καταστάσεις και 2 gaussians ανα κατάσταση είναι αρκετά ικανοποιητική.

Στη συνέχεια έγινε ο έλεγχος των μοντέλων με τα εξαγόμενα δεδομένα απο την ηχογράφηση των ψηφίων απο εμένα τον ίδιο, δεδομένα τα οποία είναι αποθηκευμένα στα cells του MGV.mat. Παρακάτω παρουσιάζουμε τον CM για μια επανάληψη και για δέκα επαναλήψεις του πειράματος χρησιμοποιώντας τις βέλτιστες παραμέτρους που εξήχθησαν παρατηρώντας το σχήμα 7.



Σχήμα 8 : Confusion Matrixes για μια (αριστερά) και 10 (δεξιά) επαναλήψεις του πειράματος με δικά μας δεδομένα και NOS=5 , NOG=2

Απο το σχήμα 8 παρατηρούμε πλέον οτι τα αποτελέσματα είναι απογοητευτηκά όταν εξετάστηκαν τα δικά μας δεδομένα σαν test. Συγκεκριμένα για μια επανάληψη κατηγοριοποιήθηκαν σωστά μόνο 5 απο τα 9 ψηφία ενώ για 10 επαναλήψεις μετρήθηκε ακρίβεια 51.1%. Ο λόγος που συμβαίνει κάτι τέτοιο είναι οτι πιθανότατα οι συνθήκες ηχογράφησης ήταν πολύ διαφορετικές, (διαφορετικό μικρόφωνο, θόρυβος περιβάλλοντος και ποιότητα συστήματος ηχογράφησης) και το σύστημα λόγω του μικρού αριθμού ατόμων που χρησιμοποιήθηκαν για εκπαίδευση δεν είναι αρκετά εύρωστο ωστε να καλύψει τις γλωσσικές ιδιαιτερότητες του κάθε ατόμου. Στη συνέχεια παρουσιάζουμε τα αποτελέσματα και για όλες τις τιμές των παραμέτρων με έλεγχο των δικό μας δεδομένων και επανάληψη του πειράματος 20 φορές.



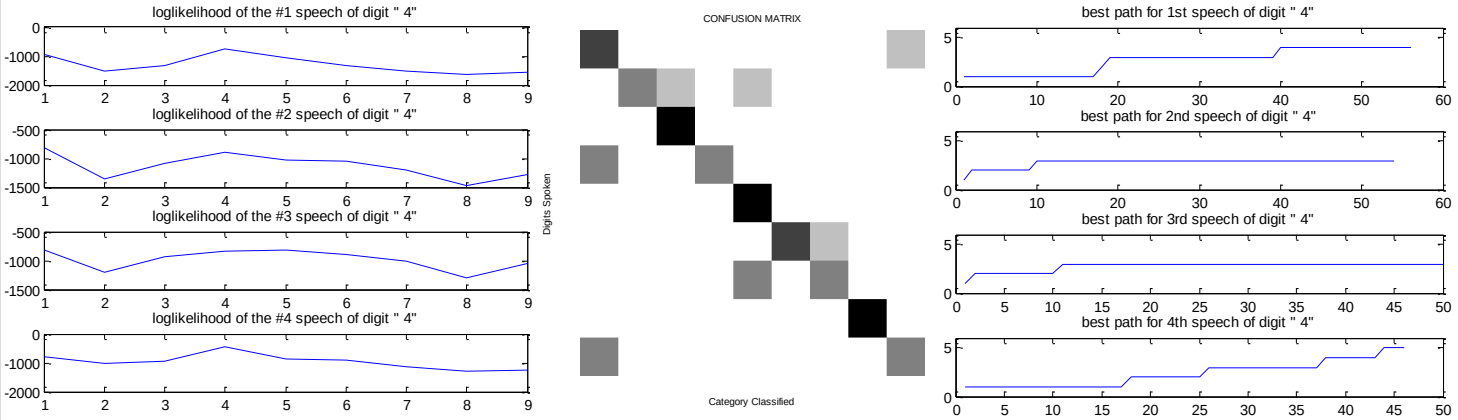
Σχήμα 9 : Διαγράμματα acuuracy για κάθε συνδυασμό NOS και NOG με έλεγχο της δική μου εκφώνησης

