UNIVERSITY OF THESSALY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

DIPLOMA THESIS

PERFORMANCE EVALUATION OF ENVIRONMENTAL SOUND CLASSIFICATION METHODS

Author:
Athina Mimina

Supervisors:
Dimitrios Katsaros
Eleni Tousidou
Georgios Stamoulis

July 02 ,2019

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ Η/Υ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΑΞΙΟΛΟΓΗΣΗ ΕΠΙΔΟΣΗΣ ΜΕΘΟΔΩΝ ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗΣ ΠΕΡΙΒΑΛΛΟΝΤΙΚΩΝ ΗΧΩΝ

Συγγραφέας:
Αθηνά Μήμινα

Επιβλέποντες:
Δημήτριος Κατσαρός
Ελένη Τουσίδου
Γεώργιος Σταμούλης

Ιούλιος 02, 2019

Περίληψη

Σε αυτή τη διπλωματική εργασία, ερευνούμε τη χρησιμότητα της time series κατηγοριοποίησης με τη βοήθεια μεθόδων του τομέα της μηχανικής μάθησης με στόχο την αναγνώριση περιβαλλοντικών ήχων. Το κίνητρο για την ενασχόληση με αυτό το πρόβλημα ήταν το πως θα μπορούσε να βοηθήσει ο τομέας της μηχανικής μάθησης κοινωνικά ευπαθείς ομάδες, όπως άνθρωποι που δεν ακούν. Σκοπός λοιπόν αυτής της διπλωματικής είναι η δημιουργία ενός μοντέλου το οποίο θα αναγνωρίζει και θα κατηγοριοποιεί περιβαλλοντικούς ήχους.

Αναλυτικά, θα παρουσιαστεί μια εισαγωγή στον τύπο δεδομένων time series δεδομένων και βασικές τεχνικές επίλυσης του προβλήματος μας. Επίσης θα παρουσιαστεί αναλυτική πειραματική ανάλυση πάνω στις τεχνικές επίλυσης που μελετήθηκαν καθώς και διάφορες συγκρίσεις και αποτελέσματα πάνω σε αυτές.

Abstract

In this diploma thesis, we investigate the usefulness of time series classification with machine learning techniques, in order to classify environmental sounds. The motivation for occupation with this problem was how the field of machine learning could help socially vulnerable groups, such as people who can't hear. So, the purpose of this thesis is to build a model that can recognize and classify environmental sounds.

In detail, it will be presented an introduction in time series data and some basic solution techniques for our problem. Also, it will be presented an analytical experimental analysis on the solving techniques that we studied, aw well as some comparisons and results on them.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τους επιβλέποντες καθηγητές μου, καθηγητές Δημήτριο Κατσαρό, Ελένη Τουσίδου και Γεώργιο Σταμούλη, για την υποστήριξη, το κίνητρο και την απαραίτητη γνώση που μου έδωσαν ώστε να πραγματοποιηθεί αυτή η διπλωματική.

Επίσης θα ήθελα να ευχαριστήσω ιδιαίτερα τον καθηγητή Ηλία Χούστη που μέσα από τα μαθήματά του μου δόθηκε το έναυσμα και η απαραίτητη γνώση για να ασχοληθώ με τον τομέα της μηχανικής μάθησης, αλλά και όλους τους υπόλοιπους καθηγητές του τμήματος μου για την στήριξη και την επιμονή τους όλο αυτό τον καιρό.

Τέλος, θα ήθελα να εκφράσω την ευγνωμοσύνη μου στην οικογένεια μου και τους φίλους μου για τη στήριξη και τη συνεχόμενη ενθάρρυνση τους σε όλη την διάρκεια των σπουδών μου.

Περιεχόμενα

Κεφάλαιο 1 Εισαγωγή

1.1 Μηχανική μάθηση

Η μηχανική μάθηση είναι ένας πολλά υποσχόμενος τομέας τα τελευταία χρόνια, ο οποίος ανήκει στην Επιστήμη των υπολογιστών. Επικεντρώνεται σε αλγορίθμους οι οποίοι μαθαίνουν από τα δεδομένα και προσαρμόζονται σε αυτά για να κάνουν προβλέψεις ή για να πάρουν αποφάσεις σχετικά με αυτά. Παραδείγματα εφαρμογών που χρησιμοποιούνται οι αλγόριθμοι αυτοί είναι η υπολογιστική όραση, ο εντοπισμός spam email και η ανίχνευση ανωμαλιών. Είναι άμεσα συνδεδεμένος με τον τομέα της στατιστικής, μιας και αρχές της στατιστικής χρησιμοποιούνται για την επεξεργασία των δεδομένων.

Υπάρχουν αλγόριθμοι στη μηχανική μάθηση που μαθαίνουν σε έναν υπολογιστή να δρα βασιζόμενος στη λειτουργία του ανθρώπινου εγκεφάλου. Η τεχνική αυτή ονομάζεται deep learning και είναι αντικείμενο προσοχής πολλών επιστημόνων, μιας και τα μοντέλα της μπορούν να επιτύχουν επιδόσεις που νωρίτερα δεν έχουν παρατηρηθεί.

Η εξέλιξη της υπολογιστικής δύναμης και η διαθεσιμότητα τεράστιου όγκου δεδομένων επιτρέπει την ενασχόληση με πιο περίπλοκους αλγόριθμους. Έτσι όταν μελετάμε μοντέλα μηχανικής μάθησης αναμένουμε λιγότερη ανθρώπινη αλληλεπίδραση και πολύ υψηλές επιδόσεις. Ο κλάδος αυτός καλύπτει μεγάλο εύρος εφαρμογών στην κοινωνίας μας, από αναγνώριση προτύπων, μετάφραση κειμένου μέχρι εφαρμογές στην Ιατρική[1] και την Οικονομία[2].

1.2 Κίνητρο και σκοπός

Στη μηχανική μάθηση συχνά επιδιώκεται η μετατροπή μιας ακολουθίας δεδομένων σε μια ακολουθία από κλάσεις. Γνωστά παραδείγματα είναι η αναγνώριση χειρόγραφης γραφής και ομιλίας. Όλα τα παραπάνω έχουν κοινό το ότι παίρνουν ως είσοδο μια ακολουθία δεδομένων, όπως για παράδειγμα μια κυματομορφή, και προσπαθούν να προβλέψουν μια ακολουθία από κλάσεις που μοντελοποιούν τα δεδομένα εισόδου με το βέλτιστο τρόπο.

Στην κοινωνία μας υπάρχουν ευπαθείς ομάδες οι οποίες θα μπορούσαν να βοηθηθούν από τις εφαρμογές της μηχανικής μάθησης. Μια τέτοια ομάδα είναι οι άνθρωποι με προβλήματα ακοής. Σκοπός αυτής της διπλωματικής είναι η δημιουργία ενός μοντέλου το οποίο θα αναγνωρίζει και θα κατηγοριοποιεί περιβαλλοντικούς ήχους. Το πρώτο βήμα για τη δημιουργία του είναι η εκπαίδευση του. Δοθείσης μιας εισόδου από πολλαπλούς ήχους, όπως μηχανή αυτοκινήτου και μουσική, το μοντέλο καλείται να μάθει να τους ξεχωρίζει και να τους κατηγοριοποιεί στη σωστή κλάση. Μόλις αυτό επιτευχθεί δίνονται ως είσοδοι δεδομένα παρόμοιων ήχων,που δεν χρησιμοποιήθηκαν όμως στην εκπαίδευση, με σκοπό να δούμε αν το μοντέλο κατάφερε να προσαρμοστεί σε αυτά.

Πρέπει όμως να εισάγουμε τα κατάλληλα δεδομένα για το πρόβλημα που θέλουμε να λύσουμε. Η διαδικασία η οποία ετοιμάζει τα δεδομένα κατάλληλα για έναν αλγόριθμο μηχανικής μάθησης λέγεται preprocessing(προεπεξεργασία) και μπορεί να συνοψιστεί σε τρία βήματα: την επιλογή δεδομένων,το preprocess και τη μετατροπή τους. Μόλις επιλεγούν τα δεδομένα πρέπει να σκεφτούμε πως θα τα

χρησιμοποιήσουμε. Κάποια από τα απαραίτητα λοιπόν βήματα είναι τα εξής[3] :

- Formatting: κατάλληλη μορφή αρχείου
- Cleaning: αφαίρεση ελλειπουσών τιμών
- Sampling: μικρότερο δείγμα από τα επιλεγμένα δεδομένα
- Scaling: ίδια κλίμακα σε όλα τα δεδομένα
- Decomposition: απλοποίηση δεδομένων
- Aggregation: συσσωμάτωση των δεδομένων

Στην περίπτωση των ήχων τα αρχεία μας από wav, mp3 ή wma format πρέπει να τα φέρουμε σε ένα format πιο κατανοητό. Πρέπει λοιπόν να τα αναπαραστήσουμε στο πεδίο της συχνότητας ή με βάση του ρυθμού δειγματοληψίας τους. Η διαδικασία αυτή ονομάζεται feature extraction και οι πιο γνωστές τεχνικές για να επιτευχθεί είναι η Mel Frequency Cepstral Coefficient(MFCC) και η Hamming window.

