«ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ»

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ &
ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Αναγνώριση Προτύπων με Έμφαση στην Αναγνώριση Φωνής

1η Εργαστηριακή Άσκηση

9ο Εξάμηνο

Αθανασίου Νικόλαος

03112074

Σκόπος της συνέχειας της πρώτης εργαστηριακής άσκησης είναι η υλοποίηση ενός συστήματος αναγνώρισης ψηφίων.Κατά τα υπόλοιπα 5 βήματα της άσκησης κατασκευάζουμε διάφορα είδη ταξινομητών και στη συνέχεια επιχειρούμε να τους συνδυάσουμε για να πετύχουμε βέλτιστα αποτελέσματα.

Βήμα 10

Για τον υπολογισμό των a-priori πιθανοτήτων για κάθε κατηγορία εκτελούμε την εξής διαδικασία: Αρχικά υπολογίζουμε το πλήθος των δεδομένων από κάθε κατηγορία και στη συνέχεια διαιρούμε το πλήθος αυτό με το συνολικό πλήθος των δεδομένων. Το αποτέλεσμα που προκύπτει αντιστοιχεί στις a-priori πιθανότητες της κάθε κατηγορίας. Οι a-priori πιθανότητες που υπολογίσαμε είναι οι ακόλουθες:

0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0.1638	0.1378	0.1003	0.0902	0.0894	0.0763	0.0911	0.0885	0.0743	0.0883

Παρατηρούμε ότι το άθροισμα των πιθανοτήτων αυτών ισούται με μονάδα όπως αναμέναμε.

Βήμα 11

Στο βήμα αυτό χρησιμοποιούμε τις apriori πιθανότητες που υπολογίσαμε στο βήμα 10 για να κατασκευάσουμε των ταξινομητή Bayes οποίος κατατάσσει σύμφωνα με την posteriori πιθανότητα.Πιο συγκεκριμένα κατατάσσει με βάση τη σχέση:

$$digit = \underset{digit}{\operatorname{argmax}} P(x|digit) \cdot P(digit)$$

Για τον υπολογισμό της δεσμευμένης πιθανότητας στην παραπάνω σχέση θεωρούμε ότι οι παρατηρήσεις μας ακολουθούν κανονική κατανομή και για μεγαλύτερη ακρίβεια λογαριθμίζοντας το όρισμα του max και χρησιμοποιώντας την ιδιότητα του λογαρίθμου => log(x1*x2)=logx1+logx2 καταλήγουμε στην ταξινόμηση με βάση την παρακάτω σχέση για κάθε ψηφίο των train δεδομένων.

$$\underset{digit}{\operatorname{argmax}}(\sum_{i=1}^{256}\log(P(x_i|digit) + \log P(digit))$$

Σύμφωνα με τη διαδικασία αυτή λάβαμε ποσοστό επιτυχίας:

Total success rate: 0.705531

Και στα επιμέρους αποτελέσματα για κάθε ψηφίο παρατήρησα όπως και στην προσπαρασκευή μεγάλη επιτυχία στην ααναγνώριση του 1 του 6 και του 9 που είναι σχετικά εύκολα ψηφία και μικρότερα στα υπόλοιπα γεγονός που μπορεί να οφείλεται και σε πολλούς διαφορετικούς τρόπους γραφής αλλά και στο ότι είναι πιο πολύπλοκα σε σχέση για παράδειγμα με το 1 που είναι μια γραμμή και εύκολα αναγνωρίζεται.

Βήμα 12

Στην περίπτωση της ενιαίας διασποράς λάβαμε αποτέλεσμα:

Total success rate: 0.812656

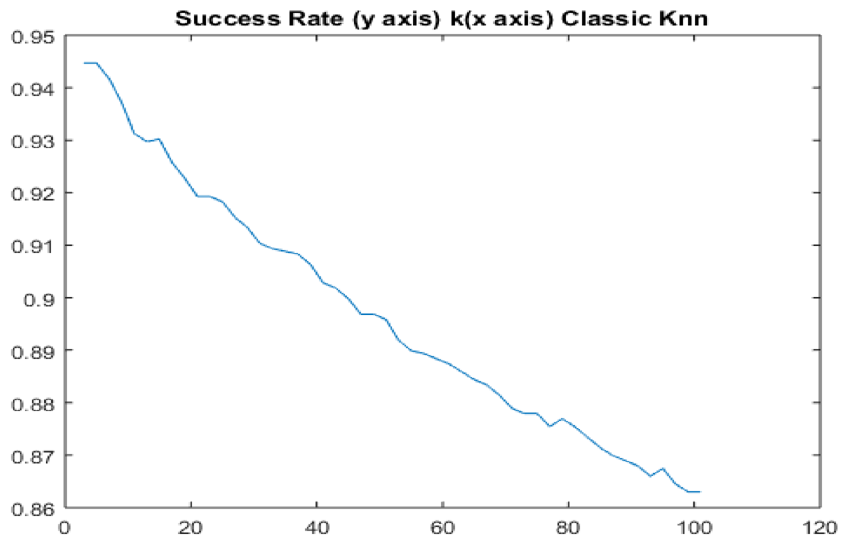
Σε αυτό το σημείο ήταν απαραίτητη η τροποποίηση των τυπικών αποκλίσεων κατά μία πολύ μικρή ποσότητα καθώς σε ορισμένες περιπτώσεις όπου η τυπική απόκλιση ήταν μηδενική, δεν οριζόντουσαν κάποιες πράξεις. Συνεχίζοντας τους πειραματισμούς με την τυπική απόκλιση καταλήξαμε στο συμπέρασμα πως όταν θέτουμε όλες τις τυπικές αποκλίσεις ίσες με τη μονάδα, τα αποτελέσματα παρουσιάζουν σημαντική βελτίωση. Αυτό συμβαίνει επειδή ορισμένα χαρακτηριστικά επηρέαζαν σημαντικά τη τελική a-posteriori πιθανότητα τα οποία είχαν πολύ μικρή διακύμανση και ίδια τιμή με την εικόνα μας και δεν κουβαλούσαν πληροφορία (πχ λευκές περιοχές γύρω από τους αριθμούς.) και λόγω της κανονικής κατανομής που χρησιμοποιούμε.

Βήμα 13,14

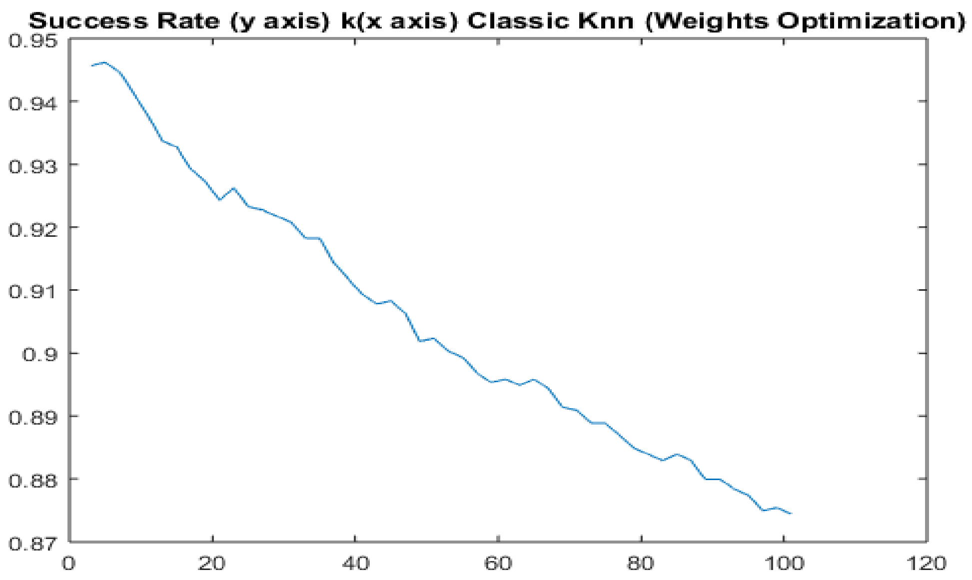
Για την υλοποίηση του NNR-k αλγορίθμου αρχικά υπολογίζουμε την Ευκλείδεια απόσταση ανάμεσα στο στοιχείο προς αναγνώριση και σε όλα τα δεδομένα. Στη συνέχεια ταξινομούμε τις αποστάσεις αυτές σε αύξουσα σειρά και επιλέγουμε τις k πρώτες αποστάσεις. Έπειτα βρίσκουμε σε ποιες κατηγορίες αντιστοιχούν οι αποστάσεις αυτές και τέλος ποια από αυτές τις κατηγορίες εμφανίζεται περισσότερες φορές όπου και ταξινομούμε το άγνωστο στοιχείο σε αυτή την κατηγορία. Παρατηρούμε ότι η μέγιστη επίδοση επιτυγχάνεται για k=3, k=5. Παρακάτω εμφανίζονται τα αποτελέσματα για διάφορες τιμές του k