Και πάλι βλέπουμε οτι και στα δικά μας δεδομένα η βέλτιστη μέση ακρίβεια παρατηρήθηκε για μοντέλα 5 καταστάσεων με 2 γκαουσιανές ανα κατάσταση. Βέβαια το μέσο ποσσοστό είναι αρκετά μικρότερο απο αυτό των δεδομένων της άσκησης αλλα αυτό οφείλεται κατα κύριο λόγο παράγοντες τους οποίους αναφέραμε ήδη.

4. Στο τελευταίο μέρος παρουσιάζουμε τη βέλτιστη ακολουθία καταστάσεων για τον έλεγχο των εκφωνήσεων των ψηφίων "4". Ως πιο πιθανή ακολουθία εννοούμε την ακολουθία η οποία επιστρέφει και τη μεγαλύτερη λογαριθμική πιθανότητα, δηλαδή το μεγαλύτερο "σκόρ". Κάτι τέτοιο ενσωματώνεται στον κώδικα με χρήση της συνάρτησης ***viterbi_path*** αφού πρώτα υπολογιστούν οι πίνακες παρατήρησης: $B(i,t) = P(y(t) \mid q(t)=i)$. Έτσι γνωρίζοντας την απόφαση που λήφθηκε για την κατηγοριοποίηση της κάθε *j* εκφώνησης του ψηφίου "4" , οι παρακάτω δύο γραμμές κώδικα περιγράφουν τον τρόπο εύρεσης τηε βέλτιστης ακολουθίας.

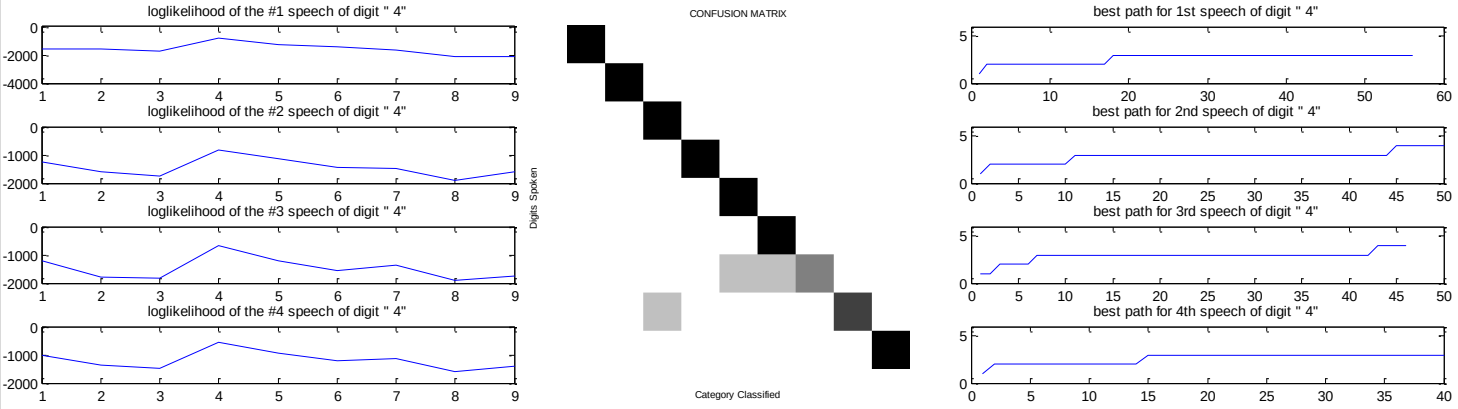
```
B = mixgauss_prob(data, MU{decision}, SIGMA{decision}, MIXMAT{decision});
path{j} = viterbi_path(PRIOR{decision}, TRANSMAT{decision}, B);
```