Όλα τα παραπάνω είναι απαραίτητα για την επίλυση του προβλήματος μας, το οποίο ονομάζεται time series classification, και είναι αντικείμενο μελέτης σε πολλούς τομείς για παραπάνω από μια δεκαετία. Για παράδειγμα, οι γεωλόγοι καταγράφουν σεισμικά κύματα για την πρόβλεψη τυχόν σεισμικών δονήσεων. Οι μετεωρολόγοι καταγράφουν την ταχύτητα του ανέμου, την καθημερινή μέγιστη και ελάχιστη θερμοκρασία. Οι οικονομολόγοι θα μπορούσαν να χρησιμοποιήσουν μια μέθοδο κατηγοριοποίησης χωρών με τον ίδιο οικονομικό δείκτη. Όλα τα παραπάνω είναι αρκετά παραδείγματα τα οποία θα μπορούσε να αναφέρει κανείς για να περιγράψει την εφαρμογή των time series στην κοινωνία μας.

Έτσι εφαρμόζοντας το μοντέλο με την καλύτερη επίδοση σε μια κατάλληλη πλατφόρμα και ίσως με την μετατροπή των προβλέψεων σε κείμενο, οι άνθρωποι εκείνοι θα έχουν την δυνατότητα να αντιλαμβάνονται τι διαδραματίζεται στο περιβάλλον τους. Για παράδειγμα όταν εντοπίζεται ένας ήχος από την εφαρμογή, όπως ο ήχος ενός κουδουνιού, να λαμβάνεται ένα μήνυμα με την κατηγορία στην οποία ανήκει.

Το time series classification περιλαμβάνει supervised και unsupervised learning. Οι δύο αυτές κατηγορίες, αλλά και κάποιες τεχνικές επίλυσης τους, θα αναλυθούν και θα συγκριθούν αργότερα σε αυτή τη διπλωματική.

1.3 Συνεισφορά

Η κατηγοριοποίηση των time series είναι τομέας της μηχανικής μάθησης που αποσκοπεί στον σωστό διαχωρισμό των δεδομένων ανάλογα με τον τύπο τους. Πιο συγκεκριμένα ο στόχος είναι να δοθεί η δυνατότητα στις μηχανές να αναγνωρίζουν με λεπτομέρεια και ακρίβεια τα δεδομένα. Αυτή η διπλωματική παρουσιάζει μια επισκόπηση του κλάδου του classification από περιβαλλοντικούς ήχους αλλά και την επιπλέον έρευνα του προβλήματος. Μέσα από την έρευνα των μεθόδων και των αλγορίθμων που μελετήθηκαν, παρέχονται συγκρίσεις και συμπεράσματα πάνω σε αυτές.

Με λίγα λόγια οι συνεισφορές της εν λόγω εργασίας μπορούν να συνοψιστούν ως:

- Περαιτέρω έρευνα μεθόδων

- Σύγκριση αποτελεσμάτων
- Σωστή χρήση τεχνικών επίλυσης των αλγορίθμων

1.4 Δομή της διπλωματικής

Η παρούσα διπλωματική είναι χωρισμένη σε 5 κεφάλαια:

- Εισαγωγή
- Time series και ανασκόπηση βιβλιογραφίας
- Βασικές τεχνικές επίλυσης
- Πειραματική ανάλυση και αποτελέσματα
- Συμπεράσματα και μελλοντική εργασία

ΚΕΦΑΛΑΙΟ 1: αποκαλύπτει το θέμα και γιατί είναι σημαντική η επίλυσή του.

ΚΕΦΑΛΑΙΟ 2: παραθέτει μια εισαγωγή στα time series και δίνει μια γενική εικόνα των προσπαθειών που έχουν γίνει μέχρι τώρα πάνω στο συγκεκριμένο πρόβλημα.

ΚΕΦΑΛΑΙΟ 3:παρέχει θεωρητικές εξηγήσεις των αλγορίθμων που χρησιμοποιήθηκαν για τη δημιουργία του μοντέλου.

ΚΕΦΑΛΑΙΟ 4: παρουσιάζει αναλυτική εξήγηση πάνω στη δομή του μοντέλου και στις δοκιμές που πραγματοποιήθηκαν για την επίτευξη της μέγιστης δυνατής επίδοσης του.

ΚΕΦΑΛΑΙΟ 5: παρατίθενται τα αποτελέσματα που πάρθηκαν καθώς και τα συμπεράσματα πάνω σε αυτά. Τέλος γίνεται λόγος για πιθανή μελλοντική εργασία πάνω στο θέμα μας.

Κεφάλαιο 2 Time series και ανασκόπηση βιβλιογραφίας

2.1 Εισαγωγή στα time series

Τα time series δηλώνουν μια μορφή αποθήκευσης δεδομένων η οποία περιέχει μονάδες χρόνου και την ανάλογη τιμή που αντιστοιχεί σε αυτές. Οι τιμές των ακολουθιών πρέπει να έχουν το ίδιο νόημα και να σχετίζονται μεταξύ τους. Σημαντική είναι και η σωστή διάταξή τους καθώς επηρεάζει την εξάρτηση των σημείων και τη σημασία των δεδομένων. Τα σημεία που διαμορφώνουν τις χρονολογικές σειρές έχουν προκαθορισμένες ιδιότητες. Μια από αυτές είναι ότι τα σημεία πρέπει να δειγματοληπτούνται μέσω επαναλαμβανόμενων μετρήσεων με τη πάροδο του χρόνου σε ίσα διαστήματα. Άλλος τρόπος είναι οι τιμές να μετριούνται για συγκεκριμένα χρονικά σήματα που μπορούν να συμβούν περιοδικά ή περιστασιακά και το αποτέλεσμα να είναι ένα διακριτό σύνολο τιμών.

Παρά τη συνεχόμενη ανάπτυξη των μεθόδων ανάλυσης, υπάρχουν ακόμα πολλά προβλήματα ως προς την ανάλυση που χρήζουν βελτιστοποίηση. Ένα από τα βασικά προβλήματα είναι ο θόρυβος που περιέχουν τα δεδομένα και η μεγάλη διάσταση τους. Έτσι γίνεται δύσκολη η μοντελοποίηση τους από τους ήδη υπάρχοντες αλγορίθμους λόγω μη ύπαρξης μη γραμμικών επιλογών. Ένας τρόπος να επιλυθεί είναι να μειωθεί η διάσταση των δεδομένων και να αφαιρεθεί ο θόρυβος μέσω διαφόρων τεχνικών, με κίνδυνο όμως να χαθεί σημαντική πληροφορία.

Μια άλλη δυσκολία είναι ο τεράστιος αριθμός των data points, δηλαδή ο μεγάλος όγκος δεδομένων γιατί αλλοιώνεται η ακρίβεια μιας λεπτομερούς ανάλυσης, μιας και ο κίνδυνος μη σωστής επεξεργασίας των δεδομένων αυξάνεται. Ένας τρόπος αντιμετώπισης είναι η μετατροπή των δεδομένων σε χαρακτηριστικά αμετάβλητου χώρου.

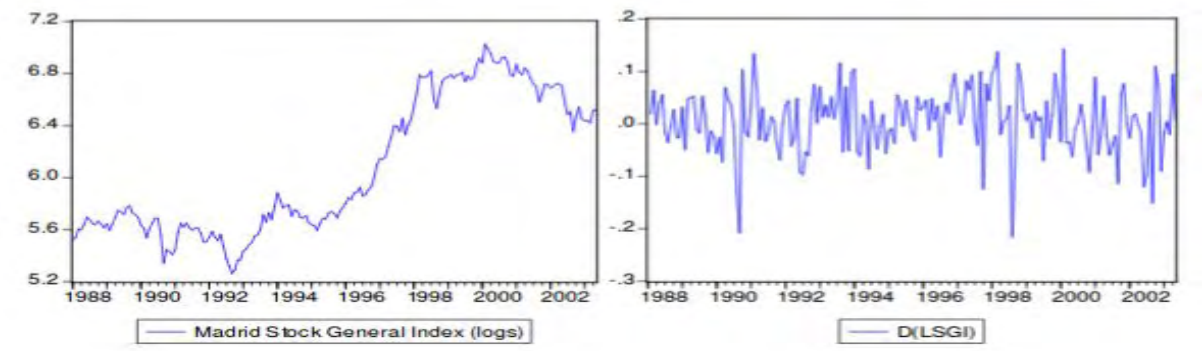
2.2 Τύποι time series κατηγοριοποίησης

Υπάρχουν πολλοί τύποι time series κατηγοριοποίησης βασιζόμενοι σε ορισμένα κριτήρια. Τα βασικότερα είναι το μήκος του time step, η μνήμη και η σταθερότητα.

Ανάλογα με τη σταθερότητα[3] η κατηγοριοποίηση διακρίνεται σε:

- Στατικά time series
- Μη στατικά time series

Ένα time series ονομάζεται στατικό όταν δεν υπάρχει συστηματική αλλαγή σε ορισμένες στατιστικές ιδιότητες τους όπως το mean(μέση τιμή), variance(διασπορά), autocorrelation(αυτοσυσχέτιση). Όταν οι τιμές των παραπάνω δε μένουν σταθερές με το χρόνο έχουμε μη στατικά time series. Τις περισσότερες φορές απαιτείται η μετατροπή των μη στατικών σε στατικά, με κατάλληλες μεθόδους προεπεξεργασίας. Όμως υπάρχουν περιπτώσεις όπου μη στατικά χαρακτηριστικά έχουν περισσότερο ενδιαφέρον μελέτης.