Πρώτα 1000 τεστ δεδομένα κ = 1	Όλα τα δεδομένα κ = 1	Όλα τα δεδομένα κ = 3	Όλα τα δεδομένα κ = 5	Όλα τα δεδομένα κ = 7	Όλα τα δεδομένα κ =15
0.9000	0.9437	0.9447	0.9447	0.9417	0.9302

Παρατηρούμε όπως φαίνεται και από το παρακάτω διάγραμμα υπάρχει ένα κατώφλι πάνω από το οποίο τα πράγματα αντί να βελτιώνονται χειροτερεύουν όπως παρατηρούμε και από τον πίνακα.Συγκεκριμένα παρατηρούμε ότι τα καλύτερα αποτελέσματα λαμβάνονται για k=3.



Στην συγκεκριμένη βελτίωση σε κάθε γείτονα από τους k δόθηκε ένας συντελεστής βάρους που είναι η αντίστροφη απόσταση του γείτονα από το test δεδομένο, ώστε ο συντελεστής να είναι μεγαλύτερος όσο πιο κοντά βρίσκεται ο γείτονας στο test δεδομένο. Έτσι, εκτός από το πλήθος των γειτόνων συνυπολογίζεται και η απόστασή τους από το δεδομένο.Παρατίθεται και πάλι διάγραμμα με την πορεία του ποσοστού επιτυχίας σε σχέση με το k.



Παρατηρούμε όπως αναμέναμε ότι υπάρχει μια μικρή βελτίωση σε σχέση με τον κλασσικό NNR αλλά και πάλι διαπιστώνεται ότι η αύξηση του k δυσχεραίνει το ποσοστό επιτυχίας όπως και πριν.

Βήμα 15

Ένα πρόβλημα που παρουσιάζεται είναι ότι ο αλγόριθμος κατατάσσει έν α δεδομένο σε μία(1) από δύο(2) διαφορετικές κατηγορίες, είτε σε αυτή που συμβολίζουμε -1 ή σε εκείνη που συμβολίζουμε με 1, πράγμα που σημαίνει ότι κρίνεται απαραίτητο να κατασκευάσουμε δέκα διαφορετικούς SVM ταξινομητές. Ο i-οστός ταξινομητής εξετάζει, δηλαδή, αν το προς εξέταση ψηφίο ανήκει στην κλάση του ψηφίου i ή όχι. Ωστόσο, αυτό δεν είναι αρκετό καθώς είναι δύο διαφορετικοί ταξινομητές να κατατάσσουν το ψηφίο στην κλάση τους. . Στη συνέχεια από τα confidence scores που παράγουν οι ταξινομητές επιλέγουμε το μεγαλύτερο από το οποίο ταξινομούμε τελικά το στοιχείο μας. Η συνάρτηση που χρησιμοποιείται για την εκπαίδευση των ταξινομητών είναι η svmtrain, η οποία παίρνει ως παραμέτρους τα χαρακτηριστικά ενός ψηφίου και ποια είναι η κλάση του. Αφού έχει εκπαιδευτεί με την svmclassify μας επιστρέφεται το score του και στη συνέχεια κατατάσσεται όπως παραπάνω.

Total success rate linear: 0.899352

Total success rate polynomial: 0.900847

Στο σημείο αυτό ζητείται η υλοποίηση ενός επιπλέον ταξινομητή. Ο ταξινομητής που επιλέχθηκε χρησιμοποιεί δέντρα απόφασης για να κατατάξει το ψηφίο. Αρχικά, με βάση τα train δεδομένα, υλοποιεί ένα δέντρο απόφασης το οποίο έχει τόσα φύλλα όσες και οι διαφορετικές κλάσεις, δηλαδή στο συγκεκριμένο πρόβλημα 10. Το παραπάνω δέντρα υλοποιούνται στο Matlab μέσω της συνάρτησης fitctree. Στην συνέχεια, κάθε ένα ψηφίο των test δεδομένων πρέπει να αντιστοιχηθεί σε μία κλάση. Το ψηφίο φορτώνεται στη ρίζα του δέντρου και με κατάλληλες συγκρίσεις ακολουθεί κάποιο μονοπάτι και φτάνει σε κάποιο φύλλο και επομένως αντιστοιχίζεται στην κλάση του φύλλου αυτού. Η ταξινόμηση στο Matlab γίνεται μέσω της συνάρτησης predict.