Η μεταβλητή decision όπως είδαμε και απο τα προηγούμενα μέρη ουσιαστικά λαμβάνει μια τιμή απο 1-9 ανάλογα με το που κατηγοριοποιήθηκε η *j* εκφώνηση του ψηφίου. Τελικά στα στοιχεία του cell path βρίσκονται οι πιθανότερες ακολουθίες καταστάσεων. Για κάθε μια επανάληψη του πειράματος θα παρουσιάζουμε τις ακολουθίες αυτές μαζί με τους confusion matrixes και τις λογαριθμικές πιθανοφάνειες - "σκόρ" που επέστρεψε το κάθε μοντέλο για NOG=1 και NOS=5.



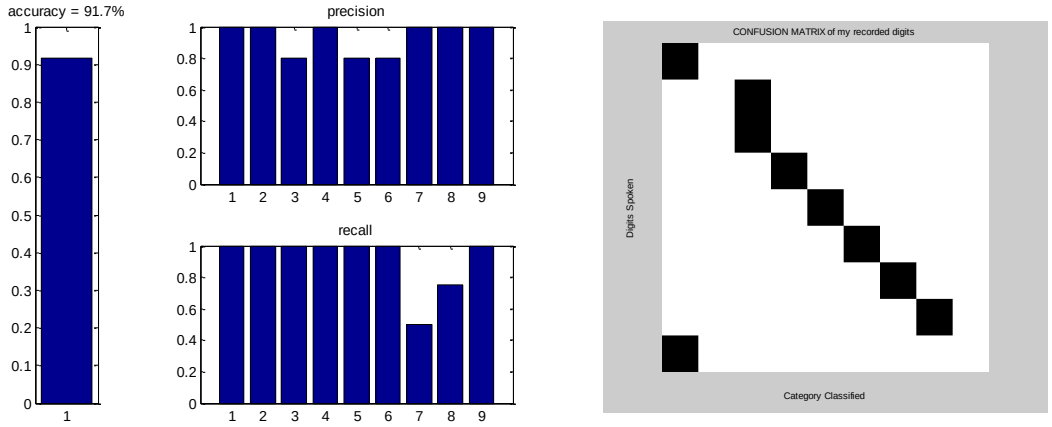
Σχήμα 10 : αριστερά: log likelihood του κάθε μοντέλου , κέντρο: confusion matrix
δεξιά: βέλτιστη ακολουθία καταστάσεων για τις εκφωνήσεις του "4" (NOG=1, NOS=5)

Απο το παραπάνω σχήμα το ζητούμενο είναι τα δεξιότερα διαγράμματα που εμφανίζουν για κάθε εκφώνηση το βέλτιστο μονοπάτι. Αν για παράδειγμα παρατηρήσουμε τον πρώτο εκφωνητή αρχικά οι κατάσταση είναι "1" στη συνέχεια περίπου μετά απο 18 χρονικές μονάδες (ουσιαστικά μετά απο 18 χρονικά παράθυρα) αλλάζει στην κατάσταση "2" και στο αμέσως επόμενο στην κατάσταση "3" και τέλος στις 40 χρονικές μονάδες στην κατάσταση "4" και παραμένει εκεί. Λίγο πολύ την ίδια σειρά παρατηρούμε και στον τελευταίο ομιλητή με τη διαφορά οτι στο τέλος αλλάζει και στην 5η κατάσταση. Βλέπουμε οτι στους δύο μεσαίους ομιλητές ωστόσο οι ακολουθίες καταστάσεων μοιάζουν αρκετά μεταξύ τους και είναι διαφορετικές απο τις δύο άλλες. Παρατηρώντας τώρα τον Confusion Matrix καταλαβαίνουμε πράγματι απο ο μέσο γκρί χρώμα των μπλόκ της τέταρτης γραμμής οτι οι δύο εκφωνήσεις κατηγοριοποιήθηκαν σωστά στο τέταρτο μοντέλο ενώ οι υπόλοιπες δύο στο πρώτο μοντέλο. Στη συνέχεια στο αριστερότερο διάγραμμα επιβεβαιώνουμε και οπτικά οτι στους δύο μεσαίους ομιλητές η μέγιστη λογαριθμική πιθανοφάνεια είναι μεγαλύτερη στο μοντέλο "1" απ'ότι στο μοντέλο "4" και γι'αυτό οι δύο αυτές εκφωνήσεις αναγνωρίστηκαν ως εκφωνήσεις του ψηφίου "1". Έτσι μπορούμε τώρα να πούμε οτι η μορφή των βέλτιστων μονοπατιών του 1ου και του 4ου ομιλητή, οπου ουσιαστικά έχουμε μια αλλαγή καταστάσεων απο την πρώτη εως την τέταρτη ή και την πέμπτη κατάσταση προκύπτουν απο 4ο μοντέλο ενώ οι δύο μεσαίες ακολυθίες προκύπτουν απο το 1ο μοντέλο. Στη συνέχεια επαναλαμβάνουμε το πείραμα για 5 καταστάσεις και 2 gaussian ανα κατάσταση.