Εικόνα 1: Αριστερά βλέπουμε παράδειγμα μη στατικού χαρακτηριστικού και δεξιά ένα παράδειγμα στατικού

Το επόμενο κριτήριο κατηγοριοποίησης των time series στηρίζεται στη μνήμη. Ανάλογα με το ποσοστό εξάρτησης από προηγούμενες χρονικές στιγμές[5] η κατηγοριοποίηση χωρίζεται σε δύο τύπους:

- Long memory time series
- Short memory time series

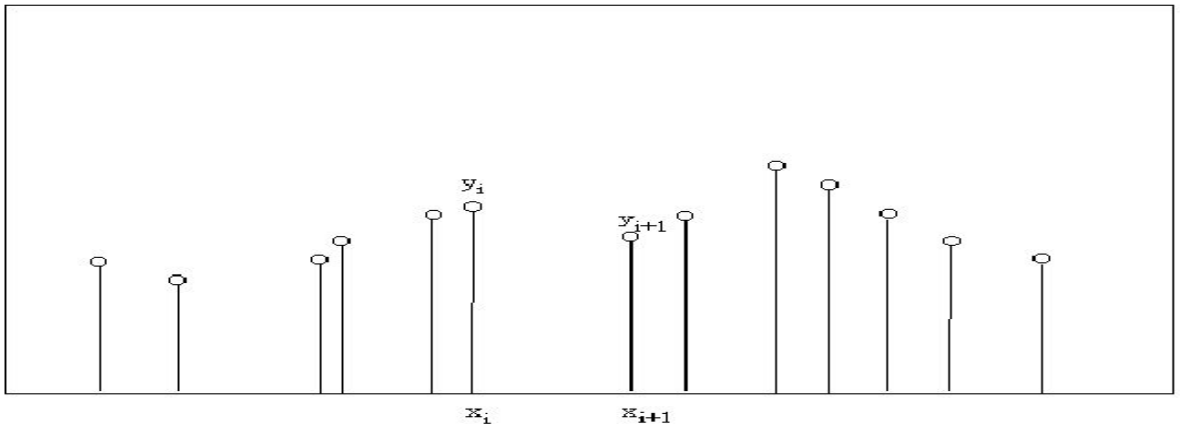
Για να κατανοηθεί η διαφορά τους πρέπει πρώτα να γίνει αναφορά στη συνάρτηση αυτοσυσχέτισης. Η συσχέτιση είναι στατιστική μέθοδος η οποία χρησιμοποιείται για να κατανοηθεί πόσο ισχυρά σχετίζονται μεταξύ τους οι τιμές ενός ζεύγους τιμών. Έτσι με τον όρο αυτοσυσχέτιση αναφερόμαστε στη συσχέτιση ενός δοσμένου σήματος με τον εαυτό του σε διάφορα σημεία στο χρόνο και είναι χρήσιμο γιατί μπορεί να βρει επαναλαμβανόμενα πρότυπα στα δεδομένα ακόμα και αν περιέχουν θόρυβο.

Όταν λοιπόν μιλάμε για long memory αναφερόμαστε στο επίπεδο στατιστικής εξάρτησης μεταξύ δύο σημείων της αλληλουχίας, όταν αυξάνουμε την απόσταση μεταξύ τους. Πιο συγκεκριμένα, χρησιμοποιείται η $y = a^x$ ως threshold. Όταν η συνάρτηση αυτοσυσχέτισης μειώνεται με πιο αργό ρυθμό από το threshold, έχουμε long memory time series. Όταν η συνάρτηση μειώνεται πιο ραγδαία έχουμε short memory time series.

Το τελευταίο κριτήριο κατηγοριοποίησης εξαρτάται από το μήκος που έχουν τα time series. Ανάλογα με την απόσταση μεταξύ των τιμών που έχουν καταγραφεί η κατηγοριοποίηση διαχωρίζεται σε δύο τύπους:

- Ισαπέχοντα time series
- Μη ισαπέχοντα time series

Όταν οι τιμές των δεδομένων καταγράφονται περιοδικά με σταθερή περίοδο τότε κατασκευάζονται ισαπέχοντα time series. Όταν τα time series δεν κρατούν τη σταθερή απόσταση μεταξύ των παρατηρήσεων έχουμε μη ισαπέχοντα. Ένα παράδειγμα τέτοιων σημείων φαίνεται παρακάτω.



Εικόνα 2 : Παράδειγμα μη ισαπέχοντων χαρακτηριστικών

2.3 Learning

Οποιαδήποτε τεχνική ή μέθοδος μπορεί να μας δώσει πληροφορίες με επεξεργασία και ανάλυση ενός συνόλου δεδομένων για τον σχεδιασμό ενός κατηγοριοποιητή ονομάζεται learning. Supervised, unsupervised και reinforced learning είναι οι γενικές μορφές της

μηχανικής μάθησης. Πριν εξηγηθούν όμως οι τύποι αυτοί θα πρέπει να αναφερθούμε σε κάποιους τύπους δεδομένων.

Υπάρχουν δύο κατηγορίες δεδομένων που αντιμετωπίζονται διαφορετικά και είναι τα labelled(προσημασμένα) και unlabelled(μη προσημασμένα) δεδομένα. Η βασική διαφορά τους είναι ότι τα πρώτα χρησιμοποιούνται για την πρόβλεψη μιας κλάσης-στόχου μέσα από καινούργιες παρατηρήσεις, και τα δεύτερα είναι χρήσιμα για clustering ή για την ανακάλυψη καινούργιων συσχετίσεων μεταξύ των δεδομένων. Τα υπόλοιπα δεδομένα που ανήκουν σε καμία από τις παραπάνω κατηγορίες ονομάζονται features στη μηχανική μάθηση.

2.3.1 Supervised learning

Τα δεδομένα που περιέχουν την πληροφορία της κλάσης δίνονται για εκπαίδευση και με βάση τους αλγορίθμους του supervised learning αξιολογούνται ώστε να παραχθεί ένα μοντέλο το οποίο θα κατηγοριοποιεί τα καινούρια δείγματα. Όταν το μοντέλο εκπαιδευτεί θα είναι έτοιμο να κατατάξει σε κατηγορίες τα καινούρια δείγματα. Στην ουσία ψάχνουμε την κατάλληλη συνάρτηση που θα μειώνει το σφάλμα της πρόβλεψης ή τον αριθμό των λαθών.

2.3.2 Unsupervised learning

Αν η ετικέτα της κλάσης δεν παρέχεται τότε έχουμε unsupervised learning. Σκοπός του είναι να ανιχνεύσει επιπλέον κανόνες συσχέτισης(association rules) μεταξύ των δεδομένων ή να βρει ομοιότητες μεταξύ τους με σκοπό να τα ομαδοποιήσει(Clustering). Η

ανίχνευση νέων association rules συσχετίζεται με τον καθορισμό των μέτρων απόστασης.

2.4 Τύποι μεταβλητών και προετοιμασία δεδομένων

Οι μεταβλητές εισόδου καθορίζουν τις στατιστικές ιδιότητες που χρησιμοποιούνται στις μεθόδους μάθησης. Κάποιες από αυτές τις ιδιότητες χρησιμοποιούνται για ποσοτικές μελέτες και κάποιες για ποιοτικές. Υπάρχουν τέσσερις βασικοί τύποι που μεταβλητών εισόδου[6]:

- Nominal variables
- Ordinal variables
- Ratio variables
- Interval variables

Οι πρώτες δύο ονομάζονται κατηγορικές μεταβλητές. Οι nominal παρέχουν περιγραφικές πληροφορίες για τη διάκριση ενός αντικειμένου από ένα άλλο. Μερικά παραδείγματα τέτοιων μεταβλητών είναι η διάκριση μεταξύ των χρωμάτων, τα νούμερα στις μπλούζες των αθλητών. Οι ordinal είναι παρόμοιες με τις nominal, με τη διαφορά ότι περιγράφουν κατάταξη και μπορούν να επεξεργαστούν βάση αυτής. Μεταβλητές που δηλώνουν υψηλή, μεσαία ή χαμηλή βαθμολογία αποτελούν παράδειγμα ordinal μεταβλητών.

Οι interval μεταβλητές περιέχουν αριθμητικές τιμές ίδιας κλίμακας όπως η θερμοκρασία. Σε αυτή την κατηγορία η απόσταση μεταξύ των μεταβλητών εισόδου έχει σημασία. Για το λόγο αυτό έχει νόημα να

υπολογίζεται ένας μέσος όρος των interval μεταβλητών. Οι ratio μεταβλητές είναι παρόμοιες με τις interval, με διαφορά ότι η παρουσία μηδενικής τιμής δηλώνει την απουσία μετρήσιμης στατιστικής ιδιότητας ενός χαρακτηριστικού.

Κατά τη διαδικασία συλλογής δεδομένων δημιουργούνται σφάλματα τα οποία επηρεάζουν αρνητικά τη διαδικασία μάθησης. Οι περιπτώσεις στις οποίες υπάρχουν outlier points ή χαρακτηριστικά των οποίων οι τιμές λείπουν πρέπει να αντιμετωπίζονται πριν ξεκινήσει η διαδικασία της μάθησης ώστε να παράγονται ορθά αποτελέσματα. Επιπλέον, όταν εντοπίζονται δεδομένα με διαφορετική κλίμακα τιμών θα πρέπει να μετατρέπονται όλα σε μία κλίμακα ώστε να μην παράγει λανθασμένα αποτελέσματα η συνάρτηση κόστους.

2.5 Train-test-validation set

Πριν αναφερθούμε στον διαχωρισμό των δεδομένων ας δούμε τη μορφή έχουν τα δεδομένα.

