ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

Σχολης θετικων Επιστημων

Τμημα Μαθηματικων

Πτυχιακη Εργασια

Αυτόματη Αναγνώριση Χειρόγραφων Μουσικών Κειμένων

Συγγραφέας:
Κοτορος Παναγιωτης

Επιβλέπων Καθηγητής:
Παπασαλουρος Ανδρεας

10 Σεπτεμβρίου 2019

Τριμελής Επιτροπή:
Παπασαλουρος Ανδρεας
Νασιου Παναγιωτης
Κορναρος Χαραλαμπος

ΕΥΧΑΡΙΣΤΙΕΣ

Ευχαριστώ θερμά τον επιβλέποντα μου επίκουρο καθηγητή κ. Παπασαλούρο Ανδρέα για την εμπιστοσύνη και την καθοδήγηση του σε όλη την διάρκεια της προσπάθειας αυτής. Ακόμη, θα ήθελα να ευχαριστήσω μέσα από τα βάθη της καρδιάς μου, τους γονείς μου για την ιδιαίτερη συνεισφορά τους και την υποστήριξη τους. Επίσης, ευχαριστώ πολύ τη συνάδελφο μου Τριακόσια Αικατερίνη για τη βοήθεια της. Τέλος, θα ήθελα να ευχαριστήσω τα μέλη της τριμελής επιτροπής, κ. Νάστου Παναγιώτη επίκουρο καθηγητή του τμήματος και τον επίκουρο καθηγητή κ. Κορνάρο Χαράλαμπο.

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Abstract

Ελληνικά: Η παρούσα εργασία έχει σκοπό αρχικά να εισάγει τον αναγνώστη στις βασικές έννοιες των μεθόδων αναγνώρισης χειρόγραφων συμβόλων. Θα γίνει επίσης αναφορά σε ένα γενικό μοντέλο διαδικασιών που χρησιμοποιεί η οπτική αναγνώριση χαρακτήρων. Έπειτα θα ασχοληθούμε με το κομμάτι της αυτόματης αναγνώρισης μουσικών κειμένων, τόσο ψηφιακών όσο και χειρόγραφων, και τους τρόπους με τους οποίους επιτυγχάνεται. Θα μιλήσουμε για τους αλγορίθμους Support Vector Machines, με και χωρίς τη μέθοδο του πυρήνα, και πλησιέστερου γείτονα και με ποιόν τρόπο επιτυγχάνουν να αναγνωρίσουν τα δεδομένα που επεξεργάζονται. Τέλος με τη χρήση του πακέτου Muscima++ θα δούμε στην πράξη τα αποτελέσματα κάθε μεθόδου και θα τα συγκρίνουμε μεταξύ τους.

English: The purpose of this thesis is to introduce the main methods of Optical Character Recognition to the reader. We will discuss a general model under witch OCR is achieved. Then, we are going to talk about Optical Music Recognition, both handwritten and digital symbols, and the methods we use in this category. Support Vector Machines and Nearest Neighbors algorithms will be discussed and a mathematical proof of SVM's and SVM's with the kernel method algorithms will be included. We will test these three methods using MUSCIMA++ software and we will compare them.

Κεφάλαιο 1

Εισαγωγή

Η Επιστήμη των Υπολογιστών έχει ραγδαία εξέλιξη στις μέρες μας. Από τα πρώιμα κιόλας στάδια της εξέλιξης των υπολογιστών υπήρχε η ιδέα ενός υπολογιστικού συστήματος το οποίο θα μπορούσε να σκεφτεί, να μάθει, να συζητά και εν ολίγοις να προσομοιώνει την ανθρώπινη συμπεριφορά. Ο τομέας αυτός σήμερα βρίσκεται σε ένα αρκετά ικανοποιητικό στάδιο αλλά δε σταματά να εξελίσσεται όσο περνάει ο καιρός και το όνομά του είναι Τεχνητή Νοημοσύνη. •Ένα από τα παρακλάδια της τεχνητής νοημοσύνης το οποίο θα πραγματευθούν τα επόμενα κεφάλαια είναι η μηχανική μάθηση. Πιο συγκεκριμένα θα ασχοληθούμε με τρόπους με τους οποίους ένα υπολογιστικό σύστημα μπορεί να αναλύσει χειρόγραφα σύμβολα με σκοπό την αναγνώριση τους. Ξεκινώντας από την αρχή συνοπτικά, λέμε ότι μία μηχανή μπορεί να μάθει αλλά τι είναι μάθηση;

1.1 Μάθηση

Η μάθηση αποτελεί μία από τις θεμελιώδεις ιδιότητες της νοήμονος συμπεριφοράς του ανθρώπου.

Ορισμός 1 *Μάθηση είναι η διαδικασία όπου ο μαθητής αποκτά γνώσεις, δεξιότητες, συμπεριφορές και αξίες μέσα από την παράθεση εκπαιδευτικού υλικού και με την εφαρμογή γνωστικών διαδικασιών.*

Με βάση τον ορισμό αυτό πόσο πιθανό θα ήταν να δημιουργηθούν υπολογιστικά συστήματα ικανά να μάθουν, να επιτύχουν δηλαδή τη λεγόμενη μηχανική μάθηση;

1.1.1 Μηχανική Μάθηση

Η μηχανική μάθηση αποτελεί υποεπίπεδο της επιστήμης των υπολογιστών που αναπτύχθηκε από την μελέτη της αναγνώρισης προτύπων και της υπολογιστικής θεωρίας

μάθησης στην τεχνητή νοημοσύνη. Η μηχανική μάθηση διερευνά τη μελέτη και την κατασκευή των αλγορίθμων που μπορούν να μαθαίνουν από τα δεδομένα και να κάνουν προβλέψεις σχετικά με αυτά. Κάποια παραδείγματα εφαρμογών της μηχανικής μάθησης αποτελούν τα φίλτρα σπαμ (spam filters), οι μηχανές αναζήτησης, η οπτική αναγνώριση χαρακτήρων (Optical Character Recognition) αλλά και εικόνων ή συμβόλων. Ένας πιο επίσημος ορισμός που προτάθηκε από τον Tom Michael Mitchell είναι ο παρακάτω

Ορισμός 2 *Ένα πρόγραμμα υπολογιστή λέγεται ότι μαθαίνει από μία εμπειρία E ως προς μία κλάση εργασιών T και ένα μέτρο επίδοσης R, αν η επίδοση του σε εργασίες της κλάσης T, όπως αποτιμάται από το μέτρο R, βελτιώνεται με την εμπειρία E.*

Υπάρχουν διάφοροι τρόποι με τους οποίους μπορούμε να επιτύχουμε τη μηχανική μάθηση ένα παράδειγμα, με το οποίο θα ασχοληθούμε παρακάτω, είναι το deep learning.

1.2 Αναγνώριση Προτύπων

Η αναγνώριση προτύπων (Pattern Recognition) αποτελεί πεδίο της επιστήμης τον υπολογιστών που έχει σαν στόχο την ανάπτυξη αλγορίθμων για την αυτοματοποιημένη απόδοση κάποιας τιμής ή διακριτού στοιχείου σε εισαγόμενα δεδομένα. Συνήθως τα δεδομένα αυτά είναι κωδικοποιημένα σαν αλληλουχίες αριθμών. Με αυτό τον τρόπο τα δεδομένα ταξινομούνται αυτόματα σε κατηγορίες ή διαχωρίζονται σε ομάδες με βάση κάποια κριτήρια. Το ερευνητικό ενδιαφέρον για την αναγνώριση προτύπων έχει τις ρίζες του στη δεκαετία του 1960, κατά την πρώτη περίοδο ανάπτυξης της πληροφορικής και ειδικότερα τις τεχνητής νοημοσύνης.

Οι άνθρωποι και οι ευφυείς οργανισμοί από τα πρώτα κιόλας χρόνια της ζωής τους, εξασκούν την ικανότητα να ταυτοποιούν πραγματικά δεδομένα χρησιμοποιώντας τις αισθήσεις τους και την αντιληπτική τους ικανότητα. Έτσι λοιπόν λαμβάνουν τις κατάλληλες αποφάσεις ώστε να επιβιώσουν στο περιβάλλον τους. Μία μηχανή, όπως ένας ηλεκτρονικός υπολογιστής, πρέπει να εκπαιδευθεί κατάλληλα ώστε να αναγνωρίζει πρότυπα (patterns) και να τα κατηγοριοποιεί αυτόματα σε κατηγορίες. Ανάλογα με την εφαρμογή, συνήθως γίνεται κατάταξη των αντικειμένων σε κλάσεις με τη βοήθεια αλγορίθμων ταξινόμησης.

Με βάση το θεωρητικό υπόβαθρο της στατιστικής τη δεδομένη εποχή, η πρώιμη έρευνα της αναγνώρισης προτύπων επικεντρώθηκε στην ανάπτυξη θεωρητικών μεθόδων. Μετά το 1970 πραγματοποιήθηκαν προσπάθειες για την για την ταχύτερη εξέλιξη του τομέα, ενώ το 1976 ιδρύθηκε η διεθνής Ένωση αναγνώρισης Προτύπων (IARP). Στις μέρες μας οι αλγόριθμοι αναγνώρισης προτύπων έχουν βρει εφαρμογή σε αρκετές επιστήμες όπως για παράδειγμα στην ιατρική (βιοϊατρική τεχνολογία, ανάλυση δεδομένων DNA και άλλες εφαρμογές της βιοπληροφορικής), ή σε άλλα πεδία

της πληροφορικής και της επιστήμης ηλεκτρονικού μηχανικού, όπως η μηχανική όραση και η ρομποτική. Η αναγνώριση προτύπων επικαλύπτεται σημαντικά με συγγενή επιμέρους πεδία της τεχνητής νοημοσύνης όπως η μηχανική μάθηση και η εξόρυξη δεδομένων.

Οι μέθοδοι αναγνώρισης προτύπων ποικίλουν. Κάθε τρόπος διαφέρει από τους υπόλοιπους και σύμφωνα με τον τρόπο λειτουργίας κάθε μεθόδου, μπορούμε σχεδόν πάντα να βρούμε την καταλληλότερη για το πρόβλημα μας μέθοδο. Κάποιες μέθοδοι είναι: κοντινότερος γείτονας (Nearest Neighbor), γραμμικοί και μη-γραμμικοί κατηγοριοποιητές, Support Vector Machines, κατηγοριοποιητές Bayes.

Σημαντικό κομμάτι κατά τη διαδικασία αναγνώρισης προτύπων επίσης είναι εκπαίδευση και η μάθηση. Η μάθηση είναι ένα φαινόμενο μέσω του οποίου ένα σύστημα έχει εκπαιδευτεί και είναι να ικανό να δώσει αποτελέσματα με συγκεκριμένο τρόπο. Είναι η βασικότερη διαδικασία που πραγματοποιεί ένας αλγόριθμος αυτής της κατηγορίας μιας και θα κρίνει την αποτελεσματικότητα του σύμφωνα με τα αποτελέσματα. Το δείγμα συνήθως χωρίζεται σε δύο κατηγορίες, το δείγμα εκπαίδευσης και το δείγμα εξέτασης. Οι διαφορές μεταξύ τους είναι:

- **Δείγμα Εκπαίδευσης:** Τα στοιχεία του δείγματος σε αυτή τη φάση χρησιμοποιούνται προκειμένου να δημιουργηθεί ένα μοντέλο, το οποίο αποτελείται από τις εικόνες που θα χρησιμοποιηθούν για την εκπαίδευση. Οι κανόνες και οι αλγόριθμοι που χρησιμοποιούνται παρέχουν σχετικές πληροφορίες σχετικά με τον τρόπο σύνδεσης των δεδομένων εισόδου με την απόφαση εξόδου. Το σύστημα εκπαιδεύεται εφαρμόζοντας αυτούς τους αλγορίθμους στο σύνολο δεδομένων. Όλες οι σχετικές πληροφορίες εξάγονται από τα δεδομένα και τα αποτελέσματα καταγράφονται.

- **Δείγμα Εξέτασης:** Τα δεδομένα εξέτασης χρησιμοποιούνται για τη δοκιμή της αποδοτικότητας και της ακρίβειας του συστήματος. Πρόκειται για το σύνολο των δεδομένων που χρησιμοποιείται για να επαληθεύσει εάν το σύστημα παράγει τη σωστή έξοδο μετά την εκπαίδευση του. Γενικά, το 20% των δεδομένων του δείγματος χρησιμοποιείται για τις δοκιμές. Για παράδειγμα αν ένα σύστημα προσδιορίζει σε ποια κατηγορία ανήκει ένα συγκεκριμένο αντικείμενο και είναι ικανό να αναγνωρίσει σωστά τις επτά κατηγορίες από τις δέκα έχει ποσοστό επιτυχίας 70%

Ένα πρότυπο είναι ένα φυσικό αντικείμενο ή μια αφηρημένη έννοια. Ενώ μιλάμε για τις διάφορες κατηγορίες ζώων, μία περιγραφή ενός ζώου αποτελεί ένα πρότυπο. Αν μιλούσαμε για διάφορους τύπους μπαλών, η περιγραφή μια μπάλας είναι ένα μοτίβο. Στην περίπτωση που οι μπάλες θεωρούνται μοτίβα, οι τάξεις θα μπορούσαν να είναι για παράδειγμα μπάλα ποδοσφαίρου, μπάλα μπάσκετ, μπάλα κρίκετ. Δεδομένου ενός νέου μοτίβου πρέπει να προσδιοριστεί η κλάση του. Η επιλογή χαρακτηριστικών

και η αναπαράσταση των μοτίβων είναι αρκετά σημαντικό βήμα για την ταξινόμηση προτύπων.

Μία προφανής αναπαράσταση ενός μοτίβου θα μπορούσε να είναι ένα διάνυσμα. Κάθε συντεταγμένη του διανύσματος θα αντιπροσωπεύει και ένα χαρακτηριστικό του μοτίβου. Για παράδειγμα, στα σφαιρικά αντικείμενα το διάνυσμα $(25, 15)$ θα μπορούσε να περιγράφει στην πρώτη θέση το βάρος της μπάλας, ενώ στη δεύτερη θέση τη διάμετρο της σε εκατοστά. Σε περίπτωση που η σφαίρα αυτή ανήκει στην κατηγορία ένα στο πρόβλημα μας θα μπορούσε να γραφεί $(25, 1, 1)$ όπου η τρίτη θέση αναπαριστά την κλάση της κάθε σφαίρας.

Κάποια πλεονεκτήματα και μειονεκτήματά της αναγνώρισης προτύπων σαν τεχνολογία είναι τα εξής Θετικά:

- Η αναγνώριση προτύπων επιλύει προβλήματα κατηγοριοποίησης.
- Η αναγνώριση μοτίβων επιλύει το πρόβλημα της λανθασμένης βιομετρικής ανίχνευσης.
- Βοηθά στη διάρθρωση των ομιλητών.
- Μπορεί να βοηθήσει στην ανάπτυξη λογισμικών για άτομα με ειδικές ανάγκες (για παράδειγμα στην αναγνώριση μοτίβων ρούχων για ανθρώπους με προβλήματα όρασης).

Αρνητικά:

- Η προσέγγιση της αναγνώρισης του συντακτικού μοντέλου είναι αρκετά περίπλοκη και χρονοβόρα διαδικασία.
- Κάποιες φορές, προκειμένου να έχουμε καλύτερα αποτελέσματα χρειαζόμαστε μεγαλύτερο δείγμα.
- Δεν υπάρχει συγκεκριμένη εξήγηση στο γιατί αναγνωρίζεται ένα συγκεκριμένο αντικείμενο.

Ο τομέας της αναγνώρισης προτύπων χωρίζεται επιπροσθέτως σε άλλες δύο κατηγορίες, την online και την offline αναγνώριση. Στην πρώτη κατηγορία η αναγνώριση γίνεται άμεσα κατά τη διαδικασία ανάγνωσης του εκάστοτε μοτίβου και τα αποτελέσματα είναι διαθέσιμα την ίδια στιγμή. Στη δεύτερη κατηγορία η αναγνώρισης γίνεται συνολικά σε όλα τα σαρωμένα στοιχεία του δείγματος όχι απαραίτητα την ίδια χρονική στιγμή με την καταγραφή τους. Εμείς στα επόμενα κεφάλαια θα ασχοληθούμε με τη δεύτερη κατηγορία, δηλαδή την offline αναγνώριση.

1.3 Σκοπός

Σκοπός της εργασίας είναι να εισάγει τον αναγνώστη στις βασικές έννοιες της οπτικής αναγνώρισης χαρακτήρων. Ξεκινώντας από τους τομείς της μηχανικής μάθησης και της αναγνώρισης προτύπων και συνεχίζοντας πιο ειδικά, για να καταλήξουμε στην αναγνώριση μουσικών συμβόλων. Θα δούμε τις βασικές έννοιες της οπτικής αναγνώρισης χειρόγραφων μουσικών κειμένων. Επίσης, θα αναφερθούμε σε κάποιους αλγόριθμους κατηγοριοποίησης και στον τρόπο λειτουργίας τους. Τέλος, θα συγκρίνουμε τα αποτελέσματα των αλγορίθμων αυτών στην πράξη.

Κεφάλαιο 2

Οπτική Αναγνώριση Χαρακτήρων (Optical Character Recognition)

Στο δεύτερο κεφάλαιο θα αναφέρουμε κάποια από τα βασικά στοιχεία του OCR. Θα μιλήσουμε για την εξέλιξη του από τα πρώτα του στάδιο μέχρι τις πιο εξελιγμένες μορφές του καθώς και τα κύρια προβλήματα τα οποία πραγματεύεται. Τέλος θα αναφέρουμε θεωρητικά σε κάποιες μεθόδους υλοποίησης που επιλύουν κάποια από τα προαναφερθέντα προβλήματα.

2.1 Εισαγωγή στο OCR

Ορισμός 3 *Η οπτική αναγνώριση χαρακτήρων (ή αυτόματη αναγνώριση χαρακτήρων) κειμένου ονομάζεται η διαδικασία μετατροπής σαρωμένων εικόνων χειρόγραφων ή έντυπων κειμένων σε ψηφιακό κείμενο.*

Ανήκει στην κατηγορία της υπολογιστικής όρασης. Η πρώτη φορά που εμφανίστηκε ήταν πολύ νωρίς συγκριτικά με την εξέλιξη του τομέα. Αυτό συνέβη διότι στα πρώιμα στάδια το OCR δε χρησιμοποιήθηκε σε συνδυασμό με οποιαδήποτε άλλη μέθοδο. Σίγουρα από μόνο του το OCR αποτελεί μία πολύ ικανοποιητική λύση για κάποια προβλήματα. Έχει παρατηρηθεί όμως ότι ο συνδυασμός του με τη βαθιά μάθηση για παράδειγμα, μπορεί να δώσει πολύ καλύτερα αποτελέσματα και σίγουρα επεκτείνει αρκετά το πεδίο εφαρμογής του OCR.

Η αρχή της αυτόματης αναγνώρισης χειρόγραφων κειμένων μπορεί να εντοπιστεί το 1914. Προς τα τέλη του 1920 ο Emanuel Goldberg έφτιαξε ένα, όπως το αποκάλεσε, "στατιστικό μηχάνημα" το οποίο έψαχνε έγγραφα που αποτελούνται από μικροφίλμ με χρήση συστήματος OCR. Η εφεύρεση του αποκτήθηκε από την IBM το 1931. Στις μέρες μας πολλές εφαρμογές που εξάγουν ψηφιακά κείμενα από φωτογραφίες κάνουν χρήση της παραπάνω τεχνολογίας. Επίσης, πολλά πρωτόκολλα

ασφαλείας από μεγάλες εταιρείες όπως η google, στα οποία ζητείται από το χρήστη να δακτυλογραφήσει τι απεικονίζουν συγκεκριμένες εικόνες ή να διαλέξει τα κομμάτια κάποιας φωτογραφίας που απεικονίζονται συγκεκριμένα αντικείμενα, χρησιμοποιούν την εν λόγω τεχνολογία.