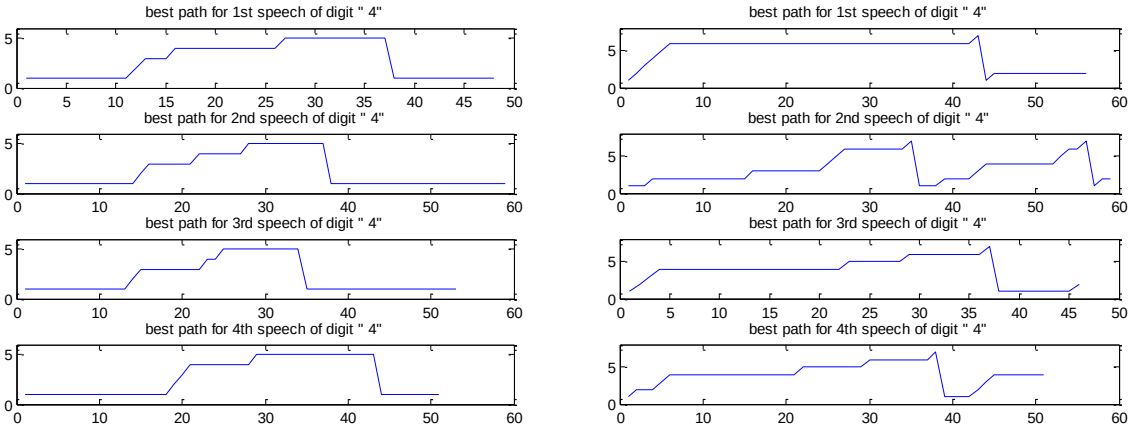
Σχήμα 11 : αριστερά: log likelihood του κάθε μοντέλου , κέντρο: confusion matrix
δεξιά: βέλτιστη ακολουθία καταστάσεων για τις εκφωνήσεις του "4" (NOG=2, NOS=5)

Να αναφέρουμε εδώ οτι σε κάθε ένα απο τα δύο παραδείγματα ο αριθμός απο 1-5 που περιγράφει την κάθε στατική κατάσταση μπορεί να διαφέρει. Έτσι για παράδειγμα ενω στο σχήμα 10 η κατάσταση 1 μπορεί να περιγράφει την "στατική" κατάσταση της σιωπής (ή γενικότερα παραθύρων με χαμηλό ενεργειακό περιεχόμενο), εδώ μπορέι να περιγράφεται απο άλλη κατάσταση, π.χ την 2η κατάσταση. Έτσι δεν πρέπει να αναμένουμε τις ίδιες ακολουθίες κάθε φορά που επαναλαμβάνουμε το πείραμα (training-testing), εν τούτοις θα θέλαμε να έχουν κοινές δομές- πορείες οι καταστάσεις των ομιλητών στον χρόνο για κάθε πείραμα ξεχωριστά. Στην προκειμένη περίπτωση κάτι τέτοιο ισχύει και μάλιστα όλοι οι ομιλητές της εκφώνησης του "4" κατηγοριοποιήθηκαν σωστά. Μάλιστα στο παραπάνω πείραμα πετύχαμε συνολική ακρίβεια 91.7% ενώ αντίστοιχα πολύ καλά αποτελέσματα είχαμε και με τον έλεγχο των δικών μας δεδομένων όπως φαίνεται στο παρακάτω σχήμα.