Ένα σύνολο δεδομένων αποτελείται από ένα σύνολο ζευγαριών (x,y) όπου το x είναι ένα διάνυσμα τιμών (features) και το y είναι η ετικέτα των κατηγοριών ή αλλιώς η τιμή κατηγοριοποίηση για το x. Σκοπός είναι να ανακαλυφθεί μια συνάρτηση κόστους $y = f(x)$ που προβλέπει με τον βέλτιστο τρόπο την τιμή του y που σχετίζεται με κάθε τιμή του x. Αν το διάνυσμα του y περιέχει μόνο τις τιμές 0 ή 1, δηλαδή έχουμε την ύπαρξη μόνο δύο κατηγοριών, η κατηγοριοποίηση ονομάζεται binary classification. Αν περιέχονται τιμές που να δηλώνουν περισσότερες από δύο κλάσεις η κατηγοριοποίηση μας ονομάζεται multiclass classification.

Ο διαχωρισμός των δεδομένων σε training set, test set και validation set είναι πολύ διαδεδομένος για τη στατιστική ανάλυση των δεδομένων. Το πρώτο σύνολο δεδομένων χρησιμοποιείται για να γίνει το fitting στο μοντέλο supervised μάθησης. Στη συνέχεια υπολογίζονται τα σφάλματα προβλέψεων χρησιμοποιώντας το validation set, για την επιλογή του πιο βολικού μοντέλου. Έπειτα το test set χρησιμοποιείται για να ελέγξει τη γενίκευση του σφάλματος για το επιλεγμένο και συντονισμένο τελικό μοντέλο. Ο παραπάνω διαχωρισμός των δεδομένων στα set γίνεται με διάφορες τεχνικές με πιο απλή τον τυχαίο διαχωρισμό.

Άλλη μία ευρέως διαδεδομένη τεχνική για το διαχωρισμό των δεδομένων είναι αυτή του k-fold cross validation. Ο κύριος σκοπός της είναι να επιτευχθεί μια σταθερή και σίγουρη εκτίμηση των προβλέψεων. Η βασική ιδέα είναι ότι το σύνολο δεδομένων χωρίζεται σε k τμήματα ίσου μεγέθους. Ένα τμήμα αντιπροσωπεύει το test set και τα υπόλοιπα το training set. Η διαδικασία επαναλαμβάνεται για κάθε τμήμα των δεδομένων. Παρακάτω παρουσιάζεται αναλυτικά ο αλγόριθμος [9].

Algorithm 2 K-fold cross-validation

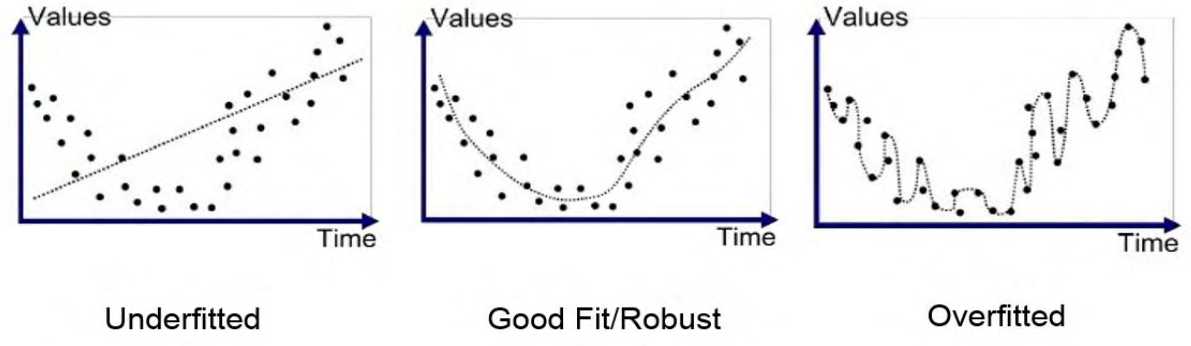
1. Input: dataset T, number of folds k, performance function $error$, computational models $L_1, \cdots, L_m, m \geq 1$
2. Divide T into k disjoint subsets $T_1, \cdots, T_k$ of the same size.
3. For $i = 1, \cdots, k$:
   $T_v \leftarrow T_i, T_{tr} \leftarrow \{T \setminus T_i\}$.
   3.1. For $j = 1, \cdots, m$:
   Train model L_j on T_{tr} and periodically use T_v to asses the model performance:
   $E_v^j(i) = error(L_j(T_v))$.
   Stop training, when a stop-criterion based on $E_v^j(i)$ is satisfied.
4. For $j = 1, \cdots, m$, evaluate the performance of the models by: $E_v^j = \frac{1}{k} \cdot \sum_{i=1}^{k} E_v^j(i)$.

Εικόνα 3: Αναλυτική παρουσίαση του αλγορίθμου k-fold cross validation

2.6 Overfitting και underfitting

Συχνά στη μηχανική μάθηση αντιμετωπίζεται το πρόβλημα του overfitting. Αυτό σημαίνει ότι τα εκπαιδευόμενα μοντέλα μπορεί να επιτυγχάνουν την τέλεια κατηγοριοποίηση στα training δεδομένα αλλά όχι στα νέα δεδομένα(test set). Αποτελεί ένα από τα πιο σημαντικά αντικείμενα μελέτης στη μηχανική μάθηση και αυτό γιατί πολύπλοκα μοντέλα τείνουν στο overfitting και τα πιο απλά δομημένα αδυνατούν να κατηγοριοποιήσουν σωστά τα δείγματα.

Με τον όρο underfitting αναφερόμαστε σε ένα μοντέλο το οποίο αδυνατεί να κατηγοριοποιήσει τόσο τα training δεδομένα όσο και τα νέα δείγματα. Έτσι έχουμε ένα ακατάλληλο μοντέλο το οποίο έχει χαμηλή επίδοση και ο τρόπος αντιμετώπισης είναι η δοκιμή ενός νέου εναλλακτικού αλγορίθμου. Στο παρακάτω γράφημα γίνεται πιο αντιληπτή η εξήγηση των παραπάνω[10].



Εικόνα 4: Αριστερά βλέπουμε ένα παράδειγμα underfitting, δεξία ένα παράδειγμα overfitting και στη μέση βλέπουμε την επιθυμητή κατάσταση

2.7 Συναφείς εργασίες στον τομέα μέχρι σήμερα

Όπως αναφέρθηκε παραπάνω τα time series δεδομένα αποτελούν αντικείμενο μελέτης την τελευταία δεκαετία. Συγκεκριμένα η αναγνώριση ήχου αποτελεί την μεγαλύτερη πρόκληση στον τομέα. Η πιο γνωστή αναγνώριση ήχου παγκοσμίως είναι το Shazam[8]. Το Shazam είναι η πιο διαδεδομένη εφαρμογή αναγνώρισης τραγουδιών και χρησιμοποιείται σχεδόν από κάθε χρήστη κινητής συσκευής. Ο χρήστης ηχογραφεί δείγμα 10 δευτερολέπτων με το μικρόφωνο της συσκευής χρησιμοποιώντας την εφαρμογή και αυτόματα δημιουργείται μια κωδικοποίηση για το τραγούδι αυτό. Η κωδικοποίηση αυτή ανεβαίνει στο server και εκεί γίνεται μια αναζήτηση για το αν υπάρχει αντιστοιχία σε ήδη υπάρχουσα κωδικοποίηση στη βάση. Αν βρεθεί αντιστοιχία οι πληροφορία του τραγουδιού επιστρέφεται στο χρήστη. Παρόμοιες εφαρμογές είναι οι ACRCloud και η Gracenote.

Η επόμενη επίτευξη στο χώρο της αναγνώρισης ήχου βρίσκεται πάλι στα κινητά μας τηλέφωνα. Κάθε συσκευή διαθέτει έναν ψηφιακό βοηθό με τον οποίο ο χρήστης αλληλεπιδρά μέσω της φωνής του. Ο βοηθός ανάλογα με την επιθυμία του χρήστη έχει τη δυνατότητα να αναζητήσει στο διαδίκτυο, να προγραμματίσει γεγονότα και ξυπνητήρια, να προσαρμόσει τις ρυθμίσεις στη συσκευή και ότι άλλο του ζητηθεί από το χρήστη. Τέτοιοι βοηθοί είναι ο Google Assistant[9], η Siri της Apple[10],η Alexa της Amazon και η Cortana της Microsoft[11]. Ο βοηθός αυτό μπορεί πέρα από τις κινητές συσκευές να τοποθετηθεί σε αυτοκίνητα,σε tablet και σε έξυπνα σπίτια.

Άλλο επίτευγμα της Google στον τομέα είναι η δημιουργία των ακουστικών Pixel Buds[12], τα οποία έχουν τη δυνατότητα να μεταφράζουν οτιδήποτε πει ο χρήστης σε τουλάχιστον 40 γλώσσες. Για να επιτευχθεί αυτό χρειάστηκαν αρκετές ξεχωριστές πληροφορίες. Αρχικά χρησιμοποιήθηκε ένας ανιχνευτής δραστηριότητας φωνής(VAD)[13] και ένας ανιχνευτής αναγνώρισης γλώσσας(LID)[14]. Για την σύνθεση του τελικού αποτελέσματος χρησιμοποιήθηκαν αυτόματη αναγνώριση ομιλίας[15], επεξεργασία φυσικής γλώσσας[16] και σύνθεση ομιλίας[17].