2.2 Αναγνώριση Κειμένου

Στην καθημερινότητα μας πολλές φορές χρειάζεται να διαβάσουμε κάποιο κείμενο γραμμένο σε χαρτί από κάποιον άλλο άνθρωπο. Η διαδικασία αυτή για το μέσω άνθρωπο είναι μία αυτοματοποιημένη λειτουργία την οποία έχει τελειοποιήσει με το πέρας της βασικής του σχολικής εκπαίδευσης. Πώς όμως αντιμετωπίζει το ίδιο κείμενο μία μηχανή;

Αν σαρώσουμε ένα χειρόγραφο κείμενο, που μόλις διαβάσαμε και αναγνωρίσαμε, σε μία μηχανή το αποτέλεσμα προφανώς δε θα είναι το ίδιο. Μία μηχανή αντιλαμβάνεται ένα σαρωμένο αρχείο κειμένου σαν μία σειρά από πίξελ καθένα από τα οποία έχει διαφορετική αξία ανάλογα με το χρωματισμό του. Παρόλα αυτά στις μέρες υπάρχουν λογισμικά που μέσα από αυτή τη σειρά με πίξελ αναγνωρίζουν τις εικονιζόμενες λέξεις και μάλιστα με αρκετά μεγάλη ακρίβεια. Ένα πολύ διαδεδομένο λογισμικό για τον τομέα του OCR είναι το Tesseract. Πρόκειται για ένα δωρεάν λογισμικό που δημοσιεύτηκε από την Apache [1]. Ο τρόπος λειτουργίας της αναγνώρισης αυτής είναι ο εξής.

Έστω ότι το αλφάβητό μας είχε μόνο ένα γράμμα, το Α. Ακόμη και τότε η δουλειά μιας μηχανής δε θα ήταν και τόσο απλή δεδομένου ότι κάθε άνθρωπος έχει το δικό του γραφικό χαρακτήρα. Μπορεί τα αποτελέσματα να είναι αρκετά όμοια το ένα στο άλλο, αλλά δύσκολα θα είναι εντελώς ίδια. Ακόμη και αν μελετήσουμε δείγμα τυπωμένο από υπολογιστή, δε μπορούμε να υποσχεθούμε ευκολότερη διαδικασία μιας και υπάρχουν πολλές διαφορετικές μορφοποιήσεις κειμένου που αλλάζουν αρκετά την αποτύπωση των γραμμάτων στο χαρτί. Στην πράξη υπάρχουν δύο πιθανοί τρόποι επίλυσης του προβλήματος αυτού. Είτε αναγνωρίζοντας τους χαρακτήρες μέσα από το σύνολο στο οποίο υπάρχουν (αναγνώριση προτύπου), είτε ανιχνεύοντας τις επιμέρους γραμμές ή σύμβολα τα οποία απαρτίζουν τον συγκεκριμένο χαρακτήρα (ανίχνευση χαρακτηριστικών).

2.2.1 Περιεχόμενα Συστήματος OCR

Ένα τυπικό σύστημα OCR συνήθως αποτελείται από διάφορα στοιχεία. Το πρώτο βήμα στη διαδικασία είναι η ψηφιοποίηση του αναλογικού εγγράφου με τη χρήση ενός οπτικού σαρωτή. Όταν εντοπίζονται περιοχές που περιέχουν κάποιο κείμενο, κάθε σύμβολο εξάγεται μέσω μίας κατηγορίας τμηματοποίησης. Τα εξαχθέντα σύμβολα

[1] Λογισμικό ανοιχτού κώδικα με άδεια Apache https://github.com/tesseract-ocr/

μπορούν στη συνέχεια να προ επεξεργαστούν, αφαιρώντας το θόρυβο, για να διευκολυνθεί η εξαγωγή χαρακτηριστικών στο επόμενο βήμα. Η ταυτότητα κάθε συμβόλου βρίσκεται με τη σύγκριση των εξαγομένων χαρακτηριστικών με τις περιγραφές των τάξεων συμβόλων που αποκτήθηκαν από μια προηγούμενη φάση μάθησης. Τέλος, οι πληροφορίες χρησιμοποιούνται για την ανασύσταση των λέξεων και αριθμών του αρχικού κειμένου. Στη συνέχεια κάποια από αυτά τα βήματα και τις μεθόδους περιγράφονται με περισσότερες λεπτομέρειες

2.2.2 Οπτική Σάρωση

Μέσω της διαδικασίας σάρωσης καταγράφεται μια ψηφιακή εικόνα του πρωτότυπου εγγράφου. Στο OCR χρησιμοποιούνται οπτικοί σαρωτές, οι οποίοι αποτελούνται από ένα μηχανισμό μεταφοράς και μία συσκευή ανίχνευσης που μετατρέπει την ένταση του φωτός σε επίπεδα γκρι. Τα τυπωμένα έγγραφα αποτελούνται συνήθως από μαύρη εκτύπωση σε λευκό φόντο. Επομένως, όταν εκτελείται OCR, είναι μία συνηθισμένη διαδικασία η μετατροπή της πολυεπίπεδης εικόνας σε ασπρόμαυρη. Συχνά αυτή η διαδικασία, γνωστή ως Thresholding, εκτελείται στον σαρωτή για εξοικονόμηση χώρου μνήμης και υπολογιστικής προσπάθειας. Η διαδικασία του thresholding είναι σημαντική καθώς τα αποτελέσματα της ακόλουθης αναγνώρισης εξαρτώνται άμεσα από την ποιότητα της ασπρόμαυρης εικόνας. Συνήθως, χρησιμοποιείται ένα threshold σαν όριο. Κάτω από αυτό τα επίπεδα γκρίζου λέγεται ότι είναι μαύρα, ενώ πάνω από αυτό τα επίπεδα λέγεται ότι είναι λευκά. Για ένα έγγραφο υψηλής αντίθεσης με ομοιόμορφο φόντο, μπορεί να επαρκεί ένα προκαθορισμένο σταθερό threshold. Ωστόσο, πολλά έγγραφα που συναντώνται στην πράξη έχουν μεγάλο εύρος αντίθεσης. Σε αυτές τις περιπτώσεις απαιτούνται πιο περίπλοκες μέθοδοι thresholding για την επίτευξη καλού αποτελέσματος.

Οι καλύτερες μέθοδοι thresholding είναι συνήθως αυτές που είναι σε θέση να μεταβάλλουν το threshold πάνω από το έγγραφο και να προσαρμοστεί στις τοπικές ιδιότητες ως αντίθεση και φωτεινότητα. Ωστόσο, αυτές οι μέθοδοι συνήθως εξαρτώνται από μια πολυεπίπεδη σάρωση του εγγράφου που απαιτεί περισσότερη μνήμη και υπολογιστική ικανότητα. Συνεπώς, τέτοιες τεχνικές σπάνια χρησιμοποιούνται σε σύνδεση με συστήματα OCR, αν και έχουν ως αποτέλεσμα καλύτερες εικόνες.

2.2.3 Ανίχνευση Και Τμηματοποίηση

Η τμηματοποίηση είναι μία διαδικασία που καθορίζει τα συστατικά μίας εικόνας.
Είναι απαραίτητο να εντοπίσουμε τις περιοχές του εγγράφου όπου έχουν τυπωθεί τα δεδομένα και να τα διακρίνουμε από τα στοιχεία και τα γραφικά. Για παράδειγμα, κατά την αυτόματη ταξινόμηση αλληλογραφίας, η διεύθυνση πρέπει να εντοπιστεί και να διαχωριστεί από άλλες εκτυπώσεις στον φάκελο, όπως γραμματόσημα και εταιρικά λογότυπα, πριν από την αναγνώριση.

Εφαρμοσμένη στο κείμενο, η τμηματοποίηση είναι η απομόνωση χαρακτήρων ή λέξεων. Η πλειονότητα των αλγορίθμων OCR χωρίζει τις λέξεις σε απομονωμένους χαρακτήρες οι οποίοι αναγνωρίζονται ξεχωριστά. Συνήθως αυτή η τμηματοποίηση πραγματοποιείται με την απομόνωση κάθε συνδεδεμένου στοιχείου, δηλαδή κάθε συνδεδεμένης μαύρης περιοχής. Αυτή η τεχνική είναι εύκολη στην υλοποίηση, αλλά προκύπτουν προβλήματα αν οι χαρακτήρες εφάπτονται ή οι χαρακτήρες είναι κατακερματισμένοι και αποτελούνται από πολλά μέρη. Τα κύρια προβλήματα στην κατάτμηση μπορούν να χωριστούν σε τέσσερις ομάδες:

- **Εξαγωγή κατακερματισμένων χαρακτήρων και χαρακτήρων που εφάπτονται:** Τέτοιες στρεβλώσεις μπορούν να οδηγήσουν σε πολλούς κοντινούς χαρακτήρες που ερμηνεύονται ως ένας μόνο χαρακτήρας, ή σε ένα κομμάτι χαρακτήρα να θεωρηθεί ένα ολόκληρο σύμβολο. Κολλημένοι χαρακτήρες μπορούν να παρουσιαστούν εάν το έγγραφο προέρθει από σκοτεινό φωτοαντίγραφο ή εάν σαρώνεται σε χαμηλό threshold. Ένας ακόμη λόγος θα μπορούσε να είναι πολύ στριμωγμένη γραμματοσειρά. Οι χαρακτήρες μπορεί να χωριστούν εάν το έγγραφο προέρχεται από φωτεινό φωτοαντίγραφο ή σαρώνεται σε υψηλό threshold.
- **Απαλοιφή του θορύβου από το κείμενο:** Κάποιες φορές οι κουκκίδες και οι τόνοι μπορεί να αντιμετωπισθούν σαν θόρυβος και αντίστροφα.
- **Εσφαλμένα γραφικά ή γεωμετρία για το κείμενο:** Αυτό μπορεί να οδηγήσει σε κομμάτια του αρχείου που δεν είναι κείμενο να συνεχίσουν στο στάδιο της αναγνώρισης.
- **Μπερδεύοντας κείμενο για γραφικά ή γεωμετρία:** Στην περίπτωση αυτή, το κείμενο δεν θα περάσει στο στάδιο της αναγνώρισης. Αυτό συμβαίνει συχνά όταν οι χαρακτήρες συνδέονται με γραφικά.

2.2.4 Προ επεξεργασία

Η εικόνα που προκύπτει από τη διαδικασία σάρωσης μπορεί να περιέχει ένα ορισμένο ποσό θορύβου. Ανάλογα με την ανάλυση του σαρωτή και την επιτυχία της εφαρμοζόμενης τεχνικής thresholding, οι χαρακτήρες μπορεί να είναι απομακρυσμένοι ή κατακερματισμένοι. Ορισμένα από αυτά τα ελαττώματα, τα οποία ενδέχεται αργότερα να επιφέρουν χαμηλά ποσοστά αναγνώρισης, μπορούν να εξαλειφθούν με τη χρήση ενός προ επεξεργαστή για την εξομάλυνση των ψηφιοποιημένων χαρακτήρων.

Η εξομάλυνση που αναφέραμε μπορεί να διορθώσει κάποια προβλήματα. Για παράδειγμα η απομάκρυνση μπορεί να εξαλειφθεί μειώνοντας το πλάτος της γραμμής. Οι πιο συνηθισμένες τεχνικές εξομάλυνσης, μετατρέπουν μία κανονική εικόνα στην

δυαδική εικόνα του χαρακτήρα και εφαρμόζουν ορισμένους κανόνες στα περιεχόμενα της.

Εκτός από την εξομάλυνση, η προ επεξεργασία συνήθως περιλαμβάνει την κανονικοποίηση. Η κανονικοποίηση εφαρμόζεται για τη λήψη χαρακτήρων οποιουδήποτε μεγέθους, κλίσης και περιστροφής. Για να μπορέσει να διορθωθεί η περιστροφή, πρέπει να βρεθεί η γωνία περιστροφής. Για τις περιστρεφόμενες σελίδες και τις γραμμές του κειμένου, χρησιμοποιούνται συνήθως παραλλαγές μετασχηματισμών για την ανίχνευση της κλίσης. Ωστόσο, η εύρεση της γωνίας περιστροφής ενός συμβόλου δεν είναι δυνατή μέχρι να αναγνωριστεί το σύμβολο.

2.2.5 Εξαγωγή Χαρακτηριστικών

Ο στόχος της εξαγωγής χαρακτηριστικών είναι να συλλάβει τα βασικά χαρακτηριστικά των συμβόλων και είναι γενικά αποδεκτό ότι αυτό είναι ένα από τα πιο δύσκολα προβλήματα στην αναγνώριση προτύπων. Ο πιο απλός τρόπος περιγραφής ενός χαρακτήρα είναι η πραγματική εικόνα του raster[2]. Μια άλλη προσέγγιση είναι να εξάγουμε ορισμένα χαρακτηριστικά που εξακολουθούν να χαρακτηρίζουν τα σύμβολα, αλλά αφήνει εκτός τα ασήμαντα χαρακτηριστικά. Οι τεχνικές για την εξαγωγή τέτοιων χαρακτηριστικών χωρίζονται συχνά σε τρεις κύριες ομάδες, όπου τα χαρακτηριστικά εντοπίζονται από:

- Η κατανομή των σημείων
- Μετασχηματισμοί και επεκτάσεις σειρών
- Ανάλυση της δομής

Οι διαφορετικές ομάδες χαρακτηριστικών μπορούν να αξιολογηθούν ανάλογα με την ευαισθησία τους στο θόρυβο και τις παραμορφώσεις, και την ευκολία εφαρμογής και χρήσης.

2.2.6 Τεχνικές Βασισμένες Σε Χαρακτηριστικά

Σε αυτές τις μεθόδους υπολογίζονται σημαντικές μετρήσεις. Έπειτα εξάγονται από έναν χαρακτήρα και συγκρίνονται με τις περιγραφές των κατηγοριών χαρακτήρων που αποκτούνται κατά τη διαδικασία της εκπαίδευσης. Το σύμβολο του οποίου η περιγραφή που ταιριάζει περισσότερο, είναι και αυτό που θα ταυτιστεί με το εξεταζόμενο στο κομμάτι της αναγνώρισης. Τα χαρακτηριστικά δίνονται ως αριθμοί σε ένα διάνυσμα χαρακτηριστικών και αυτό το διάνυσμα χαρακτηριστικών χρησιμοποιείται για να αντιπροσωπεύει το σύμβολο.

[2]Πραγματική εικόνα raster είναι ένα bitmap

Κατανομή Σημείων: Αυτή η κατηγορία καλύπτει τεχνικές που εξάγουν χαρακτηριστικά με βάση τη στατιστική κατανομή των σημείων. Αυτά τα χαρακτηριστικά είναι συνήθως ανεκτικά σε στρεβλώσεις και παραλλαγές στην εμφάνιση. Ορισμένες από τις τυπικές τεχνικές σε αυτήν την περιοχή είναι:

- **Χωρισμός σε ζώνες:** Το ορθογώνιο που περιγράφει τον χαρακτήρα χωρίζεται σε αρκετές επικαλυπτόμενες ή μη επικαλυπτόμενες περιοχές και οι πυκνότητες των μαύρων σημείων εντός αυτών των περιοχών υπολογίζονται και χρησιμοποιούνται ως χαρακτηριστικά.

- **Περιοχές:** Οι περιοχές των μαύρων σημείων για ένα επιλεγμένο κέντρο, για παράδειγμα το κέντρο βάρους, ή ένα επιλεγμένο σύστημα συντεταγμένων, χρησιμοποιούνται ως χαρακτηριστικά.

- **Διασταυρώσεις και αποστάσεις:** Στην τεχνική διασταύρωσης υπάρχουν χαρακτηριστικά από το πόσες φορές το σχήμα του χαρακτήρα διασχίζεται από φορείς σε ορισμένες κατευθύνσεις. Αυτή η τεχνική χρησιμοποιείται συχνά από εμπορικά συστήματα επειδή μπορεί να εκτελείται με μεγάλη ταχύτητα και απαιτεί χαμηλή πολυπλοκότητα.
  Όταν χρησιμοποιείται η τεχνική απόστασης, μετρώνται ορισμένα μήκη κατά μήκος των διανυσμάτων που διασχίζουν το σχήμα του χαρακτήρα. Για παράδειγμα, το μήκος των διανυσμάτων εντός του ορίου του χαρακτήρα.

- **Ν-Λίστες:** Η σχετικά συχνή εμφάνιση ασπρόμαυρων σημείων (προσκηνίου και φόντου) σε συγκεκριμένες εντολές, χρησιμοποιείται ως χαρακτηριστικά.

- **Χαρακτηριστικοί τόποι:** Για κάθε σημείο στο φόντο του χαρακτήρα δημιουργούνται κάθετα και οριζόντια διανύσματα. Ο αριθμός των φορών που τα τμήματα γραμμής που περιγράφουν τον χαρακτήρα διασταυρώνονται από αυτά τα διανύσματα χρησιμοποιούνται ως χαρακτηριστικά.

2.3 Ταξινόμηση

Η ταξινόμηση είναι η διαδικασία αναγνώρισης κάθε χαρακτήρα και της ανάθεσης του στη σωστή κατηγορία χαρακτήρων. Παρακάτω θα αναφερθούμε σε δύο διαφορετικές προσεγγίσεις για την ταξινόμηση στην αναγνώριση χαρακτήρων. Αρχικά θα μιλήσουμε για μία θεωρητική αναγνώριση (decision-theoretic). Αυτές οι μέθοδοι χρησιμοποιούνται όταν η περιγραφή του χαρακτήρα μπορεί να αναπαρασταθεί αριθμητικά σε ένα διάνυσμα χαρακτηριστικών.

Μπορεί επίσης να έχουμε χαρακτηριστικά προτύπου που προέρχονται από τη φυσική δομή του χαρακτήρα, τα οποία δεν είναι τόσο εύκολο να αναπαρασταθούν από κάποια ποσότητα. Σε αυτές τις περιπτώσεις, η σχέση μεταξύ των χαρακτηριστικών

μπορεί να είναι σημαντική όταν αποφασίζουμε για την ένταξη στην σωστή τάξη. Για παράδειγμα, αν γνωρίζουμε ότι ένας χαρακτήρας αποτελείται από μία κάθετη και μία οριζόντια γραμμή, μπορεί να είναι είτε 'L' είτε 'T' και η σχέση μεταξύ των δύο γραμμών είναι απαραίτητη για τη διάκριση των χαρακτήρων. Απαιτείται λοιπόν μια δομική (structural) προσέγγιση.

2.3.1 Θεωρητική Αναγνώριση

Οι κυριότερες προσεγγίσεις στη θεωρητική αναγνώριση αποφάσεων είναι οι ελάχιστες ταξινομητές απόστασης, οι στατιστικοί ταξινομητές και τα νευρωνικά δίκτυα. Κάθε μία από αυτές τις τεχνικές ταξινόμησης περιγράφεται συνοπτικά παρακάτω:
Αντιστοίχηση: Η αντιστοίχηση καλύπτει τις ομάδες τεχνικών που βασίζονται σε μέτρα ομοιότητας. Εκεί υπολογίζεται η απόσταση μεταξύ του διανύσματος χαρακτηριστικών, που περιγράφει τον εξαγόμενο χαρακτήρα και την περιγραφή κάθε κατηγορίας. Μπορούν να χρησιμοποιηθούν διαφορετικά μέτρα, αλλά η πιο συνηθισμένη είναι η ευκλείδεια απόσταση. Αυτός ο ταξινομητής ελάχιστης απόστασης λειτουργεί καλά όταν οι κλάσεις είναι καλά διαχωρισμένες, δηλαδή όταν η απόσταση μεταξύ των μέσων είναι μεγάλη σε σύγκριση με την εξάπλωση κάθε κατηγορίας.