Total success rate (decision trees): 0.819631

Όλοι οι ταξινομητές παρουσιάζουν αρκετά υψηλά ποσοστά επιτυχίας με τους αγλγόριθμους γειτόνων όμως να κερδίζουν κατά κράτος για μικρά k κυρίως.Στη συνέχεια ακολουθεί ο αλγόριθμος SVM και τα decision trees Με τον ταξινομητή Bayesian να υστερεί έναντι των υπολοίπων εκτός της περίπτωσης που η διαφορά τείνει στη μονάδα κατά την οποία παρατηρούμε σαφή βελτίωση αφού χρησιμοποιούμε κανονική κατανομή και στοιχεία μας συγκεντρώνονται κυρίως κοντά στη μέση τιμή.

Βήμα 16

Από την σύγκριση για την επίδοση των ταξινομητών ανά ψηφίο βλέπουμε ότι κάποιοι ταξινομητές έχουν αρκετά καλή επίδοση για πολλά ψηφία. Επομένως, κρίνεται σκόπιμο να συνδυάσουμε τα αποτελέσματα αυτών για να εξάγουμε καλύτερα αποτελέσματα. Σε πρώτη φάση ,δοκιμάζουμε τον συνδυασμό όλων παραπάνω υλοποιήσεων. Συγκεκριμένα, εισάγουμε κάθε ένα ψηφίο προς εξέταση σε όλους τους ταξινομητές, το οποίο ο καθένας το κατατάσσει σε κάποια κλάση. Το ψηφίο τελικά θα καταταγεί στην κλάση στην οποία το έχουν κατατάξει οι περισσότεροι από τους ταξινομητές.

Total success rate: 0.944694

Στην συνέχεια , επιλέγουμε μόνο τους ταξινομητές που δίνουν την καλύτερη επίδοση ανά ψηφίο .Εκπαιδεύουμε τους ταξινομητές αυτούς μόνο με το 80% των train δεδομένων και θεωρούμε ως test δεδομένα το υπόλοιπο 20%, έχοντας μοιράσει τα δεδομένα, ώστε κάθε νέο σύνολο να περιέχει το αντίστοιχο ποσοστό από κάθε ψηφίο. Στην συνέχεια, αντιστοιχίζουμε την εκτίμηση καθενός από τους ταξινομητές σε κάποιο confidence score, το οποίο είναι μια τιμή κανονικοποιημένη στο [0, 1]. Για κάποιο ψηφίο προς εξέταση, θεωρούμε ότι αντιστοιχεί στην κλάση με το μεγαλύτερο score. Στον NNR – 1 NNR-k (με μέσες τιμές)-και με βάρη κανονικοποίηση στο [0,1]- και SVM κανονικοποιούμε τις αποστάσεις στο [0, 1] αφού πρώτα τις έχουμε αντιστρέψει ώστε η μέγιστη να δίνει την επιθυμητή κλάση, αφού αρχικά η ελάχιστη απόσταση υποδηλώνει την καταλληλότερη κλάση. Ως τελική κλάση για το ψηφίο, επιλέγεται αυτή που δίνει το μεγαλύτερο score, που προκύπτει συνδυάζοντας το επιμέρους με μία από τις συναρτήσεις μέγιστου, ελάχιστου, γινομένου η μέσης τιμής. Η συνάρτηση που έδωσε την καλύτερη επίδοση για το 20% των δεδομένων θα χρησιμοποιηθεί για να ταξινομηθούν τα test δεδομένα. Στο επόμενο διάγραμμα φαίνεται η επίδοση κάθε συνάρτησης για το 20% των δεδομένων.

Συμπεραίνουμε ότι την καλύτερη επίδοση την έχει η συνάρτηση της μέσης τιμής. Χρησιμοποιώντας αυτήν ταξινομούμε τα test δεδομένα και λαμβάνουμε ποσοστό επιτυχίας 94,67%.