Η αυτόματη αναγνώριση ομιλίας και η επεξεργασία φυσικής γλώσσας αποτελούν από μόνες τους σημαντικές ανακαλύψεις στον κλάδο. Η πρόκληση ήταν πως θα επιτευχθεί ο προγραμματισμός ενός υπολογιστή για την επεξεργασία και ανάλυση μεγάλου όγκου δεδομένων φυσικής γλώσσας και ξεκίνησε να μελετάται το 1950. Η επίτευξη του στόχου βασίστηκε στη μηχανική μάθηση με τη χρήση νευρωνικών δικτύων. Από τις πρώτες αξιόλογες προσπάθειες αναγνώρισης ομιλίας είναι αυτή της αναγνώρισης ομιλίας με τη βοήθεια των Hidden Markov Models(HMM) [18] και είναι αυτή που εδραίωσε την εξελιξη της αναγνώρισης ομιλίας σε πιο σύνθετα και αποδοτικά μοντέλα. Τέτοια μοντέλα είναι τα RNN, LSTM νευρωνικά δίκτυα [19]. Επιπρόσθετα, υπάρχουν πολλές αναφορές για το πώς κατανοήθηκε η φυσική γλώσσα [20] και για την αρχιτεκτονική των μοντέλων[21].

Η διαδικτυακή πλατφόρμα Braci smart ear[22] είναι η μοναδική πλατφόρμα αναγνώρισης και ανάλυσης περιβαλλοντικών ηχών, η οποία μετατρέπει τους ήχους αυτούς σε οπτικές και αισθητήριες ειδοποιήσεις. Πλέον, διατίθεται και σαν εφαρμογή για κάθε κινητή συσκευή.

Τέλος, η επόμενη σημαντική έρευνα πάνω στην αναγνώριση περιβαλλοντικών ήχων έγινε με time-frequency δεδομένα[23]. Για την μετατροπή των αρχείων ήχου σε time-frequency χρησιμοποιήθηκε τεχνική η οποία βασίζεται στο μετασχηματισμό Fourier. Διεξήχθησαν εκτενή πειράματα για να αποδειχθεί η αποτελεσματικότητα αυτής της μετατροπής καθώς και ακουστικές δοκιμές για να αναλυθεί η ανθρώπινη ικανότητα αναγνώρισης. Τα αποτελέσματα τους έδειξαν ότι το μοντέλο τους παράγει συγκρίσιμες επιδόσεις με την ανθρώπινη ακοή.

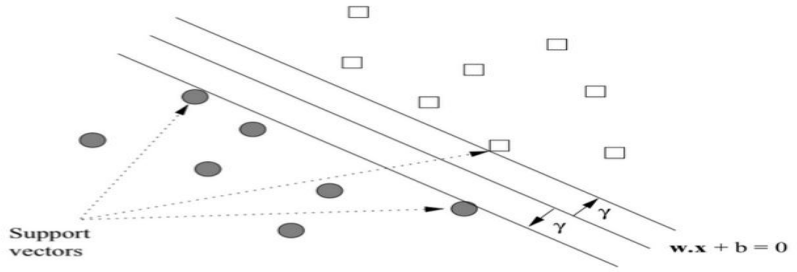
Κεφάλαιο 3 Βασικές τεχνικές επίλυσης

3.1 Support Vector Machines

Τα support vector machines (SVMs)[24] είναι μοντέλα supervised μάθησης που χρησιμοποιούν αλγορίθμους οι οποίοι αναλύουν δεδομένα με σκοπό να τα κατηγοριοποιήσουν. Δοσμένου ενός συνόλου από δείγματα εκπαίδευσης που ανήκουν σε κάποιες κατηγορίες, τα SVM δημιουργούν ένα μοντέλο που αναθέτει τα καινούρια δείγματα σε μια από αυτές τις κλάσεις με τη βοήθεια ενός μη πιθανοτικού κατηγοριοποιητή. Στην ουσία το μοντέλο είναι η αναπαράσταση των παραδειγμάτων ως σημεία στο χώρο έτσι ώστε παραδείγματα διαφορετικών κατηγοριών να διαχωρίζονται μεταξύ τους με ένα μέγιστο δυνατό κενό με τη βοήθεια ενός hyperplane.

3.1.1 Μηχανισμός ενός SVM

Σκοπός ενός SVM είναι να επιλέξει ένα hyperplane $w.x + b = 0$ δηλαδή μια γραμμή διαχωρισμού στο επίπεδο, που μεγιστοποιεί την απόσταση(margin) γ μεταξύ του hyperplane και οποιουδήποτε σημείου στο training set. Είναι επιθυμητό όλα τα training points να είναι όσο το δυνατόν πιο μακριά γίνεται από το hyperplane γιατί έτσι είμαστε πιο σίγουροι για το ποια κατηγορία ανήκουν τα σημεία.



Εικόνα 5: Ένα SVM επιλέγει το hyperplane με τη μέγιστη απόσταση γ μεταξύ του διαχωριστικού hyperplane και των εκπαιδευόμενων δειγμάτων

Από την παραπάνω εικόνα παρατηρούμε ότι υπάρχουν δύο παράλληλα hyperplane σε απόσταση γ από το κεντρικό hyperplane $w.x + b = 0$ και κάθε ένα από αυτά αγγίζει ένα ή περισσότερα support vectors. Τα support vectors είναι τα σημεία τα οποία περιορίζουν το hyperplane διαχωρισμού, με την έννοια ότι είναι όλα σε απόσταση γ από το hyperplane. Ένα σύνολο σημείων με d διαστάσεις έχει $d + 1$ support vectors. Υπάρχουν όμως περιπτώσεις όπου τα support vectors μπορεί να είναι περισσότερα. Αυτό συμβαίνει όταν πάρα πολλά σημεία βρίσκονται πάνω στα παράλληλα hyperplane.

Σκοπός λοιπόν είναι:

- Δοσμένου ενός training set (x1,y1),(x2,y2),...,(xn,yn), μεγιστοποιήστε το γ μεταβάλλοντας τα w, b, υπό τον περιορισμό ότι για όλα τα $i = 1, 2, ..., n$ να ισχύει:

$$yi(w.xi + b) \geqslant \gamma$$

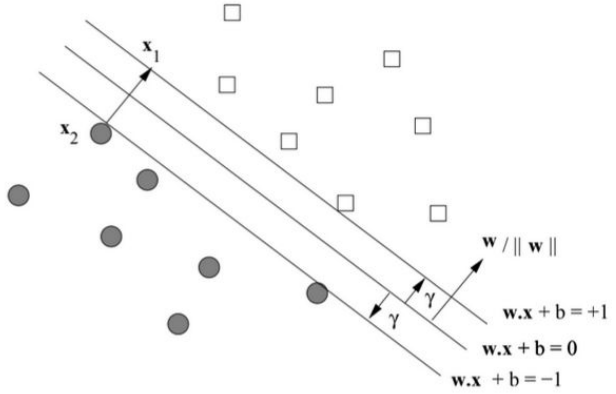
Το yi, το οποίο πρέπει να είναι +1 ή -1, καθορίζει την πλευρά του hyperplane που θα ενταχθεί το σημείο xi, οπότε η ανίσωση είναι πάντα αληθής. Όμως αυτή η υπόθεση δεν δουλεύει πάντα σωστά διότι αν αυξήσουμε τα w, b μπορούμε πάντα να επιτρέπουμε μια πολύ μεγάλη τιμή στο γ.

3.1.2 Κανονικοποίηση του Hyperplane

Το παραπάνω πρόβλημα λύνεται κανονικοποιώντας το διάνυσμα βάρους w. Δηλαδή η μονάδα μέτρησης που είναι κάθετη στο διαχωριστικό hyperplane είναι το διάνυσμα $w/||w||$. Θυμίζουμε ότι το $||w||$ είναι η τετραγωνική ρίζα του αθροίσματος των τετραγώνων των

στοιχείων του w. Θα απαιτήσουμε λοιπόν το w να είναι τέτοιο ώστε τα παράλληλα hyperplane που αγγίζουν τα support vectors να περιγράφονται από τις εξισώσεις $w.x + b = 1$ και $w.x + b = -1$.

Έτσι σκοπός είναι να μεγιστοποιήσουμε το γ, που είναι τώρα το πολλαπλάσιο της μονάδας $w/||w||$ μεταξύ του διαχωριστικού hyperplane και των παράλληλων μέσω των support vector machines.



Εικόνα 6: Κανονικοποίηση του διανύσματος βάρους για ένα SVM

Ας υποθέσουμε ότι το x1 είναι η προβολή του διανύσματος x2 πάνω στο πιο μακρινό hyperplane. Η απόσταση από το x2 στο x1 σε μονάδες $w/||w||$ είναι 2γ. Δηλαδή :

$$x1 = x2 + 2\gamma * w/||w|| \qquad (1)$$

Όπως φαίνεται από την εικόνα το x1 βρίσκεται πάνω στο hyperplane που χαρακτηρίζεται από την εξίσωση $w.x + b = 1$ άρα γνωρίζουμε ότι $w.x1 + b = 1$. Αντικαθιστώντας την εξίσωση που περιγράφει το x1 σε αυτή έχουμε :

$$w.(x2 + 2\gamma * w/||w||) + b = 1 \quad (2)$$

Εφαρμόζοντας την επιμεριστική ιδιότητα, προκύπτει :

$$w.x2 + b + 2\gamma * (w.w)/||w||) = 1 \quad (3)$$

Όμως το x2 βρίσκεται πάνω στο hyperplane που περιγράφεται από την εξίσωση $w.x + b = -1$, επομένως οι 2 πρώτοι όροι της (3) ισούνται με -1. Έτσι αντικαθιστώντας το -1 στην (3) και διαιρώντας με 2 έχουμε:

$$\gamma * (w.w)/||w||) = 1 \qquad (4)$$

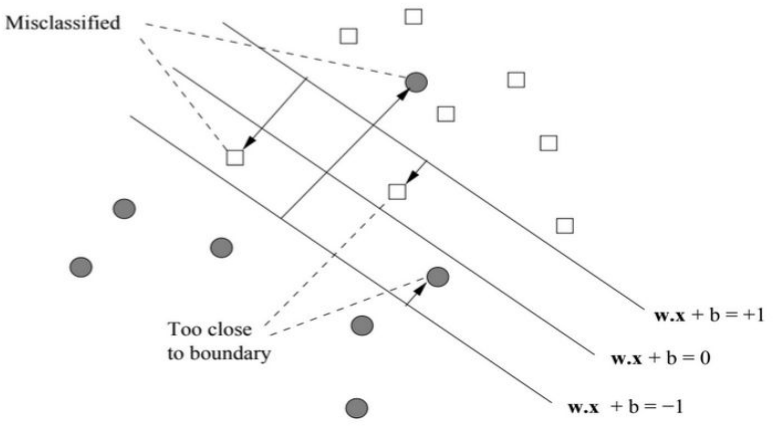
Ξέρουμε επίσης ότι $w.w = ||w||^2$ επομένως καταλήγουμε ότι $\gamma = 1/||w||$ (5). Η εξίσωση αυτή μας δίνει την δυνατότητα να αναθεωρήσουμε την αρχική μας υπόθεση. Μπορούμε αντί να μεγιστοποιήσουμε το γ να ελαχιστοποιήσουμε το w. Δηλαδή :