Όταν χρησιμοποιείται ολόκληρος ο χαρακτήρας ως είσοδος στην ταξινόμηση και δεν εξάγονται χαρακτηριστικά (προσαρμογή προτύπου), χρησιμοποιείται προσέγγιση συσχέτισης. Εδώ υπολογίζεται η απόσταση μεταξύ της εικόνας χαρακτήρων και των πρωτότυπων εικόνων που αντιπροσωπεύουν κάθε τάξη χαρακτήρων.
Βέλτιστοι Στατιστικοί Ταξινομητές: Στη στατιστική ταξινόμηση εφαρμόζεται μια πιθανοτική προσέγγιση στην αναγνώριση. Η ιδέα είναι να χρησιμοποιήσουμε ένα σχήμα ταξινόμησης που είναι βέλτιστο υπό την έννοια ότι, κατά μέσο όρο, η χρήση του δίνει τη μικρότερη πιθανότητα να κάνει λάθη ταξινόμησης. Ένας ταξινομητής που ελαχιστοποιεί τη συνολική μέση απώλεια ονομάζεται ταξινομητής Bayes. Ο ταξινομητής ελάχιστης απόστασης καθορίζεται εξ ολοκλήρου από τον μέσο όρο διανύσματος κάθε κατηγορίας. Ο ταξινομητής Bayes για τάξεις Gauss καθορίζεται εξ ολοκλήρου από τον μέσο όρο του διανύσματος και τον πίνακα συνδιακύμανσης κάθε κατηγορίας. Αυτές οι παράμετροι που καθορίζουν τους ταξινομητές λαμβάνονται μέσω μιας διαδικασίας εκπαίδευσης. Κατά τη διάρκεια αυτής της διαδικασίας, χρησιμοποιούνται πρότυπα κατάρτισης για κάθε κλάση για τον υπολογισμό αυτών των παραμέτρων και των προδιαγραφών κάθε κατηγορίας.
Νευρωνικά Δίκτυα: Λαμβάνοντας υπόψη ένα δίκτυο back-propagation, αυτό το δίκτυο αποτελείται από διάφορα στρώματα διασυνδεδεμένων στοιχείων. Ένα διάνυσμα χαρακτηριστικών εισέρχεται στο δίκτυο στο επίπεδο εισόδου. Κάθε στοιχείο του στρώματος υπολογίζει ένα σταθμισμένο άθροισμα των εισροών του και το μετατρέπει σε έξοδο από μια μη γραμμική συνάρτηση. Κατά τη διάρκεια της εκπαίδευσης τα βάρη σε κάθε σύνδεση ρυθμίζονται μέχρι να επιτευχθεί η επιθυμητή έξοδος. Ένα πρόβλημα των νευρωνικών δικτύων στο OCR μπορεί να είναι η περιορισμένη προβλε-

ψιμότητα και γενικότητα τους, ενώ ένα πλεονέκτημα είναι ο προσαρμοστικός τους χαρακτήρας.

2.3.2 Δομική Αναγνώριση

Στο πλαίσιο της δομικής αναγνώρισης, οι συντακτικές μέθοδοι συγκαταλέγονται στις πιο διαδεδομένες προσεγγίσεις. Υπάρχουν και άλλες τεχνικές, αλλά είναι λιγότερο γενικές και δεν θα ασχοληθούμε με εκείνες.
Συντακτικές Μέθοδοι: Τα μέτρα ομοιότητας που βασίζονται στις σχέσεις μεταξύ των δομικών στοιχείων μπορούν να διαμορφωθούν χρησιμοποιώντας γραμματικές έννοιες. Η ιδέα είναι ότι κάθε τάξη έχει τη δική της γραμματική που καθορίζει τη σύνθεση του χαρακτήρα. Μια γραμματική μπορεί να αναπαρασταθεί σαν γραμματοσειρά ή δέντρο και τα δομικά συστατικά που εξάγονται από έναν άγνωστο χαρακτήρα ταιριάζουν με τις γραμματικές της κάθε τάξης. Υποθέστε ότι έχουμε δύο διαφορετικές τάξεις χαρακτήρων που μπορούν να δημιουργηθούν από τις δύο γραμματικές $G1$ και $G2$, αντίστοιχα. Δεδομένου ενός άγνωστου χαρακτήρα, λέμε ότι είναι πιο όμοια με την πρώτη τάξη αν μπορεί να παραχθεί από τη γραμματική $G1$, αλλά όχι από τη $G2$.

2.4 Μετεπεξεργασία

Ομαδοποίηση: Το αποτέλεσμα της απλής αναγνώρισης συμβόλων σε ένα έγγραφο είναι ένα σύνολο διαφορετικών συμβόλων. Ωστόσο, αυτά τα σύμβολα δεν περιέχουν συνήθως αρκετές πληροφορίες. Αντίθετα, θα θέλαμε να συσχετίσουμε τα μεμονωμένα σύμβολα που ανήκουν στην ίδια συμβολοσειρά μεταξύ τους, συνθέτοντας λέξεις και αριθμούς. Η διαδικασία εκτέλεσης αυτής της συσχέτισης των συμβόλων σε συμβολοσειρές, συνήθως αναφέρεται ως ομαδοποίηση. Η ομαδοποίηση των συμβόλων σε συμβολοσειρές βασίζεται στη θέση των συμβόλων στο έγγραφο. Τα σύμβολα που βρίσκονται αρκετά κοντά το ένα στο άλλο ομαδοποιούνται.

Για τις γραμματοσειρές με συγκεκριμένο βήμα, η διαδικασία ομαδοποίησης είναι αρκετά εύκολη καθώς η θέση καθενός από τους χαρακτήρες είναι γνωστή. Για τους διάφορους τύπους χαρακτήρων η απόσταση μεταξύ χαρακτήρων είναι μεταβλητή. Ωστόσο, η απόσταση μεταξύ των λέξεων είναι συνήθως σημαντικά μεγαλύτερη από την απόσταση μεταξύ των χαρακτήρων και επομένως η ομαδοποίηση είναι ακόμα εφικτή. Τα πραγματικά προβλήματα εμφανίζονται για χειρόγραφους χαρακτήρες ή όταν το κείμενο είναι λοξό.

2.4.1 Εντοπισμός Και Διόρθωση Λαθών

Μέχρι την ομαδοποίηση, κάθε χαρακτήρας έχει αντιμετωπιστεί ξεχωριστά. Ωστόσο, σε προχωρημένα προβλήματα οπτικής αναγνώρισης κειμένου, ένα σύστημα που αποτελείται μόνο από αναγνώριση ενός χαρακτήρα δεν θα είναι αρκετό. Ακόμη και τα καλύτερα συστήματα αναγνώρισης δεν θα δώσουν το 100% ποσοστό σωστής αναγνώρισης όλων των χαρακτήρων, αλλά ορισμένα από αυτά τα σφάλματα μπορεί να ανιχνευθούν ή και να διορθωθούν χρησιμοποιώντας τα συμφραζόμενα.

Υπάρχουν δύο κύριες προσεγγίσεις, όπου η πρώτη χρησιμοποιεί τη δυνατότητα αλληλουχίας χαρακτήρων που εμφανίζονται μαζί. Αυτό μπορεί να γίνει με τη χρήση κανόνων που καθορίζουν τη σύνταξη της λέξης, λέγοντας για παράδειγμα ότι μετά από μια πρόταση θα πρέπει να υπάρχει συνήθως ένα κεφαλαίο γράμμα. Επίσης, για διαφορετικές γλώσσες οι πιθανότητες δύο ή περισσότερων χαρακτήρων που εμφανίζονται μαζί σε μια ακολουθία μπορούν να υπολογιστούν και μπορούν να χρησιμοποιηθούν για την ανίχνευση σφαλμάτων. Για παράδειγμα, στην αγγλική γλώσσα η πιθανότητα εμφάνισης "k" μετά από ένα "h" σε μια λέξη είναι μηδέν, και εάν ανιχνευθεί ένας τέτοιος συνδυασμός, τότε υποθέτουμε ότι υπάρχει λάθος.

Μια άλλη προσέγγιση είναι η χρήση λεξικών, η οποία έχει αποδειχθεί η πιο αποτελεσματική μέθοδος ανίχνευσης και διόρθωσης σφαλμάτων. Λαμβάνοντας μια λέξη, στην οποία μπορεί να υπάρχει κάποιο σφάλμα, η λέξη ελέγχεται στο λεξικό. Εάν η λέξη δεν υπάρχει στο λεξικό, έχει εντοπιστεί ένα σφάλμα και μπορεί να διορθωθεί αλλάζοντας τη λέξη με την πιο όμοια λέξη που υπάρχει. Οι πιθανότητες που προκύπτουν από την ταξινόμηση μπορούν να βοηθήσουν στην αναγνώριση του χαρακτήρα που έχει ταξινομηθεί εσφαλμένα. Εάν η λέξη υπάρχει στο λεξικό, αυτό δυστυχώς δεν αποδεικνύει ότι δεν προέκυψε κανένα σφάλμα. Ένα σφάλμα μπορεί να έχει μετασχηματίσει τη λέξη από την σωστή λέξη σε μια άλλη, και τέτοια σφάλματα είναι μη ανιχνεύσιμα με αυτή τη μέθοδο. Το μειονέκτημα των μεθόδων του λεξικού είναι ότι οι έρευνες και οι συγκρίσεις που αναφέρονται είναι χρονοβόρες.

Κεφάλαιο 3

Αναγνώριση Μουσικών Κειμένων

Η μουσική αποτελεί από τα πρώτα κιόλας χρόνια της ανθρώπινης ζωής πολύ σημαντικό κομμάτι κουλτούρας και έκφρασης. Ως συνέπεια αυτού, με την ανάπτυξη της επιστήμης των υπολογιστών, έχει αυξηθεί το ενδιαφέρον της ψηφιοποίησης μουσικών κειμένων τόσο για την ευκολότερη πρόσβαση σε αυτά, όσο και για την αποθήκευσή τους. Ποια τεχνολογία όμως μπορεί να βοηθήσει στη λύση αυτού του προβλήματος;

Το OCR φαίνεται να μπορεί να προσφέρει πολύ σημαντική βοήθεια στην επίλυση του, διότι πρακτικά μιλάμε για την αναγνώριση χειρόγραφων συμβόλων. Αυτή η σκέψη ώθησε στην ανάπτυξη μιας υποκατηγορίας του OCR, το OMR (Optical Music Recognition ή αυτόματη αναγνώριση χειρόγραφων μουσικών κειμένων. Οι δύο τεχνολογίες είναι αρκετά κοντινές, με την κύρια διαφορά ότι το OMR επιδιώκει την μετάφραση ενός χειρόγραφου μουσικού κειμένου απευθείας σε αναγνώσιμο από υπολογιστή αρχείο. Ο συνεχώς αναπτυσσόμενος τομέας του OMR λοιπόν, πραγματεύεται το πρόβλημα ψηφιοποίησης μουσικών κειμένων. Κάποια σημαντικά πλεονεκτήματα που προκύπτουν από την επίλυση του παραπάνω προβλήματος είναι η μετάφραση τους σε αναπαραγωγή τους, μουσικολογική ανάλυση, μετάφραση σε άλλα είδη μουσικής (όπως μουσική Braille) και επαναεπεξεργασία. Παρακάτω θα δούμε τις μεθόδους και τους τρόπους με τους οποίους μπορεί να επιτευχθεί η αυτόματη αναγνώριση ενός χειρόγραφου μουσικού κειμένου καθώς και τα προβλήματα που αντιμετωπίζει το OMR και πως προσεγγίζει πιθανές λύσεις.

Σε αυτό το σημείο, αξίζει να αναφερθεί ότι υπάρχουν ήδη κάποια προγράμματα αναγνώρισης χειρόγραφων μουσικών κειμένων. Οι επιλογές βέβαια είναι κυρίως ανάμεσα σε δύο κατηγορίες καθεμία από τις οποίες έχει τα δικά της αρνητικά. Στην πρώτη κατηγορία ανήκουν προγράμματα όπως το Sibelius (δημιουργία το 2005), τα οποία δεν αναγνωρίζουν τονικότητα και αξία. Ο χρήστης καλείται να την εισαγάγει επιπρόσθετα. Το πρόβλημα αυτής της μεθόδου είναι ο χρόνος που χρειάζεται να δαπανήσει ο χρήστης προκειμένου να ολοκληρώσει μία παρτιτούρα με πολλές δια-

φορετικές αξίες ανά μέτρο. Η δεύτερη κατηγορίας περιέχει προγράμματα όπως το Music Notepad (δημιουργία 1998), στην οποία ο χρήστης χρειάζεται να μάθει ένα σύνολο από κινήσεις. Κάθε χειρονομία αντιστοιχεί και σε διαφορετική νότα, παύση, αξία καλύπτοντας όλο το πιθανό φάσμα που μπορεί ο χρήστης να χρειαστεί. Αυτή η κατηγορία, αν και αποδίδει σε πολύ ικανοποιητικό ποσοστό αναγνώρισης, επιβαρύνει τον μουσικό με μία μη απαραίτητη εργασία, να μάθει όλες αυτές τις χειρονομίες.

3.1 Μουσική Σημειογραφία

Η μουσική σημειογραφία ανά τους αιώνες εξελίσσεται και αλλάζει. Αυτό συμβαίνει λόγω της προσπάθειας πολλών καλλιτεχνών και μουσικών να αναπαραστήσουν τις μουσικές τους ιδέες με σύμβολα. Η δική μας προσέγγιση θα βασιστεί στην κοινή μουσική σημειογραφία (υπάρχουν και προγράμματα που έχουν ασχοληθεί με άλλες σημειογραφίες όπως μεσαιωνική).

Προκειμένου να φτάσουμε στην επίλυση τους προβλήματος χρειάζεται να πρώτα να εξοικειωθούμε με τις διαφοροποιήσεις σε κάθε σημειογραφία. Για να αντιληφθούμε μία σημειογραφία αρκεί να κατανοήσουμε τις πληροφορίες που απεικονίζουν τα μουσικά σύμβολα στα πλαίσιά της. Για παράδειγμα στην κοινή σημειογραφία οι νότες καθορίζονται από τα ακόλουθα τέσσερα στοιχεία:

1. Ύψος
2. Αξία
3. Δυναμική
4. Τονικότητα

Αφού κατανοήσουμε αυτό το σύνολο κανόνων που μπορούν να επηρεάσουν το μουσικό μας κείμενο μπορούμε να συνεχίσουμε τη διαδικασία επίλυσης του προβλήματος.

3.2 Ιστορικά Στοιχεία

Η πρώτη έρευνα στον τομέα του OMR ξεκίνησε το 1966 όταν ο Dennis Pruslin έκανε την πρώτη απόπειρα αυτόματης αναγνώρισης χειρόγραφης μουσικής παρτιτούρας. Το σύστημα του κατάφερε να αναγνωρίσει κεφαλές νοτών και συγχορδίες. Το 1970 ο Prerau εισήγαγε την ιδέα του διαχωρισμού της σαρωμένης εικόνας προκείμενου να ανιχνευτούν βασικά στοιχεία της κάθε σημειογραφίας. Με την ευκολότερη πρόσβαση σε μηχανές οπτικού σαρώματος, το 1980 η έρευνα του τομέα αυξήθηκε σημαντικά. Μία πολύ ενδιαφέρουσα εφεύρεση έλαβε χώρα στην Ιαπωνία το 1984. Ένα ρομπότ

με το όνομα WABOT-2 ήταν το πρώτο που κατάφερε να αναγνωρίζει απλές νότες και να τις αναπαράγει σε ένα πιάνο. Το 1997 ο D. Bainbridge συγκέντρωσε όλες τις ήδη υπάρχουσες πληροφορίες και πρότεινε μία επεκτάσιμη μέθοδο αναγνώρισης μουσικών κειμένων, το οποίο δε βασιζόταν σε συγκεκριμένα χαρακτηριστικά και σημειογραφία. Όπως αναφέρεται και στη διατριβή του με τίτλο Extensible Optical Music Recognition που δημοσιεύτηκε το 1997 στο πανεπιστήμιο του Canterbury, τα ήδη υπάρχοντα συστήματα μέχρι τότε έχουν επιτύχει την αναγνώριση περιορισμένου αριθμού μουσικών συνόλων. Η πρόταση του εστιάζει σε έξι σημαντικά βήματα με τα οποία μπορούμε να αυξήσουμε τον αριθμό των αναγνωρίσιμων μουσικών συνόλων. Μαζί με τον T. Bell κατάφεραν να δημιουργήσουν ένα γενικό πλαίσιο για το OMR το οποίο χρησιμοποιήθηκε από πολλούς ερευνητές.

Μία πιο πρόσφατη δημοσίευση στον τομέα έγινε το 2012 από τους Ana Rebelo, Ichiro Fujinaga, Filipe Paszkiewicz, Andre R.S. Marcal, Carlos Guedes και Jaime S. Cardoso με τίτλο Optical Music Recognition - State-of-the-Art and Open Issues και συγκρίνει την απόδοση των μεταγενέστερων αλγορίθμων OMR με τους παλιούς.

3.3 Γενικό Πλαίσιο Αναγνώρισης

Το πρόβλημα της αυτόματης αναγνώρισης είναι αρκετά πολύπλοκο και σχετίζεται με αρκετά κομμάτια της επιστήμης των υπολογιστών. Συνεπώς στο παρασκήνιο έχουν εμφανιστεί διάφορες λύσεις. Κάθε λύση ακολουθεί τη δική της πορεία και καταλήγει σε παρόμοιας επιτυχίας αποτελέσματα μέχρι σήμερα. Η βασική μέθοδος που χρησιμοποιεί η πλειοψηφία των υπαρχόντων αλγορίθμων μπορεί να χωριστεί σε τέσσερα βασικά βήματα:

1. Επεξεργασία εικόνας
2. Τμηματοποίηση
3. Αναγνώριση συμβόλων
4. Αναδόμηση

Εμείς θα επικεντρωθούμε κυρίως στην αναγνώριση συμβόλων στα επόμενα κεφάλαια. Πάμε λοιπόν να δούμε πως καθένα από αυτά μας οδηγεί στη λύση.

3.3.1 Επεξεργασία Εικόνας

Αρχικά λοιπόν, αναπροσαρμόζουμε την εικόνα έτσι ώστε να κάνουμε τη φάση της αναγνώρισης πιο εύκολη και αποδοτική. Συνήθεις διαδικασίες που λαμβάνουν χώρα κατ΄ αυτό το βήμα είναι βελτίωση, μορφολογικές αλλαγές, αφαίρεση θορύβων,

μετατροπή σε δυαδική μορφή. Λόγω του ότι η πλειοψηφία των συστημάτων OMR χρησιμοποιεί την μετατροπή σε δυαδική μορφή θα αναλύσουμε εκείνη τη διαδικασία.

Κατά τη διαδικασία αυτή η εικόνα που εισάγουμε μετατρέπεται σε δυαδική μορφή. Εκεί γίνεται διαχωρισμός του συμβόλου που μας ενδιαφέρει με το φόντο πάνω στο οποίο είναι γραμμένο. Συνήθως αυτή η διαδικασία γίνεται αυτόματα χωρίς ιδιαίτερη γνώση του περιεχομένου της εικόνας. Αυτό συμβαίνει διότι μειώνοντας τον όγκο του αρχείου γλιτώνουμε σημαντικό χρόνο σε καθένα από τα επόμενα βήματα. Επίσης είναι πολύ ευκολότερο να συντάξουμε αλγόριθμο που επεξεργάζεται εικόνα σε δυαδική μορφή παρά μία έγχρωμη εικόνα.

3.3.2 Τμηματοποίηση

Κατά το κομμάτι της τμηματοποίησης τα μουσικά σύμβολα διαχωρίζονται σε μικρότερα κομμάτια. Ανάλογα το δείγμα που καλούμαστε να αναγνωρίσουμε το βήμα αυτό μπορεί να φανεί πολύ χρήσιμο. Για παράδειγμα αν το δείγμα μας δεν αποτελείται από μεμονωμένα μουσικά σύμβολα είναι πολύ πιθανό να αποτελείται από κάποιο μουσικό κείμενο γραμμένο σε πεντάγραμμο. Στην περίπτωση αυτή πριν τη διαδικασία αναγνώρισης είναι απαραίτητο με κάποιο τρόπο να ξεχωρίσουμε τα σύμβολα από το πεντάγραμμο. Με τη μέθοδο του αλγορίθμου συμπίεσης δεδομένων (run-length encoding) είναι εφικτό να διακρίνουμε ποιο κομμάτι αντιστοιχεί στο πεντάγραμμο. Αρκεί να εντοπίσουμε τους πιο συχνά εμφανιζόμενους αριθμούς λευκών και μαύρων πίξελ αντίστοιχα ανά στήλη. Έπειτα μπορούμε να απομονώσουμε τα μουσικά σύμβολα γυρίζοντας σε κάθε στήλη και ψάχνοντας για τα κομμάτια τα οποία διαφοροποιούνται από το μοτίβο των γραμμών του πενταγράμμου. Έπειτα κωδικοποιούμε κάθε σύμβολο με τον αλγόριθμο συμπίεσης δεδομένων.