Σχήμα 12: αριστερά: αποτελέσματα απο το μεμονωμένο πείραμα του σχήματος 11, δεξιά: confusion matrix απο τα δικά μας δεδομένα χρησιμοποιώντας τα εξαγόμενα μοντέλα του παραπάνω πειράματος

Στη συνέχεια θα πειραματιστήκαμε με διαφορετικές τοπολογίες μοντέλων. Αρχικά λάβαμε υπόψιν την περίπρωση ώντας στην τελευταία κατάσταση(π.χ 5) να μπορεί να μεταβεί το σύστημα στην πρώτη κατάσταση. Δηλαδή $a_{end,1}=0.5$ στην περίπτωσή μας. Τα αποτελέσματα για δυο διαφορετικές επαναλήψεις με διαφορετικές παραμέτρους μοντέλων φαίνονται στο παρακάτω σχήμα.



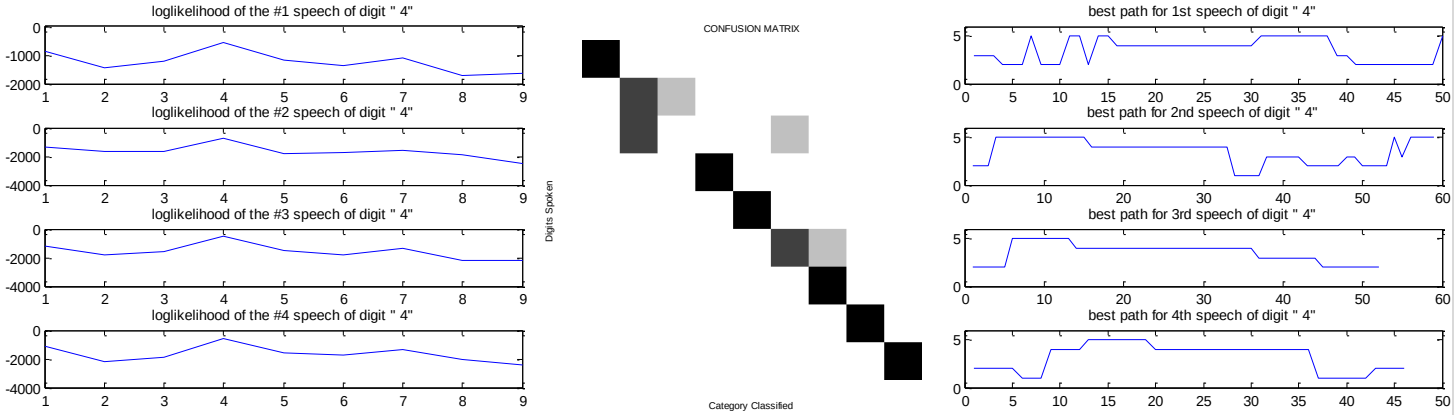
Σχήμα 13: τοπολογία με εναλλακτικό transition matrix, αριστερά: NOS=2, NOG=5, δεξιά: NOS=2, NOG=7

Σημειώνουμε οτι στις δυο παραπάνω περιπτώσεις είχαμε σωστές κατηγοριοποιήσεις του ψηφίου"4" και γενικά καλή ακρίβεια πάνω απο 80%.

Στο τέλος έγινε εκπαίδευση με εντελώς τυχαίους πίνακες αρχικών πιθανοτήτων με χρήση των συναρτήσεων normalise και randperm.