- Δοσμένου ενός training set (x1,y1),(x2,y2),...,(xn,yn), ελαχιστοποιήστε το w, υπό τον περιορισμό ότι για όλα τα $i = 1, 2, ..., n$ να ισχύει:

  $$yi(w.xi + b) \geqslant \gamma$$

3.1.3 Εύρεση βέλτιστου διαχωριστή

Σκοπός τώρα είναι να βρούμε το βέλτιστο hyperplane στην πιο γενική περίπτωση, όπου ανεξάρτητα σε ποιο hyperplane είμαστε κοντά, θα υπάρχουν σημεία στη λάθος πλευρά και μερικά στη σωστή αλλά πολύ κοντά στο διαχωριστικό hyperplane, με αποτέλεσμα να μην ικανοποιείται η απαιτούμενη συνθήκη για το γ.



Εικόνα 7: Σημεία που είναι λάθος κατηγοριοποιημένα ή πολύ κοντά στο διαχωριστικό hyperplane δημιουργούν σφάλμα

Στην παραπάνω εικόνα παρατηρούμε ότι δύο σημεία βρίσκονται στη λάθος πλευρά και δυο σημεία στη σωστή πλευρά αλλά πολύ κοντά στο διαχωριστικό hyperplane. Τα σημεία αυτά ονομάζονται bad points.

Κάθε ένα από τα σημεία αυτά δημιουργεί ένα σφάλμα όταν εκτιμούμε ένα πιθανό hyperplane. Ο αριθμός των σφαλμάτων, σε μονάδες που θα καθοριστούν ως μέρος της διαδικασίας βελτιστοποίησης, φαίνονται ως ένα βέλος που οδηγεί στο bad point και ξεκινάει από το hyperplane στη λάθος πλευρά του οποίου βρίσκεται το σημείο αυτό. Δηλαδή το βέλος μετράει την απόσταση από τα δύο παράλληλα hyperplane.

Θέλουμε λοιπόν τα σφάλματα να είναι όσο το δυνατόν λιγότερα και επίσης θέλουμε ένα w πολύ μικρό. Έτσι χρειαζόμαστε μια εξίσωση η οποία θα περιέχει έναν όρο με το άθροισμα των σφαλμάτων και έναν άλλο όρο που θα περιέχει το $||w||^2/2$.

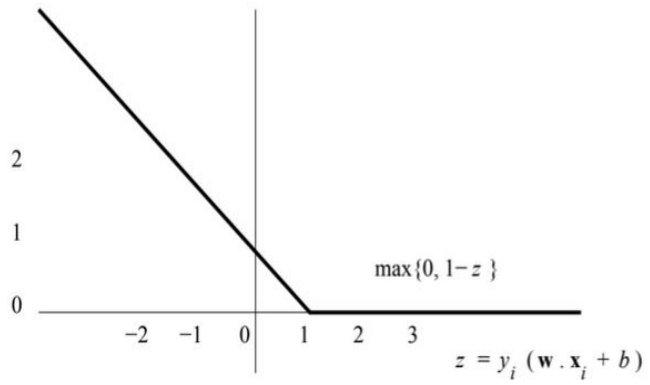
Είναι επιθυμητό να ελαχιστοποιήσουμε το $||w||^2/2$ διότι είναι το ίδιο με την ελαχιστοποίηση οποιασδήποτε μονότονης συνάρτησης του ||w||.

Δηλαδή, αν $w = [w1, w2, ..., wd]$ τότε το $||w||^2/2$ είναι $1/2\sum wi^2$ και η μερική παράγωγος του είναι wi. Αυτό μας βολεύει γιατί η παράγωγος του όρου είναι μια σταθερά ως προς κάθε σημείο του training set που ευθύνεται για το σφάλμα.

Έτσι πρέπει να σκεφτούμε πως θα ελαχιστοποιήσουμε την παρακάτω συνάρτηση :

$$f(\mathbf{w},b) = \frac{1}{2}\sum_{j=1}^{d} w_j^2 + C\sum_{i=1}^{n} \max\Big\{0,\ \ 1 - y_i\big(\sum_{j=1}^{d} w_j x_{ij} + b\big)\Big\}$$

Ο πρώτος όρος μιας βοηθά να έχουμε μικρά w. Ο δεύτερος όρος αναπαριστά τα σφάλματα από τα bad points, και περιέχει τη σταθερά C(regularization parameter) η οποία πρέπει να επιλεγεί σωστά. Αν επιλέξουμε μεγάλο C τότε θα έχουμε λίγα λάθος κατηγοριοποιημένα σημεία αλλά πολύ στενό margin. Αν επιλέξουμε μικρό C τότε έχουμε κάποια λάθος κατηγοριοποιημένα σημεία αλλά πολύ μεγάλο margin.



Εικόνα 8: Η συνάρτηση hinge μειώνεται γραμμικά και έπειτα μένει σταθερή στο 0

Η παραπάνω συνάρτηση περιγράφει τον δεύτερο όρο της συνάρτησης που θέλουμε να ελαχιστοποιήσουμε και μας δείχνει ότι ανάλογα με την τιμή του yi(θετική ή αρνητική) έχουμε :

$$\frac{\partial L}{\partial w_j} = \textbf{if } y_i(\sum_{j=1}^{d} w_j x_{ij} + b) \geq 1 \textbf{ then } 0 \textbf{ else } -y_i x_{ij}$$

3.1.4 SVM Λύσεις με Gradient Descent

Για να εφαρμόσουμε τον Gradient Descent υπολογίζουμε την παράγωγο της εξίσωσης ως προς w,b. Εφόσον θέλουμε να ελαχιστοποιήσουμε την $f(w,b)$ μετακινούμε τα b, w στην αντίθετη πλευρά από την πλευρά του gradient. Η ποσότητα που μετακινούμε είναι ανάλογη της παραγώγου σε αυτό το σημείο. Το πρώτο βήμα είναι να κάνουμε το b μέρος του διανύσματος w. Αυτό επιτυγχάνεται μόνο αν προσθέσουμε ένα επιπλέον στοιχείο με τιμή +1 σε κάθε feature διάνυσμα του training set .
Πριν εφαρμόσουμε τον αλγόριθμο θα πρέπει να επαναπροσδιορίσουμε την εξίσωση :

$$\frac{\partial f}{\partial w_j} = w_j + C\sum_{i=1}^{n}\Big(\textbf{if } y_i(\sum_{j=1}^{d} w_j x_{ij} + b) \geq 1 \textbf{ then } 0 \textbf{ else } -y_i x_{ij}\Big) \quad (12.6)$$

Για να εκτελέσουμε τον αλγόριθμο του gradient descent σε ένα training set επιλέγουμε :

- Τιμές για τις παραμέτρους C, n
- Αρχικές τιμές για το διάνυσμα w συμπεριλαμβανομένου του b μέσα σε αυτό

Έπειτα επαναληπτικά:

- Υπολογίζουμε τις μερικές παραγώγους της $f(w,b)$ ως προς τα w
- Ρυθμίζουμε τις τιμές του w αφαιρώντας ndf/dw για κάθε w

3.1.5 SVM Λύσεις με Kernels

Οι kernels[25] χρησιμοποιούνται όταν τα δεδομένα μας δεν είναι γραμμικά διαχωρίσιμα και μας επιτρέπουν να κάνουμε πιο αποδοτικούς υπολογισμούς μόνο αν ο αλγόριθμος κατηγοριοποίησης χρησιμοποιεί dot products.