Γραμμές πενταγράμμου: Μία στοιχειώδης λειτουργία κάθε αλγορίθμου στο OMR είναι η αναγνώριση των γραμμών του πενταγράμμου. Αυτό συμβαίνει διότι οι γραμμές αυτές ορίζουν ένα δισδιάστατο σύστημα συντεταγμένων το οποίο είναι απαραίτητο για την κατανόηση της κοινής μουσικής σημειολογίας.

Δυστυχώς η αναγνώριση αυτή δεν πραγματοποιείται πάντα με απόλυτη επιτυχία. Υπάρχει περίπτωση οι γραμμές αυτές να μην είναι σωστά ζωγραφισμένες και να χαλάει η παραλληλία τους ή ακόμη και να διαφέρουν σε πάχος. Παρά το γεγονός ότι υπάρχουν αρκετές μέθοδοι για αυτή τη διαδικασία καμία δεν επιτυγχάνει το απόλυτο ποσοστό επιτυχίας. Η πιο συνηθισμένη μέθοδος που χρησιμοποιείται από τους πιο στοιχειώδεις αλγορίθμους είναι η οριζόντια προβολή. Η μέθοδος αυτή μετατρέπει τη δυαδική εικόνα σε ιστόγραμμα συγκριτικά με το πόσα μαύρα πίξελ υπάρχουν σε κάθε σειρά. Από τη στιγμή που οι γραμμές αυτές είναι οριζόντιες, ο αλγόριθμος εύκολα τις αναγνωρίζει σαν τα τοπικά μέγιστα κάθε γραμμής. Ακόμη και σήμερα ένας τρόπος ο οποίος θα αποδίδει σε κάθε περίπτωση παραμένει άλυτο πρόβλημα στον τομέα του OMR.

Μουσικά σύμβολα: Μετά την αναγνώριση των γραμμών σειρά έχουν τα μουσικά σύμβολα. Αρχικά πρέπει να τα απομονώσουμε. Αυτό μπορεί να πραγματοποιηθεί είτε αφαιρώντας τις γραμμές είτε αγνοώντας τις. Υπάρχουν διάφορα επιχειρήματα που υποστηρίζουν και τις δύο μεθόδους και η επιλογή ανάμεσα σε αυτές τις δύο είναι καθαρά περιστασιακή. Ένα παράδειγμα ενός απλού αλγόριθμου αφαίρεσης γραμμών είναι το εξής. Ξεκινάμε να αφαιρούμε κομμάτι κομμάτι τη γραμμή αντικαθιστώντας τα μαύρα πίξελ που την ορίζουν με λευκά. Ελέγχουμε την τριγύρω περιοχή προκειμένου να αποφύγουμε, όσο το δυνατόν περισσότερο, να αλλοιώσουμε κάποιο σύμβολο. Το πρόβλημα είναι ότι όσο προσεκτική δουλειά και να κάνει ο αλγόριθμος μας είναι σχεδόν σίγουρο ότι μέρος του του συμβόλου θα σβηστεί σαν μέρος γραμμής. Αυτό συμβαίνει διότι τα μουσικά σύμβολα έχουν καμπύλες ή κομμάτια που ακουμπούν εφαπτομενικά σε γραμμές.

Έπειτα το σύμβολο χωρίζεται σε τμήματα τα οποία ονομάζονται περιοχές ενδιαφέροντος. Έτσι μπορούμε να εξετάσουμε τα χαρακτηριστικά που διαφοροποιούν κάθε σύμβολο (κεφαλή, στέμμα, ουρά, παύση) συγκρίνοντας συγκεκριμένα κομμάτια του. Η μέθοδος αυτή βέβαια διαφέρει από αλγόριθμο σε αλγόριθμο. Για παράδειγμα υπάρχουν μέθοδοι κατά τις οποίες εξετάζουν τα σύμβολο σαν συνολικές οντότητες χωρίς να τα διαχωρίζουν.

3.3.3 Αναγνώριση Συμβόλων

Το βήμα στο οποίο θα επικεντρωθεί η εργασία. Κατά το βήμα αυτό τα κομμάτια που έχουμε διαχωρίσει εισάγονται στον αλγόριθμο κατηγοριοποίησης. Εκεί ο αλγόριθμος προσπαθεί συγκρίνοντας τα σύμβολα μας με τα προκαθορισμένα σύμβολα να τους προσδώσει μία ετικέτα. Υπάρχουν βέβαια πιθανές αστοχίες και σε αυτό το βήμα. Αυτές μπορούν να προκύψουν είτε λόγω λανθασμένης αποκοπής μέρους του συμβόλου είτε από περίπλοκα σύμβολα που το σχέδιο τους έχει τέμνουσες μεταξύ τους γραμμές. Πρόκειται λοιπόν για μία αρκετά λεπτή διαδικασία η οποία πολλές φορές συνδυάζεται με το προηγούμενο βήμα.

Τα κύρια κομμάτια τα οποία διαχωρίσαμε βοηθούν τον αλγόριθμο κατηγοριοποίησης να τελειώσει τη ζητούμενη διαδικασία. Οι αλγόριθμοι κατηγοριοποίησης είναι διάφοροι και καθένας από αυτούς λειτουργεί με τη δική του μέθοδο. Στα επόμενα κεφάλαια καθώς και στην εφαρμογή που θα εξετάσουμε, θα ασχοληθούμε με τη μέθοδο κοντινότερων γειτόνων (KNN) και με τα Support Vector Machines και θα παρουσιάσουμε τα μαθηματικά που κρύβονται πίσω από τις μεθόδους αυτές. Το επόμενο κεφάλαιο θα παρουσιάσει τους τρόπους λειτουργίας των αλγορίθμων αυτών.

3.3.4 Σημασιολογική Αναδόμηση

Η αναπόφευκτη διαδικασία που κάθε σύστημα OMR είναι αναδημιουργήσει τη μουσική σημασιολογία από προηγουμένως αναγνωρισμένα γραφικά πρότυπα. Οι πληρο-

φορίες αυτές αποθηκεύονται σε μια κατάλληλη δομή δεδομένων. Αυτή η διαδικασία απαιτεί υποχρεωτικά μία ερμηνεία της σχέσης μεταξύ των αντικείμενων που βρίσκονται στην εικόνα. Οι σχέσεις στην κοινή μουσική σημειογραφία είναι ουσιαστικά δύο διαστάσεων και οι πληροφορίες θέσεων είναι πολύ σημαντικές. Για παράδειγμα, μία τελεία μπορεί να αλλάξει τη διάρκεια ενός συμβόλου, αν είναι τοποθετημένη στα δεξιά της κεφαλής του.

Τέτοιοι μουσικοί συντακτικοί κανόνες μπορούν να επισημανθούν χρησιμοποιώντας γραμματικές. Οι κανόνες της γραμματικής θα καθιστούν σημασιολογικά έγκυρους τους μουσικούς συμβολισμούς. Επίσης θα καθορίζουν με ποιο τρόπο θα πρέπει να κατανεμηθούν τα μουσικά χαρακτηριστικά.

Η τελευταία και θεμελιώδης πτυχή των συστημάτων μας είναι ο μετασχηματισμός των σημειολογικά αναγνωρισμένων αποτελεσμάτων σε μορφή κωδικοποίησης. Η κωδικοποίηση αυτή θα είναι ουσιαστικά ένα μοντέλο του μουσικού συμβόλου που αναγνωρίσαμε. Έπειτα αποθηκεύουμε τα νέα μας δεδομένα στην κατάλληλη μορφή για το σύστημα το οποίο χρησιμοποιούμε.

Κεφάλαιο 4

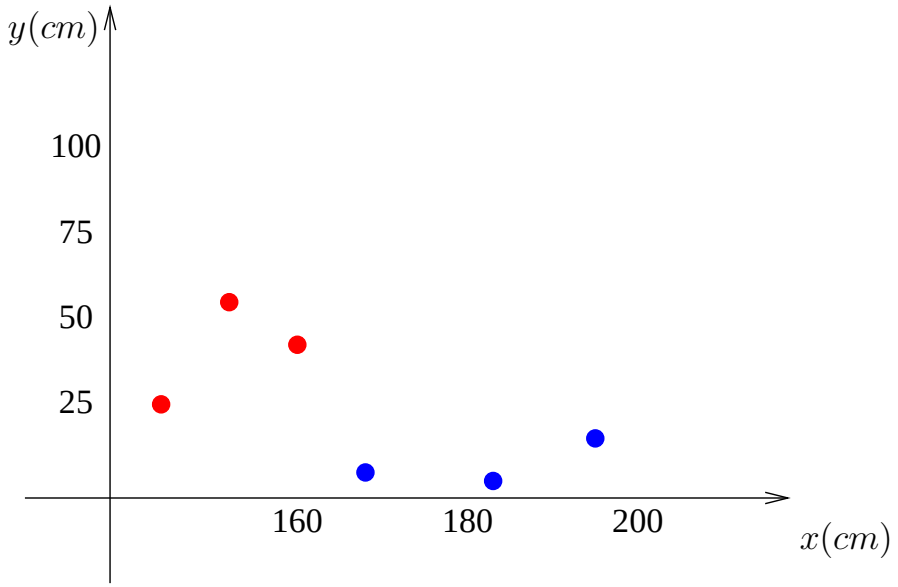
Κατηγοριοποίηση Δεδομένων

4.1 Μηχανές Διανυσμάτων Υποστήριξης (Support Vector Machines)

4.1.1 Εισαγωγή

Οι Μηχανές Διανυσμάτων Υποστήριξης (Support Vector Machines - SVM) αποτελούν αλγόριθμο κατηγοριοποίησης και ταξινόμησης στην κατηγορία της επιβλεπόμενης μάθησης, του τομέα της μηχανικής μάθησης. Η κύρια χρήση του αλγόριθμου σήμερα είναι σε προβλήματα ταξινόμησης και οι επιδόσεις του είναι αρκετά ικανοποιητικές.

Ξεκινώντας, οι Μηχανές Διανυσμάτων Υποστήριξης κατά ένα μεγάλο ποσοστό, όπως προαναφέραμε, ασχολούνται με το κομμάτι της ταξινόμησης. Αλλά πώς ακριβώς δουλεύουν; Αρχικά πρέπει να καταλάβουμε πώς ακριβώς δουλεύει η ανάλυση ταξινόμησης Θα δώσουμε ένα πολύ απλουστευμένο παράδειγμα ενός προβλήματος της ανάλυσης ταξινόμησης Έστω ότι θέλουμε να ταξινομήσουμε ένα τυχαίο δείγμα ανθρώπων σε άνδρες και γυναίκες. Σκοπός μας στην προκειμένη περίπτωση είναι να δημιουργήσουμε ένα πρόγραμμα με το οποίο ένα ρομπότ ή μία μηχανή θα μπορούσε να εκτιμήσει το φύλο κάθε ατόμου στο δείγμα αυτό. Αρχικά, πρέπει να θέσουμε κάποιους κανόνες σύμφωνα με τους οποίους θα ξεχωρίζουμε τους άνδρες από τις γυναίκες με βάση τα εξωτερικά τους χαρακτηριστικά Υποθέτουμε ότι κατά μέσο όρο οι άνδρες έχουν μεγαλύτερο ύψος από τις γυναίκες και επίσης ότι κατά μέσο όρο οι γυναίκες έχουν μεγαλύτερα σε μέγεθος μαλλιά από τους άνδρες. Σε ένα τυχαίο δείγμα έξι ατόμων των οποίων τα χαρακτηριστικά φαίνονται στο σχήμα 4.1 μπορούμε να δούμε το ύψος και το μέγεθος των μαλλιών σε εκατοστά κάθε ατόμου. Εύκολα μπορεί κανείς να παρατηρήσει ότι οι γυναίκες, οι οποίες είναι μαρκαρισμένες με το κόκκινο χρώμα, είναι σε διαφορετικό κομμάτι σε σχέση με τους άνδρες, που είναι με το μπλε χρώμα. Στο πρόβλημα αυτό έχουμε ένα καλά διαχωρισμένο δείγμα με απο-

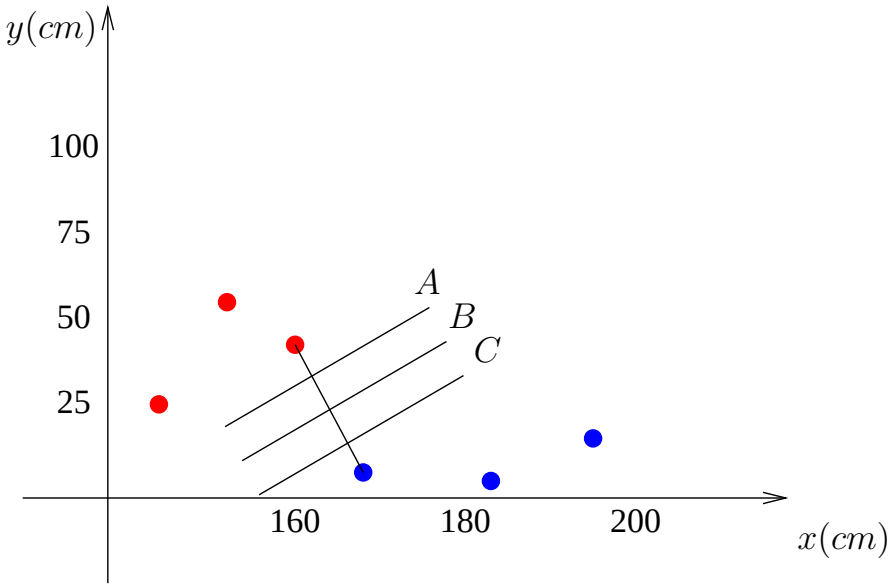


Σχήμα 4.1: Αναπαράσταση Δείγματος

τέλεσμα να έχουν δημιουργηθεί δύο διαφορετικές περιοχές, μία που περιέχει άνδρες και μία που περιέχει γυναίκες, πράγμα που κάνει τη δουλειά της ταξινόμησης πολύ ευκολότερη.

Το παραπάνω πρόβλημα αποτελεί ένα απλουστευμένο πρόβλημα του τομέα της ανάλυσης και ταξινόμησης δεδομένων. Προκειμένου να φτάσουμε σε μία μέθοδο η οποία θα μας επιλύει απλά προβλήματα όπως το προαναφερθέν, δυσκολότερα ή ακόμα και πολύ σύνθετα πολύπλοκα προβλήματα πρέπει αρχικά να κατανοήσουμε τι προσπαθούμε να επιτύχουμε. Όπως μπορούμε να δούμε στο σχήμα 4.1 το δείγμα έχει χωριστεί σε δύο ομάδες. Αυτό που θέλουμε είναι να βρούμε μία μέθοδο η οποία θα πετύχει την ταξινόμηση του δείγματος μου σε άνδρες και γυναίκες με βάση τους κανόνες που θέσαμε. Για αρχή παρατηρούμε ότι κάθε άτομο στο παραπάνω δείγμα αποτελεί και ένα σημείο του χώρου που ορίζει η γραφική παράσταση, η οποία απεικονίζει το κάθε άτομο μέσα στο δείγμα ως ένα σημείο συναρτήσει του ύψους του σε εκατοστά (άξονας xx') και του μεγέθους, σε μήκος μετρημένο σε εκατοστά, των μαλλιών του (άξονας yy'). Ο σκοπός μας είναι να βρούμε μια γραμμή η οποία θα διαχωρίσει το δείγμα. Δηλαδή ψάχνουμε εκείνη την ευθεία για την οποία ισχύει ότι κάθε σημείο πάνω από αυτή θα είναι γυναίκα ενώ κάθε σημείο κάτω από αυτή θα είναι άνδρας.

Η γραμμή-όριο στην οποία έγινε αναφορά ονομάζεται όριο απόφασης (decision boundary) στον τομέα των Μηχανών Διανυσμάτων Υποστήριξης (SVM). Στο συγκεκριμένο παράδειγμα είναι πολύ εύκολο να βρεθεί όριο απόφασης και μάλιστα υ-



Σχήμα 4.2: Υπερεπίπεδα

πάρχουν αρκετές λύσεις οι οποίες ικανοποιούν τη συνθήκη που θέσαμε στην αρχή. Το γεγονός αυτό γεννά την απορία του αν υπάρχει σωστό ή λάθος όριο και σε περίπτωση που υπάρχει, ποιο είναι το κριτήριο επιλογής. Στο σχήμα 4.2 μπορούμε να διακρίνουμε κάποιες γραμμές καθεμία από τις οποίες διαχωρίζει πλήρως τα δύο μας δείγματα. Η σωστή επιλογή ενός ορίου απόφασης γίνεται με βάση το support vector του εκάστοτε δείγματος. Ο support vector αποτελείται από $n + 1$ στοιχεία, όπου n η διάσταση του πίνακα του δείγματος μας, τα οποία βρίσκονται πιο κοντά στο κενό το οποίο χωρίζει τα δύο δείγματα. Στο παράδειγμα μας, ο support vector, θα αποτελείται από τα τρία κοντινότερα σημεία στα πιθανά όρια A, B και C του σχήματος 4.2. Για να βρούμε το σωστό decision boundary ψάχνουμε για την ευθεία εκείνη που χωρίζει όλα τα σημεία του δείγματος μας έτσι ώστε να έχουν τη μεγαλύτερη δυνατή απόσταση από το όριο απόφασης. Παρακάτω θα δούμε ότι αντί να εξεταστεί η απόσταση κάθε σημείου από το όριο απόφασης αρκεί να εξετάσουμε τις αποστάσεις ως προς τα σημεία του support vector. Η απλότητα του συγκεκριμένου προβλήματος, δεν καθιστά προφανή τον λόγο επιλογής του ορίου με βάση την απόσταση του από τα σημεία αυτά. Σε μεγαλύτερα δείγματα όπου τα σημεία έρχονται πολύ πιο κοντά το ένα στο άλλο και πιθανότατα πιο κοντά στο όριο απόφασης, όσο πιο μεγάλη είναι η απόσταση, τόσο μικρότερο το σφάλμα γενίκευσης του ταξινομητή.

4.1.2 Μαθηματική Ανάλυση των Μηχανών Διανυσμάτων Υποστήριξης (SVM)

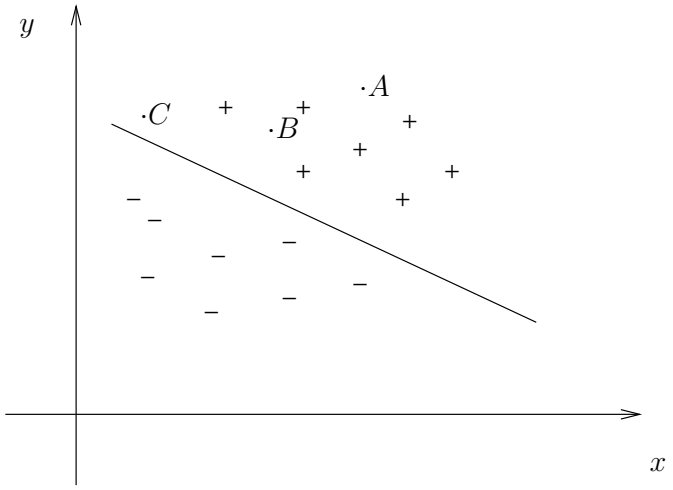
Το πρώτο βήμα για να κατανοήσουμε την μαθηματική πλευρά των Μηχανών Διανυσμάτων Υποστήριξης (SVM) είναι να καταλάβουμε την έννοια τις απόστασης (margin). Έστω x τα εισαγόμενα δεδομένα, y τα δεδομένα εκπαίδευσης και θ βάρος. Έστω ένα πρόβλημα παλινδρόμησης, στο οποίο η πιθανότητα $p(y = 1|x; \theta)$ διαμορφώνεται από την σχέση $h^\theta(x) = g(\theta^T x)$. Ένα στοιχείο x θα έχει σαν αποτέλεσμα $'1'$ αν και μόνο αν $h^\theta(x) \geq 0.5$ ή ισοδύναμα αν και μόνο αν $\theta^T x \geq 0$. Αν πάρουμε σαν παράδειγμα ένα θετικό στοιχείο δείγματος ($y = 1$), τότε όσο πιο πολύ μεγαλώνει η ποσότητα $\theta^T x$ απολύτως ανάλογα μεγαλώνει και το $h^\theta(x)$ και συνεπώς μεγαλώνει ο βαθμός βεβαιότητας μας για το αποτέλεσμα. Ανάλογα μπορούμε να συλλογιστούμε ότι για να προκύψει το αποτέλεσμα $y = 0$ θα πρέπει οι ποσότητες αυτές να είναι αρκετά μεγάλοι αρνητικοί αριθμοί. Δηλαδή θέλουμε $\theta^T x << 0$. Γενικά, η μέθοδος φαίνεται να λειτουργεί αν καταφέρουμε να βρούμε θ τέτοιο ώστε η ποσότητα $\theta^T x$ θα είναι πολύ μεγαλύτερη του μηδενός όταν θα έχω αποτέλεσμα $y^{(i)} = 1$ και αντίστοιχα πολύ μικρότερη του μηδέν αν το αποτέλεσμα είναι $y^{(i)} = 0$. Έχοντας αυτή τη βασική ιδέα συνεχίζουμε χρησιμοποιώντας τα λειτουργικά περιθώρια (functional margins).