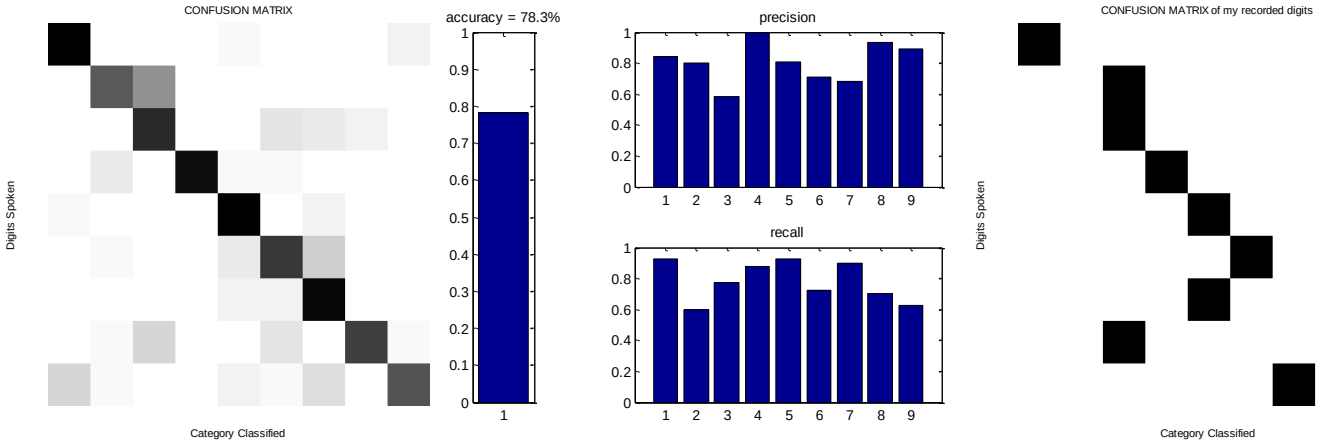
```
P = normalise(rand(NOS,1));
T = mk_stochastic(rand(NOS,NOS));
```

Τα αποτελέσματα για μια επανάληψη του πειράματος με NOG=2 και NOS=5 φαίνονται παρακάτω.



Σχήμα 14 : αριστερά: log likelihood του κάθε μοντέλου , κέντρο: confusion matrix
δεξιά: βέλτιστη ακολουθία καταστάσεων για τις εκφωνήσεις του "4" (NOG=2, NOS=5)

Παρατηρούμε οτι για ενα μεμονωμένο πείραμα τα αποτελέσματα δεν είναι άσχημα. Στη συνέχεια επαναλάβαμε το πείραμα 10 φορές για να υπολογίσουμε μια μέση ακρίβεια. Επίσης έγινε έλεγχος σε δικά μας δεδομένα για μια επανάληψη και τα αποτελέσματα φαίνονται στον παρακάτω σχήμα.