Οι πιο γνωστοί Kernels που χρησιμοποιούνται είναι οι εξής :

- Polynomial kernel βαθμού p $k(x1, x2) = (x1^T x2 + 1)^p$
- Radial basis functions
  $k(x1, x2) = exp(-1/2\sigma^2 * ||x1 - x2||^2)$
- Two layer perceptron $k(x1, x2) = tanh(\beta o x1^T x2 + \beta 1)$

Έτσι η εξίσωση με τη χρήση των kernel μετασχηματίζεται :

$$q(x) = \Phi^\top(x) \underbrace{\sum_{i=1}^{L} \alpha_i y_i \Phi(x_i)}_{w \in \mathcal{F}} + b = \sum_{i=1}^{L} \alpha_i y_i \kappa(x, x_i) + b\,,$$

3.2 Naive Bayes

Σε αυτή την υποενότητα θα συζητηθεί πως θα κατηγοριοποιήσουμε διανύσματα από διακριτά χαρακτηριστικά $x \in \{1, ..., K\}^D$ όπου K είναι ο αριθμός των τιμών για κάθε feature και D είναι ο αριθμός των feature. Πρέπει λοιπόν να προσδιοριστεί η κατά συνθήκη κατανομή της κλάσης $p(x|y) = c$. Η πιο απλή προσέγγιση είναι να θεωρήσουμε ότι τα features είναι υπό όρους ανεξάρτητα από την κλάση. Έτσι μπορούμε να γράψουμε την πυκνότητα(density) της κλάσης ως ένα προϊόν μονοδιάστατων πυκνοτήτων :

$p(x|y = c, \theta) = \Pi p(xj|y = c, \theta c)$ για j=1,...,D (1)

Το μοντέλο αυτό ονομάζεται Naive Bayes Classifier(NBC)[26]. Το μοντέλο, παρόλο που δεν περιμένουμε τα features να είναι ανεξάρτητα ούτε κατά συνθήκη, φαίνεται να κατηγοριοποιεί σωστά τα δεδομένα. Αυτό συμβαίνει γιατί το μοντέλο είναι πολύ απλό και έτσι δεν μπορεί να προβεί σε overfitting.

Η εξίσωση (1) εξαρτάται από τον τύπο κάθε feature. Ας δούμε κάποια παραδείγματα παρακάτω 1:

- Στην περίπτωση όπου έχουμε feature πραγματικών τιμών χρησιμοποιούμε Γκαουσιανή κατανομή $p(x|y = c, \theta) = \Pi N(xj|\mu jc, \sigma jc^2)$ όπου j=1,...,D , μjc είναι η μέση τιμή του feature j στην κλάση c και σjc^2 είναι η διασπορά.
- Στην περίπτωση που έχουμε binary features $xj \in \{0, 1\}$ χρησιμοποιούμε Bernoulli κατανομή $p(x|y = c, \theta) = \Pi Ber(xj|\mu jc)$ όπου μjc είναι η πιθανότητα το feature j να ανήκει στην κλάση c.
- Στην περίπτωση που έχουμε categorical features $xj \in \{1,...,K\}$, χρησιμοποιούμε την κατανομή $p(x|y = c, \theta) = \prod Cat(xj|\mu jc)$ όπου μjc ένα ιστόγραμμα πάνω στις K πιθανές τιμές το xj να ανήκει στην κλάση c.

3.2.1 Εκπαίδευση ενός Naive Bayes Classifier

Για να εκπαιδεύσουμε ένα κατηγοριοποιητή Naive Bayes πρέπει να υπολογίσουμε τον εκτιμητή MLE ή MAP για τις παραμέτρους.

3.2.1.1 MLE για NBC

Η πιθανότητα για μια περίπτωση δεδομένων δίνεται από την εξίσωση :

$$p(\mathbf{x}_i, y_i|\boldsymbol{\theta}) = p(y_i|\boldsymbol{\pi}) \prod_j p(x_{ij}|\boldsymbol{\theta}_j) = \prod_c \pi_c^{\mathbb{I}(y_i=c)} \prod_j \prod_c p(x_{ij}|\boldsymbol{\theta}_{jc})^{\mathbb{I}(y_i=c)}$$

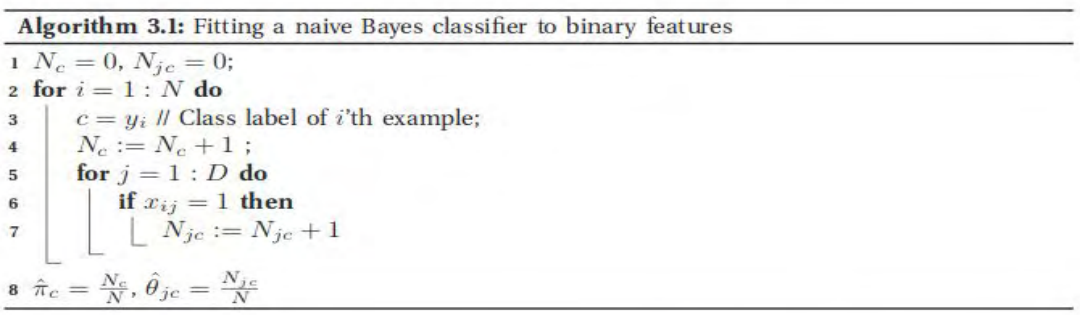
Έτσι το log-likelihood δίνεται από την εξίσωση :

$$\log p(\mathcal{D}|\boldsymbol{\theta}) = \sum_{c=1}^{C} N_c \log \pi_c + \sum_{j=1}^{D} \sum_{c=1}^{C} \sum_{i:y_i=c} \log p(x_{ij}|\boldsymbol{\theta}_{jc})$$

Παρατηρούμε ότι αυτή η έκφραση αποσυντίθεται σε μια σειρά από όρους. Ως εκ τούτου μπορούμε να βελτιστοποιήσουμε όλες αυτές τις παραμέτρους ξεχωριστά.

Ο MLE για την πιθανότητα της κλάσης δίνεται από την εξίσωση $\pi c = \mathrm{N}c/\mathrm{N}$ όπου $\mathrm{N}c \triangleq \Sigma \parallel (yi = c)$ είναι ο αριθμός παραδειγμάτων στην κλάση c.

Ο MLE για το likelihood εξαρτάται από τον τύπο της κατανομής που επιλέγουμε να χρησιμοποιήσουμε για κάθε feature. Για ευκολία ας υποθέσουμε ότι όλα τα features είναι binary $xj|y = c \sim Ber(\theta jc)$. Σε αυτή την περίπτωση ο MLE μετατρέπεται σε $\theta jc = Njc/Nc$. Η παρακάτω εικόνα περιγράφει πως δουλεύει ο αλγόριθμος για binary features.



Εικόνα 9: Περιγραφή αλγορίθμου Naive Bayes για binary χαρακτηριστικά

Αυτός ο αλγόριθμος παίρνει χρόνο O(CD) πολυπλοκότητας και είναι εύκολο να γενικευθεί για τη χρήση του σε mixed δεδομένα.

3.3 K-Nearest Neighbor

Ο αλγόριθμος K-Nearest Neighbor(KNN)[27] είναι η πιο απλή και γνωστή χρησιμοποιούμενη τεχνική κατηγοριοποίησης στη μηχανική μάθηση. Χρησιμοποιείται για αναγνώριση προτύπων για κατηγοριοποίηση και στηρίζεται στη χρήση μέτρων βασισμένων στην απόσταση. Για την κατηγοριοποίηση η είσοδος αποτελείται από τα k πλησιέστερα παραδείγματα εκπαίδευσης και η έξοδος εξαρτάται από την κατηγοριοποίηση.

Η κεντρική ιδέα είναι ότι ανάλογα με τα k κοντινότερα εκπαιδευόμενα δείγματα αποφασίζεται η κατηγορία που ανήκει ένα νέο μη εκπαιδευμένο παράδειγμα. Για να καθοριστεί ο αλγόριθμος χρειάζεται να αποφασιστεί :

- Η τιμή του k
- Ο τύπος της απόστασης που θα χρησιμοποιηθεί

Υπάρχουν πολλές διαφορετικές επιλογές για τον τύπο της απόστασης με κάποιες από αυτές να είναι οι εξής :

- Ευκλείδεια απόσταση : $\sqrt{\Sigma(xi - yi)^2}$

- Manhattan απόσταση: $\Sigma|xi - yi|$

- Minkowski απόσταση: $(\Sigma(|xi - yi|^q)^{1/q}$

Η πιο δημοφιλής επιλογή για να μετρήσουμε την απόσταση μεταξύ δύο σημείων είναι η ευκλείδεια απόσταση.

Όσον αφορά την τιμή του K πρέπει να επιλεγεί με στόχο την επίτευξη του βέλτιστου δυνατού μοντέλου για τα δεδομένα μας. Όταν η τιμή του K είναι μικρή, περιορίζουμε την περιοχή μιας δεδομένης πρόβλεψης και αναγκάζουμε τον κατηγοριοποιητή να παραβλέπει την συνολική κατανομή. Παρέχεται όμως μια πιο εύκαμπτη εφαρμογή και γραφικά το όριο απόφασης θα είναι πιο πριονωτή. Αντιθέτως, μια υψηλότερη τιμή του K δίνει περισσότερους γείτονες σε μια πρόβλεψη με κίνδυνο όμως να προκύψουν outliers. Γραφικά όμως το όριο απόφασης θα είναι πιο ομαλό.