Στο παρακάτω σχήμα 4.3 έχουμε κάποια + τα οποία αναπαριστούν θετικά στοιχεία εκπαίδευσης και κάποια − τα οποία θα είναι αρνητικά στοιχεία εκπαίδευσης. Επίσης έχουμε μια γραμμή απόφασης (η οποία προκύπτει $\theta^T x = 0$ και ονομάζεται γραμμή διαχωρισμού) και τρία σημεία A, B και C.

Παρατηρώντας το σημείο A αν μας ζητηθεί να προβλέψουμε σε ποία από τις δύο κατηγορίες ανήκει (όπου $y = 1$ τα θετικά και $y = 0$ τα αρνητικά) είναι πολύ λογικό να μαντέψουμε ότι είναι θετικό. Αντίστροφα, το σημείο C είναι πολύ κοντά στην διαχωριστική γραμμή. Παρόλο που μπορούμε να δούμε ότι είναι στη μεριά των θετικών στοιχείων, με μία πολύ μικρή αλλαγή της διαχωριστικής γραμμής μπορεί να βρεθεί στην αρνητική μεριά. Το σημείο B βρίσκεται ενδιάμεσα στις δύο αυτές, σχεδόν ακραίες, περιπτώσεις και μας δείχνει ότι αν ένα στοιχείο έχει απόσταση από τη διαχωριστική γραμμή κάνει τη δουλειά της κατηγοριοποίησης πολύ πιο απλή. Ο στόχος μας λοιπόν είναι να καταφέρουμε να διαχωρίσουμε το δείγμα με μεγάλο βαθμό βεβαιότητας και σιγουριάς.

4.1.3 Χρήσιμα Σύμβολα

Για αρχή θα χρειαστεί να αναφερθούμε σε μία νέα σημειογραφία προκειμένου να διευκολύνουμε τη διαδικασία της απόδειξης. Αρχικά, θεωρούμε μία γραμμική συνάρτηση κατηγοριοποίησης η οποία θα απευθύνεται σε προβλήματα δυαδικής κατηγοριοποίησης. Έχουμε ταμπέλες y που χαρακτηρίζονται από χαρακτηριστικά x η καθεμία. Αντί για πίνακα θ όπως είδαμε παραπάνω, θα χρησιμοποιήσουμε παραμέτρους w, b. Η συνάρτηση κατηγοριοποίησης θα είναι:



Σχήμα 4.3: Ρόλος Γραμμής Απόφασης

$$h_{w,b}(x) = g(w^T x + b)$$

Η συνάρτηση μας θα δίνει $g(z) = 1$ αν $z \geq 0$ ενώ σε κάθε άλλη περίπτωση θα έχουμε $g(z) = -1$. Κάνοντας χρήση των w, b έχουμε το πλεονέκτημα να μπορούμε να αντιμετωπίσουμε τον όρο b χωρίς εξάρτηση από τις υπόλοιπες παραμέτρους. Πρακτικά το b που έχουμε εδώ αντικαθιστά το θ_0 της προηγούμενης παραγράφου, ενώ το w παίρνει τη θέση του πίνακα $[\theta_1, \theta_2, ..., \theta_n]^T$

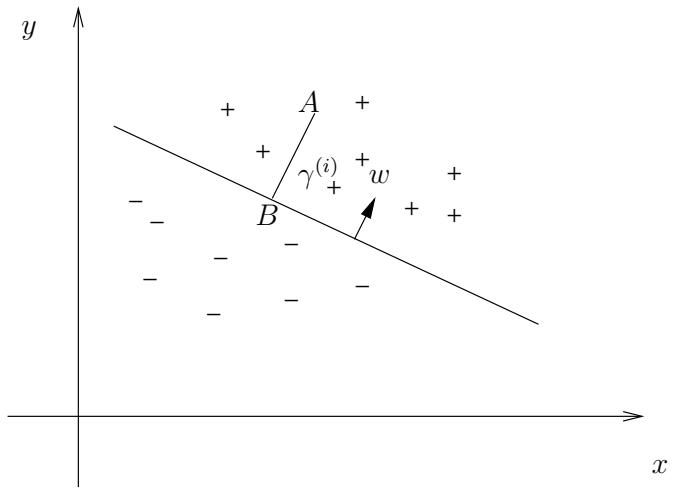
Σε αυτό το σημείο μπορούμε εύκολα να παρατηρήσουμε ότι ο τρόπος με τον οποίο ορίσαμε τη συνάρτηση g μπορεί να δώσει δύο πιθανά αποτελέσματα. Συνεπώς η διαδικασία που θα ακολουθηθεί θα είναι ένας έλεγχος για τον αν το αποτέλεσμα θα είναι η πρώτη περίπτωση ($z \geq 0$), αλλιώς θα μεταφερθούμε αυτόματα και χωρίς άλλο έλεγχο στη δεύτερη περίπτωση. Η μέθοδος αυτή διαφοροποιείται από εκείνη της λογιστικής παλινδρόμησης.

4.1.4 Ορισμός Αποστάσεων (Margins)

Σε αυτό το σημείο θα ορίσουμε τις αποστάσεις. Δεδομένου πάντα ενός δείγματος $(x^{(i)}, y^{(i)})$. Το functional margin των w, b ως προς το δείγμα μας θα είναι:

$$g^{(i)} = y^{(i)}(w^T x + b)$$

Σε περίπτωση που το $y^(i) = 1$, τότε προκειμένου να έχουμε μεγαλύτερη απόσταση, και συνεπώς η πρόβλεψη μας να είναι πιο βέβαιη και σωστή, θέλουμε το $w^T x + b$ να είναι ένας μεγάλος θετικός αριθμός. Με απόλυτη αναλογία στην περίπτωση όπου



Σχήμα 4.4: Υπερεπίπεδα

$y^{(i)} = -1$ για να πετύχουμε μεγαλύτερη απόσταση, θέλουμε το $w^T x + b$ να είναι ένας μεγάλος αρνητικός αριθμός. Είναι εύκολο να παρατηρήσουμε ότι αν η ποσότητα $y^{(i)}(w^T x + b) > 0$ τότε το πόρισμα που βγάλαμε για το συγκεκριμένο παράδειγμα είναι σωστή.

Ωστόσο στο functional margin υπάρχει ένα πρόβλημα. Για μία γραμμική συνάρτηση g η οποία λαμβάνει τιμές από το σύνολο $-1, 1$ αν αντικαταστήσουμε τα w, b με $2w, 2b$ τότε προκύπτει:

$$g(2w^T x + 2b)$$

Το αποτέλεσμα δεν επηρεάζεται άμεσα μιας και έχουμε δυαδική συνάρτηση. Οι $h_{w,b}$ και g δε θα επηρεαστούν, και αυτό διότι εξαρτώνται από τα πιθανά αποτελέσματα και όχι από την αξία της ποσότητας $w^t x + b$. Όμως, η αντικατάσταση των w, b με $2w, 2b$ είναι πρακτικά ο πολλαπλασιασμός της ήδη υπάρχουσας απόστασης μας με τον αριθμό δύο. Μπορούμε πολύ εύκολα να παρατηρήσουμε ότι κάνοντας χρήση της παραπάνω σκέψης είναι δυνατό να μεγαλώσουμε την απόσταση αυτή απειροστά, χωρίς να έχουμε να προσφέρουμε κάποια ουσιαστική λύση στο πρόβλημα. Ένας τρόπος να γλυτώσουμε από το πρόβλημα αυτό είναι η κανονικοποίηση της απόστασης.

Δεδομένου ενός δείγματος εκπαίδευσης $S = (x^{(i)}, y^{(i)}) : i = 1, ..., m$, επίσης ορίζουμε την απόσταση της συνάρτησης των (w, b) ως προς S, ως τη μικρότερη από τις αποστάσεις κάθε στοιχείου στο δείγμα μας. Αν το συμβολίσουμε με γ (σχήμα 4.4) τότε θα ισούται με:

$$\gamma = min_{i=1,...,m}\gamma_{(i)}$$

Συνεχίζοντας, πρέπει να βρούμε ένα τρόπο να υπολογίσουμε το $\gamma_{(i)}$. Σύμφωνα με το παραπάνω σχήμα 4.4 μπορούμε να δούμε ότι το $w/\|w\|$ είναι ένα μοναδιαίο διάνυσμα ομόρροπο με το w το οποίο είναι κάθετο στο υπερεπίπεδο. Αφού το A αναπαριστά το $x(i)$ τότε μπορούμε να πάρουμε το σημείο B από την αφαίρεση $x(i)-\gamma(i)$. Παρατηρούμε όμως ότι το συγκεκριμένο σημείο ανήκει στο υπερεπίπεδο απόφασης, και κάθε τέτοιο σημείο ικανοποιεί τη συνάρτηση $w^T x + b = 0$. Συνεπώς:

$$w^T \left(x^{(i)} - \gamma^{(i)} \frac{w}{\|w\|} \right) + b = 0$$

και ως προς $\gamma^{(i)}$ έχουμε

$$\gamma^{(i)} = \frac{w^T}{\|w\|} x^{(i)} + b = \frac{w^T}{\|w\|} x^{(i)} + \frac{b}{\|w\|}$$

Από την παραπάνω σχέση μπορούμε να ορίσουμε τη γεωμετρική απόσταση geometric margin των (w, b) ως προς ένα δείγμα $(x^{(i)}, y^{(i)})$ να είναι:

$$\gamma^{(i)} = y^{(i)} \left(\frac{w^T}{\|w\|} x^{(i)} + \frac{b}{\|w\|} \right)$$

Η γεωμετρική απόσταση που μόλις ορίσαμε μπορεί εύκολα να συσχετιστεί με την αρχική μας εκτίμηση. Αρκεί απλά να στη σχέση 4.1.4 να πάρουμε $\|w\| = 1$. Το πλεονέκτημα που κρύβει η νέα σχέση απόστασης που ορίσαμε είναι ότι επειδή είναι κανονικοποιημένη δεν παρουσιάζει πρόβλημα με την κλιμακοποίηση. Σε περίπτωση που αντικαταστήσουμε το w με $2w$ η απόσταση θα παραμείνει ακριβώς ίδια. Με προσαρμογή στα νέα δεδομένα λοιπόν, δεδομένου δείγματος $S = (x^{(i)}, y^{(i)}) : i = 1, ..., m$ η γεωμετρική απόσταση των (w, b) ως προς το S θα είναι η μικρότερη πιθανή απόσταση από καθεμία από τις ξεχωριστές αποστάσεις κάθε στοιχείου στο δείγμα:

$$\gamma = min_{i=1,..,m} \gamma^{(i)}$$

4.1.5 Βελτιστοποίηση

Μπορούμε λοιπόν να δούμε ότι μία λύση με με μεγάλη ακρίβεια και βαθμό βεβαιότητας είναι η μεγιστοποίηση της γεωμετρικής απόστασης. Σαν γεωμετρική εξήγηση, θέλουμε να βρούμε γραμμή όριο η οποία θα έχει το μεγαλύτερο δυνατό κενό από τα κοντινότερα σε εκείνη στοιχεία(και προφανώς θα τα διαχωρίζει).

Αρχικά θεωρούμε ότι δείγμα μας είναι γραμμικό, και συνεπώς είναι εφικτό να διαχωριστεί με μία γραμμή. Σκοπός μας είναι η λύση του προβλήματος βελτιστοποίησης:

$$max_{\gamma,w,b} \gamma$$

τέτοιο ώστε

$$y^{(i)}(w^T x^{(i)} + b) \geq \gamma, i = 1, ..., m, \|w\| = 1$$

Ουσιαστικά, θέλουμε να μεγιστοποιήσουμε το γ σε σχέση με κάθε στοιχείο στο δείγμα, του οποίου η απόσταση είναι τουλάχιστον γ. Αφού όλες οι γεωμετρικές αποστάσεις είναι τουλάχιστον γ, η λύση αυτού το προβλήματος βελτιστοποίησης θα μου δώσει το ζεύγος (w, b) που επιτυγχάνει τη μέγιστη δυνατή απόσταση. Το βασικό πρόβλημα που αντιμετωπίζουμε είναι η συνθήκη $\|w\| = 1$, η οποία υπάρχει για τη συσχέτιση των δύο αποστάσεων που ορίστηκαν στην προηγούμενη υποενότητα. Αν η λύση του προβλήματος ήταν εφικτή στη δεδομένη μορφή, η απόδειξη θα τελείωνε εδώ. Δυστυχώς όμως η επίλυση γίνεται αρκετά δύσκολη από τη συνθήκη $\|w\| = 1$, η οποία κάνει το πρόβλημα μη-κυρτό γραμμικό πρόβλημα. Όντας δύσκολο να ανιχνεύσουμε τη λύση βελτιστοποίησης ακόμα και με κάποιο υπολογιστικό πρόγραμμα, πρέπει να τροποποιήσουμε τη σχέση για να βρούμε λύση.

Αρχικά, μπορούμε να μετατρέψουμε το γ με βάση τον τύπο $\gamma = \frac{\hat{\gamma}}{\|w\|}$ ο οποίος σχετίζει τις δύο αποστάσεις (γεωμετρική, πρακτική) ως εξής (με $\hat{\gamma}$ συμβολίζουμε τη νέα απόσταση):

$$max_{\gamma,w,b} \frac{\hat{\gamma}}{\|w\|}$$

τέτοιο ώστε

$$y^{(i)}(w^T x^{(i)} + b) \geq \hat{\gamma}, i = 1, ..., m$$

Συνεπώς, το πρόβλημα τώρα είναι η μεγιστοποίηση της ποσότητας $\frac{\hat{\gamma}}{\|w\|}$ ως προς τις αποστάσεις που είναι τουλάχιστον $\hat{\gamma}$. Έχοντας πλέον απαλλαγεί από το πρόβλημα της συνθήκης $\|w\| = 1$ συνεχίζουμε, αλλά αρκετά γρήγορα συνειδητοποιούμε, ότι αντιμετωπίζομε ακριβώς το ίδιο πρόβλημα και αυτή τη φορά με το $\|w\|$ στον παρονομαστή της σχέσης που θέλουμε να μεγιστοποιήσουμε.

Αυτή τη φορά θα κάνουμε χρήση ενός συμπεράσματος που είδαμε παραπάνω. Κατά το συμπέρασμα αυτό είδαμε ότι η μεταβολή των (w, b) κλιμακωτά στη γεωμετρική απόσταση, δεν επηρεάζει καθόλου την απόσταση. Εισάγουμε λοιπόν αρχικά μία συνθήκη, ότι η πρακτική απόσταση των (w, b) ως προς δείγμα μας πρέπει να είναι ένα $(\hat{\gamma} = 1)$.

Οπότε, με το νέο περιορισμό, μπορούμε να μετατρέψουμε τη σχέση $\frac{\hat{\gamma}}{\|w\|}$ σε $\frac{1}{\|w\|}$. Χωρίς βλάβη της γενικότητας μπορώ να μετατρέψω την τελευταία σχέση σε $\frac{1}{2}\|w\|$ και τελικά, να καταλήξω στο πρόβλημα:

$$min_{w,b} \frac{1}{2}\|w\|$$

τέτοιο ώστε

$$y^{(i)}(w^T x^{(i)} + b) \geq \hat{\gamma}, i = 1, ..., m$$

Τελικά, μετατρέψαμε το πρόβλημα σε εύκολα επιλύσιμη μορφή από κάποιο πρόγραμμα επίλυσης κυρτών τετραγωνικών προβλημάτων με γραμμικές συνθήκες. Η λύση της κυρτής τετραγωνικής μορφής που προέκυψε στο τελικό βήμα, θα μας δώσει τον βέλτιστο ταξινομητή.

Ενώ η απόδειξη τελειώνει εδώ, μπορούμε να συνεχίσουμε και να επεκταθούμε στη δυϊκή μορφή Lagrange του προβλήματός μας. Έτσι θα μπορέσουμε να αναφερθούμε στους πυρήνες, οι οποίοι παίζουν καθοριστικό ρόλο στην επίλυση πολυδιάστατων προβλημάτων. Ακόμη θα βοηθήσει να επιλύσουμε ακόμη πιο αποδοτικά το παραπάνω πρόβλημα από τις μεθόδους του γραμμικού προγραμματισμού.

4.1.6 Χρήση Μεθόδου Lagrange

Αρχικά, αντιμετωπίζουμε το πρόβλημα από μαθηματικής πλευράς. Έστω ότι είναι της μορφής:

$$min_w f(w)$$

τέτοιο ώστε

$$h_i(w) = 0, i = 1, ..., m$$

Η μέθοδος Lagrange για την επίλυση του παραπάνω προβλήματος θα μας δώσει τη σχέση:

$$L(w, a, b) = f(w) + \sum_{i=1}^{I} b_i h_i(w) \tag{4.1}$$

όπου b_i οι πολλαπλασιαστές Lagrange. Έπειτα παίρνουμε τις μερικές παραγώγους της σχέσης 4.1 ίσες με μηδέν και λύνουμε ως προς w και b:

$$\frac{\partial L}{\partial w_i} = 0, \quad \frac{\partial L}{\partial b_i} = 0 \tag{4.2}$$

Στη συνέχεια θα ασχοληθούμε με τις ιδέες και τα αποτελέσματα που θα μας οδηγήσουν στη βελτιστοποίηση του αλγόριθμου κατηγοριοποίησης. Το πρόβλημά μας

σε μια πρώιμη μορφή είναι:

$$g_i(w) \leq 0, i = 1, .., k \tag{4.3}$$
$$h_i(w) = 0, i = 1, .., l \tag{4.4}$$

Ο γενικευμένος τύπος Lagrange λοιπόν θα είναι:

$$L(w, a, b) = f(w) + \sum_{i=1}^{k} a_i g_i(w) + \sum, b_i h_i(w) \tag{4.5}$$

όπου a_i και b_i πολλαπλασιαστές Lagrange. Θέτουμε $\theta(w) = max_{a,ba_i \geq 0} L(w, a, b)$ Σε περίπτωση που η παραπάνω ποσότητα λάβει w που παραβιάζει κάποιον από τους αρχικούς περιορισμούς τότε η σχέση θα γίνει:

$$\theta(w) = max_{a,b:a_i \geq 0} f(w) + \sum_{i=1}^{k} a_i g_i(w) + \sum_{i=1}^{l} b_i h_i(w)$$

Αν οι περιορισμοί ικανοποιούνται για κάποια τιμή του w τότε

$$\theta(w) = f(w)$$

Συνεπώς $\theta(w) = f(w)$ αν το w ικανοποιεί τις αρχικές συνθήκες, ενώ θα είναι ίσο με άπειρο σε κάθε άλλη περίπτωση. Άρα, το πρόβλημα μεταφράζεται ως:

$$min_w \theta(w) = min_w max_{a,b:a_i \geq 0} L(w, a, b)$$

Μέχρι στιγμής η διατύπωση δεν έχει ουσιαστικές αλλαγές. Στο σημείο αυτό θα εξετάσουμε μία διαφορετική προσέγγιση:

$$\theta_D(a, b) = min_w L(w, a, b)$$

Ο δείκτης D υπάρχει για να δείχνει τη διπλή συνθήκη . Έχοντας ορίσει το θ_0, τώρα θέλουμε να βρούμε:

$$max_{a,b:a_i \leq 0} \theta_d(a, b) = max_{a,b:a_i \leq 0} min_w L(w, a, b)$$

Το παρόν πρόβλημα διαφέρει από το αρχικό στο σημείο όπου τα min και max έχουν ανταλλάξει θέσεις. Ορίζουμε $d^* = max_{a,b:a_i \leq 0} \theta_D(w)$. Τότε θα ισχύει:

$$d^* = max_{a,b:a_i \leq 0} min_w L(w, a, b) \leq min_w max_{a,b:a_i \leq 0} L(w, a, b) = p^* \tag{4.6}$$

Υπό συγκεκριμένες συνθήκες μπορεί να προκύψει ισότητα ανάμεσα στα d^* και p^*. Οι συνθήκες αυτές είναι οι εξής:

Έστω f, g κυρτές συναρτήσεις και h_i γραμμική. Επίσης υποθέτουμε ότι οι περιορισμοί g_i είναι εφικτοί. Τότε θα υπάρχει w τέτοιο ώστε $g_i(w) < 0$ για κάθε i. Σύμφωνα με τα παραπάνω, θα πρέπει να υπάρχουν w^*, a^*, b^* τέτοια ώστε το w^* να είναι η λύση του αρχικού μας προβλήματος και τα a^*, b^* να είναι λύσεις του δυϊκού προβλήματος. Επιπροσθέτως, θα έχουμε $p^* = d^* = L(w^*, a^*, b^*)$ και τα w^*, a^*b^* θα ικανοποιούν τις συνθήκες Karush-Kuhn-Tucker (KKT) οι οποίες είναι:

$$\frac{\partial}{\partial w_i} L(w^*, a^*, b^*) = 0, \quad i = 1, .., n \tag{4.7}$$

$$\frac{\partial}{\partial b_i} L(w^*, a^*, b^*) = 0, \quad i = 1, ..., l \tag{4.8}$$

$$a_i^* g_i(w^*) = 0, \quad i = 1, ..., k \tag{4.9}$$

$$g_i(w^*) \leq 0, \quad i = 1, ..., k \tag{4.10}$$

$$a^* \geq 0, \quad i = 1, .., k \tag{4.11}$$

Επιπρόσθετα, αν κάποια από τα w^*, a^*, b^* ικανοποιούν τις συνθήκες KKT τότε αποτελούν λύση και για το αρχικό και για το δυικό πρόβλημα.