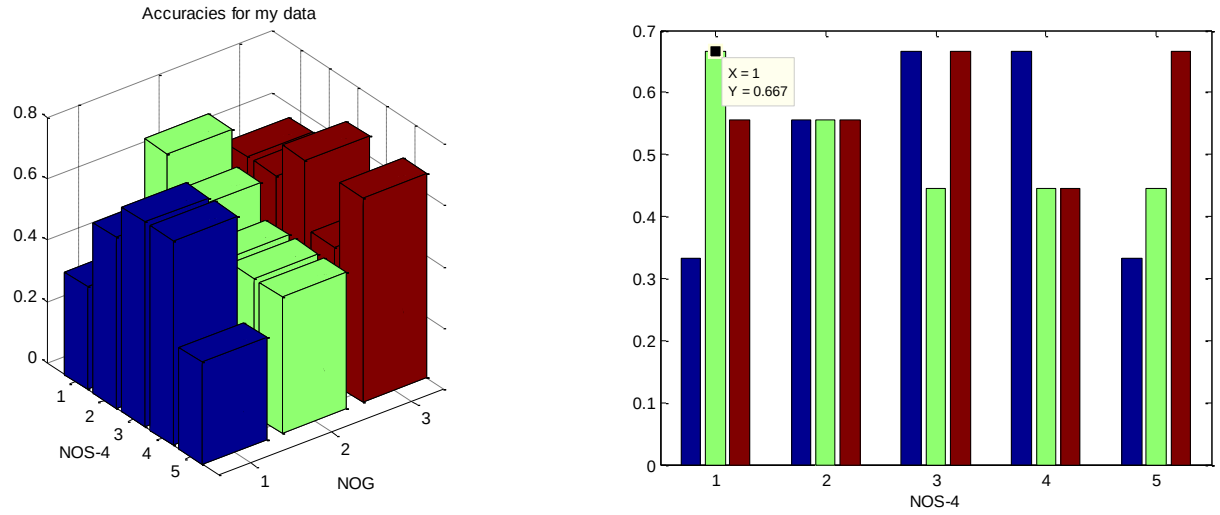


Σχήμα 15 : αριστερά: confusion matrix απο δεδομένα βάσης για 10 επαναλήψεις , κέντρο : αποτελέσματα για το testing των δεδομένων αυτών, δεξιά: confusion matrix για δικά μας δεδομένα και 1 επανάληψη

Παρατηρούμε οτι τα αποτελέσματα της απόδωσης είναι εμφανώς μικρότερα απο αυτά που φαίνονται στο σχήμα 7 τα οποία εξήχθησαν απο την αρχική τοπολογία μοντέλων που επιλέξαμε.

Συμπεράσματα

Γνωρίζουμε οτι τα κρυφά μαρκοβιανά μοντέλα είναι ίσως το πιο ισχυρό εργαλείο σε θέματα αναγνώρισης προτύπων όταν στο πρόβλημα υπεισέρχεται η έννοια του χρόνου. Στην εργασία αυτή μελετήθηκαν κατα κύριο λόγο left-right τοπολογίες μοντέλων με δυνατότητα μετάβασης μόνο σε διαδοχικές καταστάσεις.Ο βέλτιστος συνδιασμός καταστάσεων και γκαουσιανών που περιγράφουν κάθε κατάσταση μετά απο 10 επαναλήψεις του πειράματος (και 20 επαναλήψεις πάνω σε δικά μας δεδομένα) εκτιμήθηκε πως είναι 5 καταστάσεις και 2 γκαουσιανές που περιγράφουν την συγκέντρωση των δεδομένων κάθε κατάστασης. Δεδομένου του μικρού πλήθους ατόμων που είχαμε στη διάθεσή μας για εκπαίδευση του συστήματος τα αποτελέσματα είναι αρκετά ικανοποιητικά. Παρατηρήθηκαν μικρότερες αποδόσεις πάνω στα δικά μας δεδομένα ως testing set αλλα αυτό μπορεί να οφείλεται σε διάφορους λόγους όπως αναφέρθηκε στη διαδικασία της άσκησης. Για να μπορέσουμε να αποφανθούμε πιο ορθά πάνω σε αυτό το θέμα χρησιμοποιήσαμε σε τελευταία φάση όλα τα δεδομένα των ομιλητών του πίνακα DATA.mat για εκπαίδευση και ο έλεγχος έγινε πάνω σε δικά μας δεδομένα.



Σχήμα 16 : υπολογισμών ακρίβειας για όλους τους συνδιασμούς NOG και NOS με έλεγχο στα δικά μας δεδομένα

Έτσι παρατηρούμε οτι η ακρίβεια αυξήθηκε αισθητά (κατα 10% περίπου) σε κάποιους συνδυασμούς. Αναμένουμε λοιπόν με πιθανή αύξηση του training set να είχαμε πολυ καλύτερα αποτελέσματα και πάνω στα δικά μας δεδομένα. Οι μεγάλες αποκλίσεις στα παραπάνω διαγράμματα απο μοντέλο σε μοντέλο οφείλονται στο οτι το πείραμα έγινε μόνο μία φορά καθώς δεν υπήρχε λόγος να επαναληφθεί αφού χρησιμοποιήθηκαν όλοι οι ομιλητές για την εκπαίδευση και ο έλεγχος (μετα απο επανεκπαίδευση) θα μας έδινε ακριβώς τα ίδια αποτελέσματα.