Η προσοχή μας θα στραφεί στη συνθήκη 4.9 η οποία είναι γνωστή και ως δυϊκή συμπληρωματική συνθήκη των KKT. Πιο συγκεκριμένα μας δείχνει ότι αν έχουμε $a_i^* > 0$, τότε $g_i(w^*) = 0$

4.1.7 Βελτιστοποίηση

Στην πέμπτη ενότητα καταλήξαμε στην εξής διατύπωση για το πρόβλημα:

$$min_{w,b} \frac{1}{2} \|w\|^2$$

τέτοιο ώστε

$$y_{(i)}(w^T x^{(i)} + b) \geq 1, i = 1, .., m$$

Οι περιορισμοί μπορούν να γραφούν επίσης με τη μορφή $g_i(w) = y^{(i)}(w^T x^{(i)} + b) + 1 \leq 0$ και υπάρχει ένας αντίστοιχος περιορισμός για κάθε στοιχείο στο δείγμα μας. Λόγω της συνθήκης διπλής συμπληρωματικότητας των (KKT), θα έχουμε $a_i > 0$ μόνο για τα στοιχεία που έχουν πρακτική απόσταση ακριβώς ίση με ένα. Έστω ότι έχουμε το παρακάτω σχήμα στο οποίο η συνεχής γραμμή είναι το βέλτιστο hyperplane. Τα σημεία με τη μικρότερη απόσταση είναι εκείνα που βρίσκονται κοντινότερα στη γραμμή

απόφασης. Στο σχήμα μας αναφερόμαστε στα τρία μαρκαρισμένα στοιχεία(ένα αρνητικό και δύο θετικά) που βρίσκονται πάνω στις διακεκομμένες γραμμές. Τα τρία αυτά σημεία $a_i, i = 1, 2, 3$ θα είναι τα μη μηδενικά σημεία τα οποία θα υπολογίσουμε στις ακόλουθες πράξεις και είναι οι support vectors του προβλήματός μας. Στη συνέχεια θα δούμε γιατί ο περιορισμός των στοιχείων που καλούμαστε να υπολογίσουμε με τη χρήση των support vectors είναι τόσο σημαντικός.

Συνεχίζοντας την απόδειξη, θα προσπαθήσουμε να εκφράσουμε τον αλγόριθμο μας σαν όρους εσωτερικού γινομένου ανάμεσα στα στοιχεία που του εισάγουμε. Το βήμα αυτό θα φανεί πολύ χρήσιμο κατά την ανάλυση της μεθόδου του πυρήνα. Η κατασκευή του τύπου Lagrange του προβλήματος ήταν:

$$L(w, a, b) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{m} a_i(y^{(i)}(w^T x^{(i)} + b) \tag{4.12}$$

Για να μεταφραστεί η παραπάνω σχέση σε διπλό πρόβλημα πρέπει πρώτα να ελαχιστοποιήσουμε το $L(w, a, b)$ ως προς w και b (για εάν δοσμένο a). Έτσι θα βρούμε το θ_D το οποίο προκύπτει εξισώνοντας τις μερικές παραγώγους του L, ως προς w και b, να είναι ίσες με μηδέν. :

$$w = \sum_{i=1}^{m} a_i y^{(i)} x^{(i)} \tag{4.13}$$

Και η μερική παράγωγος ως προς b είναι:

$$\frac{\partial}{\partial b} L(w, a, b) = \sum_{i=1}^{m} a_i y^{(i)} = 0 \tag{4.14}$$

Αν συνδυάσουμε τον ορισμό του w με τη σχέση 4.13 και το αποτέλεσμα με τη σχέση 4.12 προκύπτει:

$$L(w, a, b) = \sum_{i=1}^{m} a_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} a_i a_j (x^{(i)})^T x^{(j)} - b \sum_{i=1}^{m} a_i y^{(i)}$$

και λόγω της σχέσης 4.14 ο τελευταίος όρος είναι μηδέν, άρα:

$$L(w, a, b) = \sum_{i=1}^{m} a_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} a_i a_j (x^{(i)})^T x^{(j)}$$

Δεδομένου ότι οι συνθήκες KKT ισχύουν και ότι $p^* = d^*$ αρκεί να λύσουμε το νέο διπλό πρόβλημα αντί του αρχικού. Ουσιαστικά είναι ένα πρόβλημα μεγιστοποίησης

με παραμέτρους a_i. Θα ασχοληθούμε παρακάτω με την επίλυση του δ προβλήματος, αλλά σε περίπτωση που έχουμε τη λύση του, μπορούμε πολύ εύκολα από τη σχέση 4.13 να βρούμε τα βέλτιστα w με βάση τα a. Έπειτα έχοντας βρει το w^* μπορούμε να βρούμε άμεσα το b^* από τη σχέση:

$$b^* = -\frac{max_{i:y^{(i)}=-1} w^{*T} x^{(i)} + min_{i:y^{(i)}=1} w^{*T} x^{(i)}}{2} \tag{4.15}$$

Ας ασχοληθούμε λίγο παραπάνω με την εξίσωση 4.13. Υποθέτουμε ότι προσαρμόζουμε τις παραμέτρους του μοντέλου μας σε ένα σετ εκπαίδευσης και θέλουμε να προβλέψουμε το αποτέλεσμα σε μία νέα είσοδο x. Τότε θα έπρεπε να υπολογίσουμε την ποσότητα $w^T x + b$, και θα προβλέπαμε $y = 1$ αν και μόνο αν η ποσότητα ήταν μεγαλύτερη του μηδέν. Από την σχέση 4.13 η ποσότητα μπορεί να γραφεί:

$$w^T x + b = \left(\sum_{i=1}^{m} a_i y^{(i)} x^{(i)} \right)^T x + b \tag{4.16}$$

$$= \sum_{i=1}^{m} a_i y^{(i)} < x^{(i)}, x > + b \tag{4.17}$$

Συνεπώς, αν είχαμε βρει τα a_i, πριν κάνουμε πρόβλεψη, χρειάζεται να βρούμε μία ποσότητα που εξαρτάται μόνο από το εσωτερικό γινόμενο των x με τα στοιχεία στο δείγμα. Επιπλέον, γνωρίζουμε ότι πολλά από τα a_i θα είναι μηδενικά. Αυτή είναι η διευκόλυνση που δίνουν οι support vectors. Θα υπολογίσουμε λοιπόν τα εσωτερικά γινόμενα για αυτά τα μη μηδενικά στοιχεία με το x και έτσι θα προκύψει η πρόβλεψη μας.

Εξετάζοντας την διπλή μορφή βελτιστοποίησης του αρχικού προβλήματος, κερδίσαμε σημαντικές γνώσεις •για τη δομή του προβλήματος. Ακόμη καταφέραμε να εκφράσουμε ολόκληρο τον αλγόριθμο με όρους εσωτερικού γινομένου μεταξύ των στοιχείων εισόδου. Στην ακόλουθη ενότητα θα εκμεταλλευτούμε την ιδιότητα αυτή για να εφαρμόσουμε τη μέθοδο του πυρήνα. Ο αλγόριθμος που θα προκύψει θα είναι ικανός να εκπαιδευτεί και να μάθει αποδοτικά σε μεγαλύτερες διαστάσεις.

4.1.8 Μέθοδος Πυρήνα

Πριν συνεχίσουμε με το μαθηματικό κομμάτι αξίζει να αναφέρουμε κάποια πράγματα για τη μέθοδο του πυρήνα. Ο λόγος ύπαρξης της μεθόδου του πυρήνα μπορεί να συνεισφέρει σε περιπτώσεις μη-γραμμικής διαχωρισιμότητας. Σε περίπτωση που δεν μπορούμε με κάποιο υπερεπίπεδο να διαχωρίσουμε το δείγμα μας προκειμένου διευκολυνθεί η διαδικασία της κατηγοριοποίησης, τότε η μέθοδος του πυρήνα επιχειρεί να

διαχωρίσει το δείγμα. Στην πράξη η διαδικασία η οποία ακολουθεί είναι να απεικονίσει το δείγμα σε ένα χώρο μεγαλύτερης διάστασης, στον οποίο η διαχωρισιμότητα είναι εφικτή, και έτσι να πετύχει το επιθυμητό αποτέλεσμα.

Αρχικά πρέπει να διαχωρίσουμε κάποιες έννοιες. Θα αποκαλούμε αρχική είσοδο τα γνωρίσματα εισόδου ενός προβλήματος. Όταν αυτό αντιστοιχηθεί σε κάποιο νέο σύνολο ποσοτήτων που δίνουμε στον αλγόριθμο εκμάθησης θα τα ονομάζουμε χαρακτηριστικά εισόδου. Ακόμη, το f θα υποδηλώνει την αντιστοίχιση χαρακτηριστικών, η οποία αντιστοίχεί από τα γνωρίσματα στα χαρακτηριστικά. Αυτό γίνεται διότι μπορεί να θέλουμε να εφαρμόσουμε τον αλγόριθμο Μηχανών Διανυσμάτων Υποστήριξης (SVM) με κάποια χαρακτηριστικά $f(x)$ αντί των αρχικών γνωρισμάτων. Αυτό επιτυγχάνεται βρίσκοντας το $f(x)$ για κάθε x στα στοιχεία εισόδου.

Αφού είδαμε ότι ο αλγόριθμος μπορεί να έχει τη μορφή όρων εσωτερικού γινομένου $< x, z >$, μπορούμε να τα αντικαταστήσουμε με $< f(x), f(z) >$. Δεδομένης μιας αντιστοίχισης $f(x)$, ορίζουμε τον αντίστοιχο πυρήνα να είναι:

$$K(x, z) = f(x)^T f(z)$$

Έτσι όπου είχαμε πριν $< x, z >$ αντικαθιστούμε με $K(x, z)$ και τώρα ο αλγόριθμος μας μαθαίνει χρησιμοποιώντας τα χαρακτηριστικά f.

Έπειτα, δεδομένου f μπορούμε να βρούμε το $K(x, z)$ υπολογίζοντας το $f(x)$ και το $f(z)$ και έπειτα το εσωτερικό τους γινόμενο. Όμως, πολύ ενδιαφέρουσα είναι η παρατήρηση ότι πολλές φορές το $K(x, z)$ μπορεί να είναι υπερβολικά χρονοβόρο να υπολογιστεί, όπως και το $f(x)$ (πιθανότατα διότι πρόκειται για έναν πολυδιάστατο πίνακα). Έτσι, εισάγοντας στον αλγόριθμο μας έναν αποτελεσματικό τρόπο να υπολογίζει το $K(x, z)$ μπορούμε να τον κάνουμε να επιτυγχάνει μάθηση σε μεγαλύτερες διαστάσεις, χωρίς να χρειαστεί ποτέ να υπολογίσει αυτές τις ποσότητες.

Ας δούμε ένα παράδειγμα. Έστω $x, z \in R^n$ και:

$$K(x, z) = (x^T z)^2$$

Μπορούμε επίσης να το γράψουμε σαν:

$$K(x, z) = \left(\sum_{i=1}^{n} x_i z_i\right)\left(\sum_{j=1}^{n} x_j z_j\right) \tag{4.18}$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{n} x_i z_i x_j z_j \tag{4.19}$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{n} (x_i x_j)(z_i z_j) \tag{4.20}$$

Άρα μπορούμε να δούμε ότι το $K(x,z) = f(x)^T f(z)$, όπου η συνάρτηση χαρακτηριστικών f προκύπτει (για παράδειγμα στην περίπτωση που $n = 3$) από:

$$\begin{bmatrix} x_1x_1 \\ x_1x_2 \\ x_1x_3 \\ x_2x_1 \\ x_2x_2 \\ x_2x_3 \\ x_3x_1 \\ x_3x_2 \\ x_3x_3 \end{bmatrix}$$

Αξίζει να σημειωθεί ότι αν ο υπολογισμός της υψηλών διαστάσεων $f(x)$ απαιτούσε χρόνο $O(n^2)$, ο υπολογισμός του $K(x,z)$ θα χρειαστεί μόλις $O(n)$ χρόνο.
Για έναν σχετικό πυρήνα, θα έχω:

$$\begin{aligned} K(x,z) &= (x^T z + c)^2 \\ &= \sum_{i,j=1}^{n} (x_i x_j)(z_i z_j) + \sum_{i,j=1}^{n} (\sqrt{2c}x_i)(\sqrt{2c}z_i) + c^2 \end{aligned}$$

Με αντίστοιχο πίνακα χαρακτηριστικών:

$$\begin{bmatrix} x_1x_1 \\ x_1x_2 \\ x_1x_3 \\ x_2x_1 \\ x_2x_2 \\ x_2x_3 \\ x_3x_1 \\ x_3x_2 \\ x_3x_3 \\ \sqrt{2c}x_1 \\ \sqrt{2c}x_2 \\ \sqrt{2c}x_3 \\ c \end{bmatrix}$$

όπου η παράμετρος c ελέγχει τα σχετικά βάρη μεταξύ των όρων x_i (πρώτης τάξης) και x_ix_j (δεύτερης τάξης).
Γενικά ο πυρήνας $K(x,z) = (x^T z + c)$ αντιστοιχεί σε μία αντιστοίχιση χαρακτηριστικών σε $\binom{n+d}{d}$ χώρο χαρακτηριστικών, που αντιστοιχεί στα μονώνυμα της μορφής

$x_{i1}, x_{i2}, ..., x_{ik}$ έως τάξης d. Ωστόσο, παρά το γεγονός ότι εργαζόμαστε σε αυτόν τον $O(n^d)$ διαστάσεων χώρο, ο υπολογισμός $K(x, z)$ εξακολουθεί να παίρνει μόνο $O(n)$ χρόνο και επομένως δεν χρειάζεται να εκφράσουμε τα διανύσματα χαρακτηριστικών σε αυτόν τον πολύ υψηλών διαστάσεων χώρο.

Τώρα, ας δούμε μία λίγο πιο διαφορετική σκοπιά του πυρήνα. Διαισθητικά, αν τα $f(x)$ και $f(z)$ βρίσκονται κοντά το ένα στο άλλο, τότε θα περιμένουμε το $K(x, z) = f(x)^T f(z)$ να είναι μεγάλο. Αντίστοιχα για $f(x)$, $f(z)$ απομακρυσμένα το ένα από το άλλο θα πρέπει η αντίστοιχη ποσότητα $K(x, z)$ να είναι μικρή. Συνεπώς μπορούμε να δούμε το $K(x, z)$ σαν ένα μέτρο σύγκρισης του πόσο κοντινά είναι τα $f(x)$ και $f(z)$, ή ακόμη και τα z, z.
Δεδομένου του παραπάνω συλλογισμού, έστω ότι σε ένα τυχαίο πρόβλημα εκμάθησης έχετε καταλήξει στην παρακάτω εξίσωση για την σύγκριση των x, z:

$$K(x, z) = e^{\left(-\frac{\|x - z\|^2}{2s^2}\right)}$$

Αποτελεί μία λογική συνάρτηση σύγκρισης μιας και το αποτέλεσμα πλησιάζει στο ένα αν τα x και z βρίσκονται κοντά, ενώ πλησιάζει το μηδέν αν τα x και z είναι μακριά. Μπορεί όμως αυτός ο ορισμός του K να χρησιμοποιηθεί σαν πυρήνας σε αλγόριθμο Μηχανών Διανυσμάτων Υποστήριξης (SVM). Για τη συγκεκριμένη σχέση η απάντηση είναι θετική και μάλιστα πρόκειται για τον πυρήνα Gauss. Όμως με ποιο τρόπο μπορούμε γενικά να αποφανθούμε αν ένας πυρήνας είναι κατάλληλος·
Υποθέτουμε αρχικά ότι ο K είναι κατάλληλος πυρήνας που αντιστοιχεί σε αντιστοίχιση χαρακτηριστικών f. Έστω ένα πεπερασμένο πλήθος m σημείων $x^1, x^2, ..., x^m$, και έστω ένας τετραγωνικός $m_x m$ πίνακας K τέτοιος ώστε η είσοδος (i, j) υπολογίζεται από τον τύπο $K_{ij} = K(x^i, x^j)$. Ο πίνακας αυτός καλείται πίνακας πυρήνα. Η απόφαση να ονομάσουμε K και τον πίνακα και την συνάρτηση δεν αποτελεί πιθανό λάθος, αλλά είναι σκόπιμο λόγω της κοντινής τους σχέσης.

Τώρα, έχοντας K έναν αποδεκτό πυρήνα, $K_{ij} = K(x^i, x^j) = f(x^i)^T f(x^j) = f(x^i) f(x^j)^T = K(x^j, x^i) = K_{ji}$, και συνεπώς το K πρέπει να είναι συμμετρικό. Έστω $f_k(x)$ να είναι η κ-οστή συντεταγμένη του διανύσματος $f(x)$. Για κάποιο διάνυσμα z έχουμε:

$$z^T K z = \sum_i \sum_j z_i K_{ij} z_j \tag{4.21}$$

$$= \sum_i \sum_j z_i f(x^i)^T f(x^j) z_j \tag{4.22}$$

$$= \sum_i \sum_j z_i \sum_k f_k(x^i) f_k(x^j) z_j \tag{4.23}$$

$$= \sum_k \sum_i \sum_j z_i f_k(x^i) f_k(x^j) z_j \tag{4.24}$$

$$= \sum_k \left(\sum_i z_i f_k(x^i) \right)^2 \tag{4.25}$$

Συνεπώς, αφού το z είναι αυθαίρετο, το K θα είναι θετικό ($K \geq 0$).

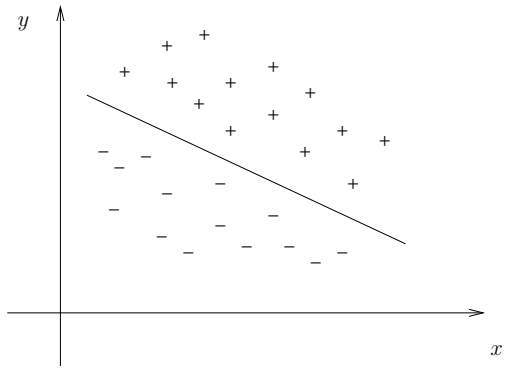
Μέχρι στιγμής έχουμε δείξει ότι εάν το K είναι ένας έγκυρος πυρήνας, τότε ο αντίστοιχος πίνακας πυρήνα $K \in R^{mxm}$ είναι συμμετρικός και θετικά ορισμένος. Αυτό αποδεικνύεται όχι μόνο απαραίτητη, αλλά και επαρκής προϋπόθεση προκειμένου το K να είναι ένας έγκυρος πυρήνας. Ο πυρήνας αυτός ονομάζεται και πυρήνας Mercer.

Θεώρημα (Mercer) 1 *Έστω $K : R^n x R^n \to R$. Τότε προκειμένου το K να είναι έγκυρος πυρήνας Mercer είναι αναγκαίο και επαρκές ότι για κάθε $x_1, x_2, ..., x_m : m < \infty$, ο αντίστοιχος πίνακας πυρήνα είναι συμμετρικός και θετικά ορισμένος.*

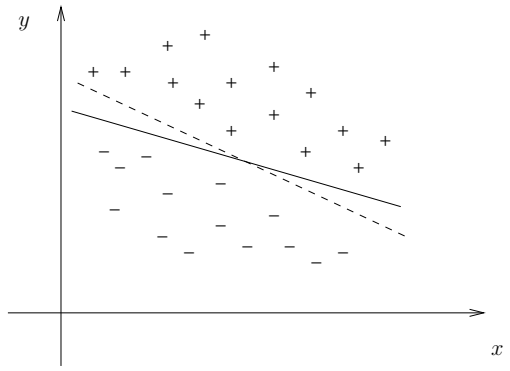
4.1.9 Κανονικοποίηση Και Περίπτωση Μη-Διαχωρισιμότητας

Οι παραλλαγές των Μηχανών Διανυσμάτων Υποστήριξης (SVM) που έχουμε συναντήσει ως τώρα είχαν ως δεδομένο ότι το δείγμα είναι γραμμικά διαχωρίσιμο. Ενώ είναι πολύ πιθανό η αντιστοίχιση των στοιχείων σε έναν υψηλής διάστασης χώρο, διαμέσου της f, να μας δώσει ένα αποτέλεσμα που διαχωρίζεται δεν μπορούμε να το εγγυηθούμε. Επίσης, σε ορισμένες περιπτώσεις δεν είναι ξεκάθαρο ότι η εύρεση διαχωριστικού υπερεπιπέδου είναι ακριβώς αυτό που θέλουμε να κάνουμε, καθώς αυτό μπορεί να είναι αναξιόπιστο σε υπερβολικά υψηλά επίπεδα. Για παράδειγμα, το σχήμα 4.5 δείχνει ένα βέλτιστο όριο απόφασης, και αν προσθέσουμε ένα μόνο στοιχείο στο δείγμα μας, στην πάνω αριστερή περιοχή εικόνα 4.6, προκαλεί μια δραματική μεταβολή στο όριο απόφασης και ο νέος ταξινομητής έχει πολύ μικρότερο περιθώριο.

Για να μπορέσει ο αλγόριθμος να επεξεργαστεί δεδομένα που δεν διαχωρίζονται γραμμικά, και να αποδίδει καλύτερα σε υψηλές διαστάσεις, μετατρέπουμε το υπάρχον



Σχήμα 4.5: Διαχωρισμός 1



Σχήμα 4.6: Διαχωρισμός 2

πρόβλημα ως εξής:

$$min_{g,w,b}\frac{1}{2}\|w\|^2 + c\sum_{i=1}^{m} g_i$$

τέτοια ώστε:

$$y^i(w^T x^i + b) \geq 1 - g_i, i = 1, ..., m$$

$$g_i \leq 0, i = 1, ..., m$$

Με την αλλαγή αυτή το λειτουργικό περιθώριο μπορεί πια να λάβει τιμές μικρότερες του ένα. Η παράμετρος c ελέγχει τα σχετικά βάρη ανάμεσα στους δύο μας στόχους που είναι η ελαχιστοποίηση του $\|w\|^2$ (το οποίο είδαμε ότι κάνει το περιθώριο μεγαλύτερο), και να εξασφαλίσει ότι τα περισσότερα στοιχεία στο δείγμα έχουν λειτουργικό περιθώριο τουλάχιστον ίσο με ένα. Όμοια με πριν, θα προκύψει και ο τύπος του Lagrange:

$$L(w, b, g, a, r) = \frac{1}{2}w^T w + c\sum_{i=1}^{m} g_i - \sum_{i=1}^{m} a_i[y^{(i)}(x^T w + b) - 1 + g_i] - \sum_{i=1}^{m} r_i g_i$$

Εδώ, τα a_i και r_i είναι οι πολλαπλασιαστές Lagrange (οι οποίοι είναι μεγαλύτεροι η ίσοι με το μηδέν). Όμοια με πριν, αν παραγωγίσουμε ως προς w και b αντικαταστήσουμε και απλοποιήσουμε τη σχέση μας, θα προκύψει η διατύπωση:

$$max_a W(a) = \sum_{i=1}^{m} a_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} a_i a_j < x^{(i)}, x^{(j)} >$$

τέτοια ώστε:

$$0 \leq a_i \leq c, i = 1, ..., m$$

$$\sum_{i=1}^{m} a_i y^{(i)} = 0$$

Όπως αναφέραμε και παραπάνω τα w μπορούν να εκφραστούν με βάση τα a_i όπως στη σχέση $w = \sum_{i=1}^{m} a_i y^{(i)} x^{(i)}$. Μετά τη λύση τους διπλού προβλήματος μπορούμε να χρησιμοποιήσουμε τη σχέση $w^T x + b = \sum_{i=1}^{m} a_i y^{(i)} < x^{(i)}, x > +b$ για να εξάγουμε τις προβλέψεις μας. Η μόνη ουσιαστική αλλαγή που προέκυψε από την τροποποίηση του διπλού προβλήματος είναι ότι η συνθήκη $a_i \leq 0$ έγινε $0 \leq a_i \leq c$. Ακόμη πρέπει να τροποποιηθεί ο υπολογισμός του b^*. Τέλος οι συνθήκες KKT θα γίνουν:

$$a_i = 0 \Rightarrow y^{(i)}(w^T x^{(i)} + b) \geq 1$$

$$a_i = c \Rightarrow y^{(i)}(w^T x^{(i)} + b) \leq 1$$

$$0 < a_i < c \Rightarrow y^{(i)}(w^T x^{(i)} + b) = 1$$

Τέλος, μένει να βρούμε έναν αλγόριθμο ικανό να λύσει αυτό το πρόβλημα. Έχοντας καταλήξει λοιπόν σε ένα δυικό πρόβλημα βελτιστοποίησης, υπάρχουν διάφοροι αλγόριθμοι για την επίλυση του. Ένας εξ αυτών είναι και ο αλγόριθμος Sequential Minimal Optimization (SMO). Η επίλυση από αυτό το σημείο και έπειτα είναι καθαρά υπολογιστικό ζήτημα, συνεπώς η απόδειξη τελειώνει εδώ.

Τέλος, θα αναφερθούμε σε κάποια ιστορικά στοιχεία σύμφωνα με την εξέλιξη του αλγορίθμου. Η αρχική ιδέα των Μηχανών Υποστήριξης Διανυσμάτων (SVM) επινοήθηκε από τους Vladimir N. Vapnik και Alexey Ya. Chervonenkis το έτος 1963. Το έτος 1992 οι Bernhard E. Boser, Isabelle M. Guyon και Vladimir N. Vapnik πρότειναν έναν τρόπο δημιουργίας μη γραμμικών ταξινομητών εφαρμόζοντας το τέχνασμα του πυρήνα σε υπερβολικά περιθώρια μέγιστου περιθωρίου. Το τελικό στάδιο στο κομμάτι εξέλιξης των Μηχανών Υποστήριξης Διανυσμάτων (SVM)

προέκυψε έπειτα από την πρόταση των Corinna Cortes και Vapnik το 1993 η οποία εισήγαγε τα soft margins και η τελική δημοσίευση έγινε το 1995.

4.2 Αλγόριθμος Κ-Πλησιέστερων Γειτόνων

Ο αλγόριθμος k-πλησιέστερων γειτόνων (k-NN) είναι μια μη παραμετρική μέθοδος που χρησιμοποιείται για ταξινόμηση και παλινδρόμηση. Και στις δύο περιπτώσεις, η είσοδος αποτελείται από τα πλησιέστερα παραδείγματα εκπαίδευσης στο χώρο των χαρακτηριστικών. Η έξοδος εξαρτάται από το εάν ο αλγόριθμος χρησιμοποιείται για ταξινόμηση ή παλινδρόμηση:

- Στην ταξινόμηση η έξοδος είναι μέλος κάποιας τάξης. Ένα αντικείμενο ταξινομείται από μια πλειονότητα των ψήφων των γειτόνων του, με το αντικείμενο να ανατίθεται στην τάξη που είναι πιο συνηθισμένη στους k πλησιέστερους γείτονές του (όπου k είναι ένας θετικός ακέραιος, συνήθως μικρός). Σε περίπτωση που $k = 1$, τότε το αντικείμενο απλώς αποδίδεται στην κλάση του συγκεκριμένου πλησιέστερου γείτονα.

- Στην παλινδρόμηση, η έξοδος είναι η τιμή ιδιότητας για το αντικείμενο. Αυτή η τιμή είναι ο μέσος όρος των τιμών των πλησιέστερων γειτόνων.

Οι γείτονες λαμβάνονται από ένα σύνολο αντικειμένων για τα οποία είναι γνωστή η τάξη (για την ταξινόμηση) ή η τιμή ιδιότητας αντικειμένου (για παλινδρόμηση). Αυτό μπορεί να θεωρηθεί ως το δείγμα εκπαίδευσης για τον αλγόριθμο, αν και δεν απαιτείται ρητό εκπαιδευτικό βήμα.

Τόσο για την ταξινόμηση όσο και για την παλινδρόμηση, μια χρήσιμη τεχνική μπορεί να χρησιμοποιηθεί για να αποδώσει το βάρος στις συνεισφορές των γειτόνων, έτσι ώστε οι πλησιέστεροι γείτονες να συμβάλλουν περισσότερο στον μέσο όρο από τους πιο μακρινούς. Για παράδειγμα, ένα κοινό μέσο βάρος μπορεί να αποδοθεί σε κάθε γείτονα με την τιμή $\frac{1}{d}$, όπου d είναι η απόσταση από τον γείτονα.

Το k-NN είναι ένας τύπος μάθησης βασισμένου σε παραδείγματα, όπου η λειτουργία προσεγγίζεται μόνο τοπικά και όλος ο υπολογισμός αναβάλλεται μέχρι την ταξινόμηση. Αποτελεί ένας από τους απλούστερους αλγορίθμους του συνόλου των αλγορίθμων μηχανικής μάθησης.

Κεφάλαιο 5

Εφαρμογή: Κατηγοριοποίηση Χειρόγραφων Μουσικών Συμβόλων

5.1 Εισαγωγή

Στο κεφάλαιο αυτό θα ασχοληθούμε με μια πρακτική εφαρμογή κάποιων από τους προαναφερθέν τρόπους. Θα παρουσιάσουμε ένα πρόγραμμα (Python Script, Κεφάλαιο 5.5 Κώδικας) που αναγνωρίζει νότες δύο κατηγοριών. Αρχικά το script που θα επεξεργαστούμε έχει δημιουργηθεί μαζί με το πακέτο Muscima++. Πληροφορίες για τους συγγραφείς μπορούν να βρεθούν , [8], [9]. Τέλος θα γίνει ανάλυση και σύγκριση των αποτελεσμάτων ανάλογα με τις διαφορετικές μεθόδους τις οποίες θα χρησιμοποιήσουμε.

5.2 Περιγραφή του Προγράμματος

Η βάση Muscima++ είναι ένα δείγμα από χειρόγραφα μουσικά σύμβολα. Αποτελείται από (ενενήντα ένα χιλιάδες διακόσια πενήντα πέντε 91255) σύμβολα. Περιέχει τόσο βασικά σύμβολα, όσο και εξειδικευμένα όπως κλειδιά ή στίξης διαφοροποίησης αξίας. Υπάρχουν (εικοσιτρείς χιλιάδες τριακόσια πενήντα δύο 23352) νότες, (εικοσιμία χιλιάδες τριακόσια πενήντα έξι 21356) από τις οποίες έχουν γεμάτη κεφαλή, (χίλιες εξακόσιες σαράντα οχτώ 1648) έχουν κενή κεφαλή, και (τριακόσιες σαράντα οχτώ 348) είναι ποικιλματικές νότες. Το Muscima++ έχει δύο μέρη, το εργαλείο Muscima Marker το οποίο έχει ως μοναδικό ρόλο την εισαγωγή νέων στοιχείων στο δείγμα και το muscima που είναι μια διεπαφή εισόδων και εξόδων ικανή να εκπαιδευτεί και να αναγνωρίσει στοιχεία του δείγματος. Συνεχίζουμε με μία γενική περίληψη της λειτουργία του κώδικα του κύριου σεναρίου και έπειτα θα δώσουμε έμφαση στα

σημεία που αναφέρθηκαν νωρίτερα στη θεωρία.

5.2.1 Εισαγωγή Δείγματος

Αρχικά ξεκινάμε εισάγοντας την βιβλιοθήκη os η οποία είναι υπεύθυνη για την ομαλή λειτουργία του script σε διάφορα λειτουργικά συστήματα (windows,unix-linux,macos). Έπειτα εισάγουμε την λίστα parse cropobject list η οποία αποτελείται από στοιχεία, σε μορφή XML, τα οποία έχουν αφαιρεθεί από τις αρχικές νότες με τη μέθοδο parse [1] με σκοπό να μελετηθούν στα κομμάτια που παρουσιάζουν διαφορές κάνοντας τη διαδικασία της ομαδοποίηση και του διαχωρισμού τους πολύ ευκολότερη. Στη συνέχεια αφού ανοίξουμε τον κατάλογο με τα αρχεία μας, διαβάζουμε τα ονόματα τους και δημιουργώ μία λίστα με όνομα croopobjects fnames. Τέλος, εφαρμόζουμε τη μέθοδο parse σε καθένα από τα στοιχεία της λίστας που μόλις δημιουργήσαμε, φτιάχνοντας την τελική μορφή με την οποία θα εισαχθούν τα δεδομένα μας στη λίστα docs. Συνεχίζουμε λοιπόν με την επεξεργασία του δείγματος

5.2.2 Επεξεργασία Δείγματος

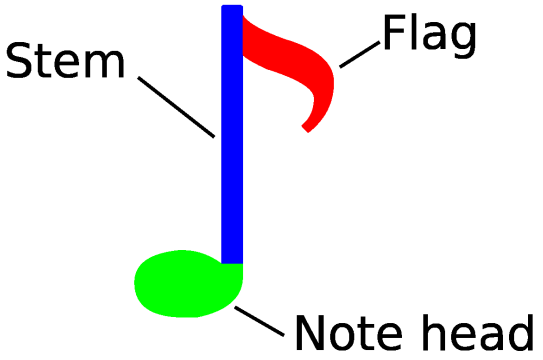
Το επόμενο κομμάτι λοιπόν είναι η κατηγοριοποίηση των αρχείων που έχουμε εισάγει στη λίστα docs. Αρχικά δημιουργούμε μία μέθοδο με όνομα extract notes from doc που παίρνει σαν παράμετρο στοιχεία της λίστας cropobjects και επιστρέφει δύο λίστες που περιέχουν cropobject tuples[2] μία με τέταρτα και μία με μισά(αξίες νοτών). Το επόμενο βήμα είναι να πάρουμε κάθε στοιχείο στη λίστα cropobjects με τη σειρά και να το περάσουμε από τους ακόλουθους ελέγχους (τα διάφορα κομμάτια μίας νότας φαίνονται στο σχήμα 5.1):

- Αν η νότα έχει stem
- Αν η νότα έχει beam
- Αν η νότα τελειώνει με flag

Ένας επιπλέον έλεγχος γίνεται για να σιγουρευτούμε ότι δεν έχει γίνει λάθος με τα τέταρτα. Μπορεί να προκληθεί κάποιο πρόβλημα δεδομένου του ότι έχουν κενό στην κεφαλή και κάποιες φορές είναι πάνω σε γραμμή πενταγράμμου. Οπότε ο έλεγχος επιβεβαιώνει ότι οι γραμμές που υπάρχουν εντός του κενού της νότας είναι λόγω πενταγράμμου.

[1] Η μέθοδος parse συνήθως χρησιμοποιείται για την ανάλυση ενός κειμένου. Στο συγκεκριμένο πρόβλημα εμείς τη χρησιμοποιούμε προκειμένου να πάρουμε από κάθε νότα ένα κομμάτι συγκεκριμένης θέσεις και διαστάσεων κάθε φορά.

[2] tuple είναι μία συλλογή που είναι ταξινομημένη και δεν αλλάζει



Σχήμα 5.1: Μέρη Νότας

Τελικά χωρίζουμε τις νότες σε μισά σε περίπτωση που έχουν άδεια κεφαλή νότας ή τέταρτα σε περίπτωση που δεν έχουν τίποτα μέσα στην κεφαλή. Εδώ τελειώνει η διαδικασία εξαγωγής των νοτών και επιστρέφουμε τις τελικές λίστες τέταρτα,μισά.

5.2.3 Τελική Επεξεργασία

Συνεχίζουμε με την αποθήκευση των νοτών από τις δύο νέες λίστες που δημιουργήσαμε, στις λίστες με όνομα qns,hns αντίστοιχα και επιστρέφουμε και τυπώνουμε το μέγεθος και των δύο λιστών(δηλαδή πόσες νότες έχει η καθεμία από αυτές). Έπειτα εισάγουμε τη βιβλιοθήκη numpy της python, η οποία είναι μία βιβλιοθήκη για επεξεργασία πολύπλοκων πινάκων, προκειμένου να φτιάξουμε τις εικόνες κάθε νότας με τον παρακάτω τρόπο. Πρώτα φτιάχνουμε ένα αρχικό κουτί μέσα στο οποίο χωρούν όλα τα αντικείμενα. Στη συνέχεια φτιάχνουμε τον καμβά πάνω στον οποίο θα βάλουμε τις μάσκες για τις δύο κατηγορίες που έχουμε και θα συγκρίνουμε τα αντικείμενα σε κάθε λίστα προκειμένου να καταλήξουμε με τις νέες λίστες qn images καιhn images οι οποίες περιέχουν τις εικόνες που έχουν προκύψει από τις νότες σε κάθε λίστα.

Το επόμενο βήμα είναι να κάνουμε αλλαγή στις διαστάσεις των εικόνων σε 40×10 pixel. Έπειτα ξανακάνουμε έναν έλεγχο για τυχόν αλλαγές λόγω της αλλαγής διαστάσεων, και αν υπάρχουν νέα κενά πεδία που τώρα θα έπρεπε να είναι μέρος της μάσκας, το διορθώνουμε. Έπειτα, σειρά έχει η δημιουργία του δείγματος εκπαίδευσης το οποίο ξεκινάει διαλέγοντας ένα μέρος των τετάρτων και δείχνοντας στο script την κατηγορία στην οποία ανήκουν. Όμοια γίνεται και η διαδικασία και για τα μισά. Με το τέλος της διαδικασίας εκπαίδευσης ο κώδικας μας είναι θεωρητικά έτοιμος να επεξεργαστεί το δείγμα εξέτασης και να το κατατάξει ανάλογα.

5.2.4 Μέθοδοι Κατηγοριοποίησης

Έχοντας όλα τα στοιχεία έτοιμα λοιπόν σειρά έχει η κατηγοριοποίηση. Στη συγκεκριμένη εφαρμογή δοκιμάσαμε τη διαδικασία αυτή με τρεις μεθόδους. Η πρώτη είναι αυτή του αλγορίθμου κοντινότερων γειτόνων. Η περιγραφή του τρόπου λειτουργία του αλγορίθμου βρίσκεται στο κεφάλαιο 4. Είναι μία μέθοδος που χρησιμοποιείται σχετικά συχνά σε αντίστοιχα προβλήματα και ήταν και η προτεινόμενου από τους δημιουργούς του συγκεκριμένου script. Στην αναζήτηση μας για μεγαλύτερη ακρίβεια δοκιμάσαμε την μέθοδο των απλών γραμμικών Μηχανών Διανυσμάτων Υποστήριξης (SVM) τα οποία επίσης έχουν αναλυθεί στο κεφάλαιο 4. Η τρίτη και τελευταία μέθοδος είναι πάλι στην κατηγορία των Μηχανών Διανυσμάτων Υποστήριξης (SVM) αλλά αυτή τη φορά δοκιμάσαμε με τη μέθοδο του πυρήνα. Συγκεκριμένα έγινε χρήση της μεθόδου πυρήνα RBF η οποία όπως θα φανεί παρακάτω ήταν η επιλογή με την μεγαλύτερη ακρίβεια.

5.3 Αποτελέσματα Και Σύγκριση

Καθεμία από τις μεθόδους που χρησιμοποιήθηκαν για την κατηγοριοποίηση δουλεύει με διαφορετικό τρόπο. Η επιλογή της μεθόδου κοντινότερου γείτονα είναι μια συνηθισμένη επιλογή για προβλήματα κατηγοριοποίησης με δύο κατηγορίες. Ανάλογα την κατανομή του δείγματος μπορεί να αποφέρει αρκετά υψηλά αποτελέσματα. Στην περίπτωση μας μπορεί να πετυχαίνει ένα αρκετά μεγάλο ποσοστό στην αναγνώριση των μισών, αλλά όπως μπορούμε να δούμε αυτό επηρεάζει την αναγνώριση της δεύτερης κατηγορίας. Αυτό ίσως να οφείλεται και στη φύση του αλγορίθμου, σύμφωνα με την οποία αν πολλά τέταρτα βρίσκονται αρκετά κοντά σε ομάδες που απαρτίζονται από μισά μπορεί να καταλήξουν στη λάθος κατηγορία. Συνεπώς, έχοντας μία αρκετά μεγάλη διαφορά στα ποσοστά επιτυχίας των δύο κατηγοριών τα αποτελέσματα κρίνονται ικανοποιητικά αλλά όχι ιδανικά. Οι συμβολισμοί στους επόμενους πίνακες είναι αρχικά τα half, quarter, τα οποία είναι οι δύο κατηγορίες από νότες, μισά και τέταρτα αντίστοιχα. Ακόμη έχουμε τα micro avg, macro avg και weighted avg τα οποία αντίστοιχα υπολογίζουν τις μετρήσεις, είτε υπολογίζοντας το σύνολο των σωστών προβλέψεων της μίας κατηγορίας και των λανθασμένων της άλλης, είτε υπολογίζοντας τις μετρήσεις για κάθε ετικέτα και βρίσκοντας τον μη σταθμισμένο μέσο όρο, είτε βρίσκοντας τον σταθμισμένο μέσο όρο. Αυτές οι ποσότητες δίνουν σαν αποτέλεσμα ποσοστά επιτυχίας. Στη συνέχεια τα precision, recall αντίστοιχα είναι η ακρίβεια που είναι ένα μέτρο της σχετικότητας των αποτελεσμάτων, και η ανάκληση που είναι ένα μέτρο του πόσα πραγματικά αληθή αποτελέσματα επιστρέφονται. Επίσης το f1-score είναι ο σταθμισμένος μέσος όρος της ακρίβειας και της ανάκλησης. Τέλος, support είναι ο αριθμός των εμφανίσεων κάθε κατηγορίας αντίστοιχα. Τα αποτελέσματα του αλγορίθμου μπορούν να φανούν στο παρακάτω σχήμα:

Πίνακας 5.1: Κοντινότερος Γείτονας

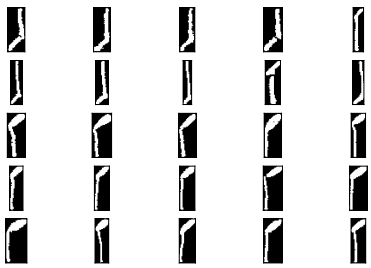
	precision	recall	f1-score	support
half	0.98	0.86	0.92	296
quarter	0.87	0.98	0.93	295
micro avg	0.92	0.92	0.92	591
macro avg	0.93	0.92	0.92	591
weighted avg	0.93	0.92	0.92	591

Δοκιμάζοντας στη συνέχεια τις Μηχανές Διανυσμάτων Υποστήριξης (SVM) τα αποτελέσματα παρουσιάζουν διαφορά. Υπάρχει μία μείωση στην κατηγορία των μισών, αλλά μια αρκετά σημαντική αύξηση στην κατηγορία των τετάρτων. Μπορεί αυτή η μέθοδος να μην πλησιάζει το ποσοστό 98% που παρουσιάστηκε στην πρώτη μέθοδο, αλλά παρουσιάζει πολύ μεγαλύτερη σταθερότητα. Μπορούμε λοιπόν να δούμε ότι η μέθοδος των Μηχανών Διανυσμάτων Υποστήριξης (SVM) είναι μέχρι στιγμής καλύτερη σαν σύνολο, αλλά κάτι εμποδίζει τον αλγόριθμο από το να δώσει υψηλότερα αποτελέσματα. Ίσως αυτό να συμβαίνει διότι το δείγμα δεν είναι εύκολα γραμμικά διαχωρίσιμο. Συνεπώς ο αλγόριθμος χωρίς τη μέθοδο πυρήνα του πυρήνα ίσως να αντιμετωπίζει κάποια μικρά προβλήματα. Τα αποτελέσματα της μεθόδου Μηχανών Διανυσμάτων Υποστήριξης (SVM) φαίνονται εδώ:

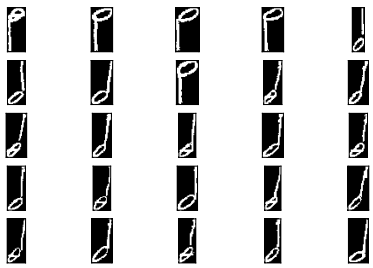
Πίνακας 5.2: Support Vector Machines

	precision	recall	f1-score	support
half	0.92	0.92	0.92	355
quarter	0.92	0.92	0.92	354
micro avg	0.92	0.92	0.92	709
macro avg	0.93	0.92	0.92	709
weighted avg	0.93	0.92	0.92	709

Η τρίτη μας μέθοδος λοιπό είναι Μηχανές Διανυσμάτων Υποστήριξης (SVM) με πυρήνα RBF. Τα αποτελέσματα είναι εμφανώς αυξημένα από την προηγούμενη δοκιμή μας, η οποία ήταν χωρίς πυρήνα, και μάλιστα έχουν διατηρήσει τη σταθερότητα τους. Η διαφορά αυτή ανάμεσα στις δύο αυτές μεθόδους προκύπτει λόγω της εφαρμογής του συγκεκριμένου πυρήνα. Η δυνατότητα επεξεργασίας δεδομένων, έως και αρκετά μεγαλύτερων των δύο διαστάσεων, που προσφέρει ο συγκεκριμένος πυρήνας είναι αρκετή για να ξεπεράσει σε ακρίβεια τους δύο άλλους αλγόριθμους, διατηρώντας σε κάθε περίπτωση σταθερά τα αποτελέσματα. Τέλος τα αποτελέσματα των Μηχανών



Σχήμα 5.2: Νότες αξίας τετάρτου



Σχήμα 5.3: Νότες μισής αξίας

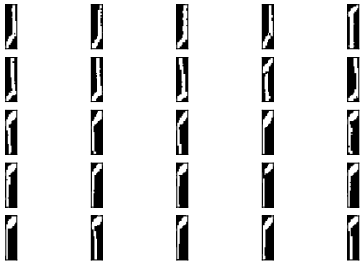
Διανυσμάτων Υποστήριξης (SVM) με πυρήνα RBF μπορούμε να τα δούμε εδώ:

Πίνακας 5.3: Support Vector Machines RBF Kernel

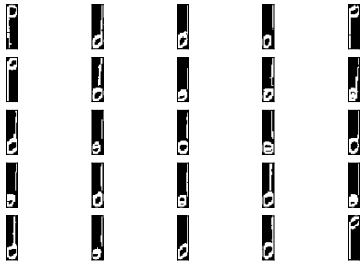
	precision	recall	f1-score	support
half	0.94	0.93	0.93	296
quarter	0.93	0.94	0.93	295
micro avg	0.93	0.93	0.93	591
macro avg	0.93	0.93	0.93	591
weighted avg	0.93	0.93	0.93	591

5.4 Παρατηρήσεις Πάνω Στην Ανάλυση Των Αποτελεσμάτων

Κάποιες φορές, όπως είναι λογικό, το σφάλμα μπορεί να μην είναι της μηχανής αλλά του ανθρώπου. Όταν η αναγνώριση συμβόλων έχει να κάνει με σύμβολα κοντινά σχηματικά το ένα στο άλλο μία λάθος γραμμή, ένα υπερβολικό γέμισμα ή και κάποια τελεία, μπορεί στο μάτι να φαίνονται σαν μικρά λάθη. Ο συνδυασμός τους όμως μπορεί να κάνει τη δουλειά ενός προγράμματος αναγνώρισης πολύ δύσκολη. Παρακάτω μπορούμε να δούμε κάποιες από τις νότες που έλαβαν μέρος στο κομμάτι της αναγνώρισης (πριν και μετά την αναπροσαρμογή μεγέθους αντίστοιχα).
Μπορούμε να δούμε ότι κάποιες από αυτές τις νότες είναι όντως καλοσχεδιασμένες και συνεπώς εύκολα ανιχνεύσιμες. Αλλά υπάρχουν και άλλες οι οποίες είναι πολύ βιαστικά σχεδιασμένες, πράγμα που έχει ως συνέπεια την ταξινόμηση τους σε λάθος κατηγορία. Για παράδειγμα στην πρώτη σειρά στις νότες αξίας τετάρτου (σχήμα 5.4) μπορούμε να δούμε ότι η κεφαλή της νότας (note head) είναι ζωγραφισμένη σαν



Σχήμα 5.4: Νότες αξίας τετάρτου (resized)



Σχήμα 5.5: Νότες μισής αξίας (resized)

γραμμή και όχι σαν κεφαλή. Συνεπώς, μπορούμε να δούμε ότι το σφάλμα δεν μπορεί να καταλογιστεί πάντα απόλυτα στη μηχανή.

Βιβλιογραφία

[1] *CS229 Lecture Notes*. Andrew Ng, 2008.

[2] *Wikipedia:* Οπτική Αναγνώριση Χαρακτήρων

[3] *Wikipedia:* OMR

[4] *Wikipedia:* k-nearest neighbors algorithm

[5] *Wikipedia:* Support-vector machine

[6] *Introduction to Optical Music Recognition: Overview and Practical Challenges* Jir Novotny and Jaroslav Pokorny

[7] *Feature Extraction Methods For Character Recognition (Survey)* Anil K. Jain and Torfinn Taxt, July 19, 1995

[8] *The MUSCIMA++ Dataset for Handwritten Optical Music Recognition. 14th International Conference on Document Analysis and Recognition, ICDAR 2017. Kyoto, Japan* Jan Hahic jr. and Pavel Pecina November 13-15 2017

[9] *CVC-MUSCIMA:A Ground-truth of Handwritten Music Score Images for Writer Identification and Staff Removal. International Journal on Document Analysis and Recognition, Volume 15, Issue 3, pp 243-251,* Alicia Fornes, Anjan Dutta, Albert Gordo, Josep Llados 2012

[10] *scikit-learn:* Machine Learning in Python (https://scikit-learn.org/)

Παραρτήματα

Παράρτημα Α'

Κώδικας

```
        import os
        from muscima.io import parse_cropobject_list

        #***LOAD DATA START
        cropobject_fnames = [os.path.join(CROPOBJECT_DIR, f) for f in
        os.listdir(CROPOBJECT_DIR)]
        docs = [parse_cropobject_list(f) for f in cropobject_fnames]

        #***EXTRACTING NOTES
        def extract_notes_from_doc(cropobjects):
:returns: quarter_notes, half_notes
_cropobj_dict = {c.objid: c for c in cropobjects}
notes = []
    for c in cropobjects:
        if (c.clsname == 'notehead-full') or (c.clsname ==
        'notehead-empty'):
            _has_stem = False
            _has_beam_or_flag = False
            stem_obj = None
            for o in c.outlinks:
                _o_obj = _cropobj_dict[o]
                if _o_obj.clsname == 'stem':
                    _has_stem = True
                    stem_obj = _o_obj
                elif _o_obj.clsname == 'beam':
                    _has_beam_or_flag = True
```

```
                elif _o_obj.clsname.endswith('flag'):
                    _has_beam_or_flag = True
            if _has_stem and (not _has_beam_or_flag):
                if len(stem_obj.inlinks) == 1:
                    notes.append((c, stem_obj))

quarter_notes = [(n, s) for n, s in notes if n.clsname ==
 'notehead-full']
half_notes = [(n, s) for n, s in notes if n.clsname ==
 'notehead-empty']
    return quarter_notes, half_notes

qns_and_hns = [extract_notes_from_doc(cropobjects) for cropobjects
 in docs]

#***KEEP NOTES IN LISTS
import itertools

qns = list(itertools.chain(*[qn for qn, hn in qns_and_hns]))
hns = list(itertools.chain(*[hn for qn, hn in qns_and_hns]))

len(qns), len(hns)


#***CREATING NOTE IMAGES
import numpy

def get_image(cropobjects, margin=1):
 top = min([c.top for c in cropobjects])
    left = min([c.left for c in cropobjects])
    bottom = max([c.bottom for c in cropobjects])
    right = max([c.right for c in cropobjects])
    # Create the canvas onto which the masks will be pasted
    height = bottom - top + 2 * margin
    width = right - left + 2 * margin
    canvas = numpy.zeros((height, width), dtype='uint8')


 for c in cropobjects:
        # Get coordinates of upper left corner of the CropObject
```

```
            # relative to the canvas
            _pt = c.top - top + margin
            _pl = c.left - left + margin
            # We have to add the mask, so as not to overwrite
            # previous nonzeros when symbol bounding boxes overlap.
            canvas[_pt:_pt+c.height, _pl:_pl+c.width] += c.mask

        canvas[canvas > 0] = 1
    return canvas

qn_images = [get_image(qn) for qn in qns]
hn_images = [get_image(hn) for hn in hns]

"""TEST VISUALIZATION OF NOTES
import matplotlib.pyplot as plt
def show_mask(mask):
    plt.imshow(mask, cmap='gray', interpolation='nearest')
    plt.show()
def show_masks(masks, row_length=5):
    n_masks = len(masks)
    n_rows = n_masks // row_length + 1
    n_cols = min(n_masks, row_length)
    fig = plt.figure()
    for i, mask in enumerate(masks):
        plt.subplot(n_rows, n_cols, i+1)
        plt.imshow(mask, cmap='gray', interpolation='nearest')
    for ax in fig.axes:
        ax.set_yticklabels([])
        ax.set_xticklabels([])
        ax.set_yticks([])
        ax.set_xticks([])
    plt.show()

show_masks(qn_images[:25])
show_masks(hn_images[:25])

#***FEATURE EXTRACTION
from skimage.transform import resize

qn_resized = [resize(qn, (40, 10)) for qn in qn_images]
```

```
hn_resized = [resize(hn, (40, 10)) for hn in hn_images]
for qn in qn_resized:
    qn[qn > 0] = 1
for hn in hn_resized:
    hn[hn > 0] = 1
"""TEST HOW RESIZE LOOKS
show_masks(qn_resized[:25])
show_masks(hn_resized[-25:])

#***CREATE TRAINING DATA
n_hn = len(hn_resized)
import random
random.shuffle(qn_resized)
qn_selected = qn_resized[:n_hn]

#MERGE TRAINING DATA INTO ONE DATASET
Q_LABEL = 1
H_LABEL = 0

qn_labels = [Q_LABEL for _ in qn_selected]
hn_labels = [H_LABEL for _ in hn_resized]

notes = qn_selected + hn_resized
# Flatten data
notes_flattened = [n.flatten() for n in notes]
labels = qn_labels + hn_labels

#ONE TRAIN TEST SPLIT
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    notes_flattened, labels, test_size=0.25, random_state=42,
    stratify=labels)

#Don't use all three methods at the same time

#CLASSIFY THE DATA KNN METHOD
from sklearn.neighbors import KNeighborsClassifier
K=5
# Trying the defaults first.
```

```
clf = KNeighborsClassifier(n_neighbors=K)
clf.fit(X_train, y_train)
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric=
'minkowski',metric_params=None, n_jobs=1, n_neighbors=5,
p=2, weights='uniform')

#CLASSIFY THE DATA SVM LINEAR METHOD
from sklearn.svm import SVC
svclassifier = SVC(kernel='linear')
svclassifier.fit(X_train, y_train)

#CLASSIFY THE DATA SVM KERNEL METHOD
from sklearn.svm import SVC
svclassifier = SVC(kernel='rbf')
svclassifier.fit(X_train, y_train)

#EXECUTE CLASSIFIER AND EVALUATE RESULTS
y_test_pred = clf.predict(X_test)

from sklearn.metrics import classification_report
print(classification_report(y_test, y_test_pred, target_names=
['half', 'quarter']))

# Import packages to visualize the classifer
from matplotlib.colors import ListedColormap
import matplotlib.pyplot as plt
import warnings

# Import packages to do the classifying
import numpy as np
from sklearn.svm import SVC

def versiontuple(v):
    return tuple(map(int, (v.split("."))))

def plot_decision_regions(X, y, classifier, test_idx=None,
 resolution=0.02):

    # setup marker generator and color map
    markers = ('s', 'x', 'o', '^', 'v')
```

```
    colors = ('red', 'blue', 'lightgreen', 'gray', 'cyan')
    cmap = ListedColormap(colors[:len(np.unique(y))])

    # plot the decision surface
    x1_min, x1_max = X[:, 0].min() - 1, X[:, 0].max() + 1
    x2_min, x2_max = X[:, 1].min() - 1, X[:, 1].max() + 1
    xx1, xx2 = np.meshgrid(np.arange(x1_min, x1_max, resolution),
                           np.arange(x2_min, x2_max, resolution))
    Z = classifier.predict(np.array([xx1.ravel(), xx2.ravel()]).T)
    Z = Z.reshape(xx1.shape)
    plt.contourf(xx1, xx2, Z, alpha=0.4, cmap=cmap)
    plt.xlim(xx1.min(), xx1.max())
    plt.ylim(xx2.min(), xx2.max())

    for idx, cl in enumerate(np.unique(y)):
        plt.scatter(x=X[y == cl, 0], y=X[y == cl, 1],
                    alpha=0.8, c=cmap(idx),
                    marker=markers[idx], label=cl)

    # highlight test samples
    if test_idx:
        # plot all samples
        if not versiontuple(np.__version__) >= versiontuple
        ('1.9.0'):X_test, y_test = X[list(test_idx), :],
        y[list(test_idx)]warnings.warn
        ('Please update to NumPy 1.9.0 or newer') else:
            X_test, y_test = X[test_idx, :], y[test_idx]

        plt.scatter(X_test[:, 0],
                    X_test[:, 1],
                    c='',
                    alpha=1.0,
                    linewidths=1,
                    marker='o',
                    s=55, label='test set')

np.random.seed(0)
X_xor = np.random.randn(200, 2)
y_xor = np.logical_xor(X_xor[:, 0] > 0,
                       X_xor[:, 1] > 0)
```

```
y_xor = np.where(y_xor, 1, -1)

plt.scatter(X_xor[y_xor == 1, 0],
            X_xor[y_xor == 1, 1],
            c='b', marker='x',
            label='1')
plt.scatter(X_xor[y_xor == -1, 0],
            X_xor[y_xor == -1, 1],
            c='r',
            marker='s',
            label='-1')

plt.xlim([-3, 3])
plt.ylim([-3, 3])
plt.legend(loc='best')
plt.tight_layout()
plt.show()
```