Δοσμένου ενός θετικού ακεραίου K και ενός νέου παραδείγματος x0, ο κατηγοριοποιητής KNN πρώτα αναγνωρίζει τα K σημεία, που υπάρχουν στο training set, που βρίσκονται κοντά στο x0 και τα οποία αναπαριστώνται ως N0. Στη συνέχεια εκτιμάται η πιθανότητα τα σημεία αυτά να ανήκουν σε μια κατηγορία :
$Pr(Y = j|X = x0) = 1/K\sum I(yi = j)$ με $i \in N0$

Τέλος, το νέο παράδειγμα x0 κατηγοριοποιείται στην κατηγορία με τη μεγαλύτερη πιθανότητα. Παρακάτω δίνεται η αναλυτική περιγραφή του αλγορίθμου σε ψευδοκώδικα.

k-Nearest Neighbor
Classify ($\mathbf{X}, \mathbf{Y}, x$) // $\mathbf{X}$: training data, $\mathbf{Y}$: class labels of $\mathbf{X}$, x: unknown sample
for $i = 1$ **to** m **do**
  Compute distance $d(\mathbf{X}_i, x)$
end for
Compute set I containing indices for the k smallest distances $d(\mathbf{X}_i, x)$.
return majority label for $\{\mathbf{Y}_i$ where $i \in I\}$

Εικόνα 10: Αναλυτική περιγραφή KNN

Κεφάλαιο 4 Πειραματική ανάλυση και αποτελέσματα

4.1 Dataset

Για να εφαρμοστούν οι παραπάνω μέθοδοι επιλέχθηκε το UrbanSound8K dataset[28][29] το οποίο περιέχει 8732 labelled εγγραφές(4sec) περιβαλλοντικών ήχων δέκα κατηγοριών. Οι κατηγορίες αυτές είναι οι εξής :

- ήχος κλιματιστικού(air-conditioner)
- ήχος από γάβγισμα σκύλου(dog bark)
- ήχος παιδιών που παίζουν(children playing)
- ήχος από κόρνα αυτοκινήτου(car horn)
- ήχος από γεώτρηση (drilling)
- ήχος μηχανής(engine idling)
- ήχος πυροβολισμού(gun shot)
- ήχος πριονιού(jackhammer)
- ήχος σειρήνας (siren)
- ήχος μουσικής του δρόμου(street music)

Το dataset περιέχει επίσης ένα csv αρχείο με πληροφορίες για κάθε αρχείο ήχου που παρέχεται. Ας ρίξουμε μια πρώτη ματιά σε αυτό.

Out[5]:

0	100032-3-0-0.wav	100032	0.0	0.317551	1	5	3	dog_bark
2	100263-2-0-121.wav	100263	60.5	64.500000	1	5	2	children_playing
4	100263-2-0-137.wav	100263	68.5	72.500000	1	5	2	children_playing

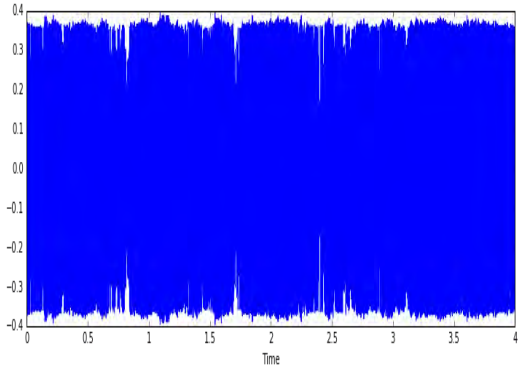
Εικόνα 11: Πρώτη ματιά στο πως μοιάζουν τα δεδομένα

Αφού είδαμε πως μοιάζουν τα δεδομένα πάμε τώρα να δούμε την συσχέτιση που έχουν μεταξύ τους τα features. Χρησιμοποιήθηκε για το σκοπό αυτό ο πίνακας συσχέτισης όπως παρέχεται από τη βιβλιοθήκη pandas. Ο πίνακας αυτός ανάλογα με τα χρώματα μας δείχνει και την συσχέτιση. Όσο πιο σκούρα είναι τα χρώματα τόσο μικρή εως και μηδαμινή είναι η συσχέτιση μεταξύ τους. Όσο πιο ανοιχτά τόσο ισχυρότερη.

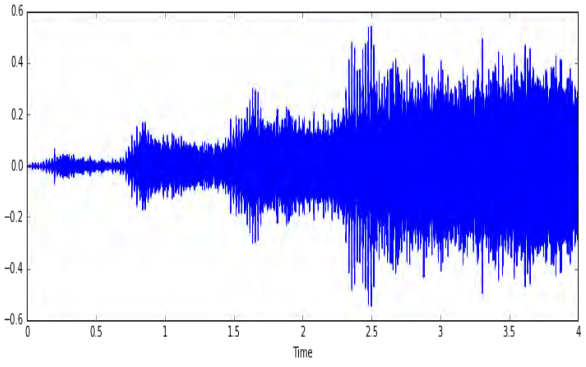
Παρατηρήθηκε λοιπόν ότι τα features δεν σχετίζονται μεταξύ τους και δεν έχουν να μας δώσουν καμία ενδιαφέρουσα πληροφορία. Επίσης πολλά από τα αρχεία ήχων ήταν κατεστραμμένα. Έτσι χρησιμοποιήθηκε ένα τροποποιημένο dataset [30] που περιέχει μόνο το όνομα του αρχείου και την κλάση που ανήκει.

Το επόμενο βήμα στην επεξεργασία των δεδομένων μας ήταν να δούμε πως μοιάζει η κυματομορφή της κάθε κλάσης. Αυτό μας ενδιαφέρει γιατί μπορεί με μια ματιά να μας δώσει να καταλάβουμε ποιες κατηγορίες θα είναι εύκολα αναγνωρίσιμες από το μοντέλο μας και ποιες όχι.

street music

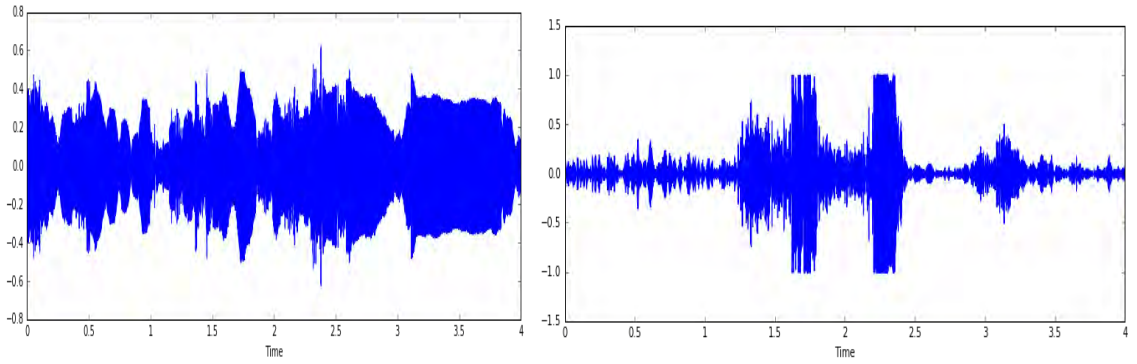


drilling

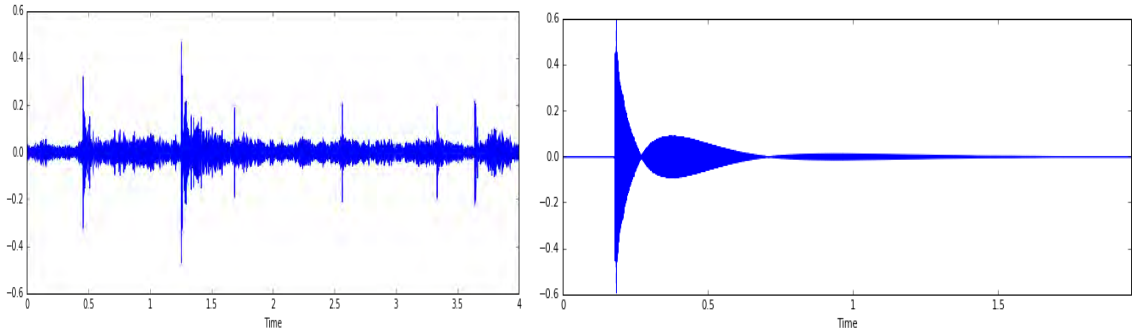


siren

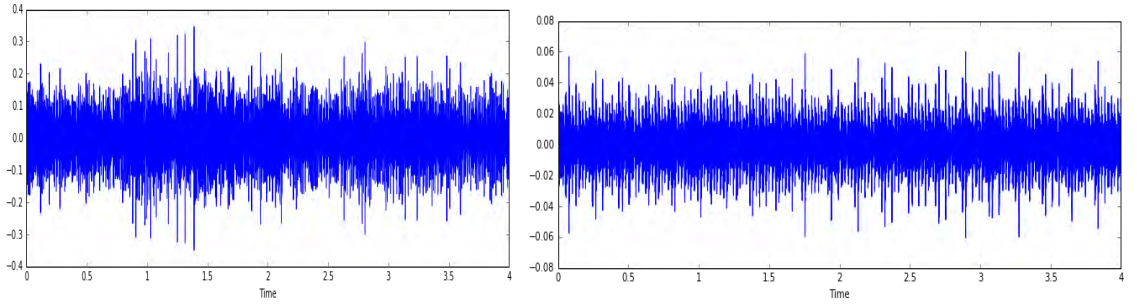
dog bark



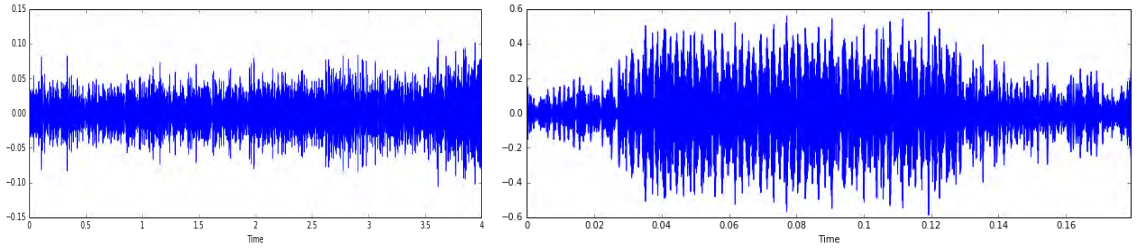
children playing gun shot



jackhammer engine idling



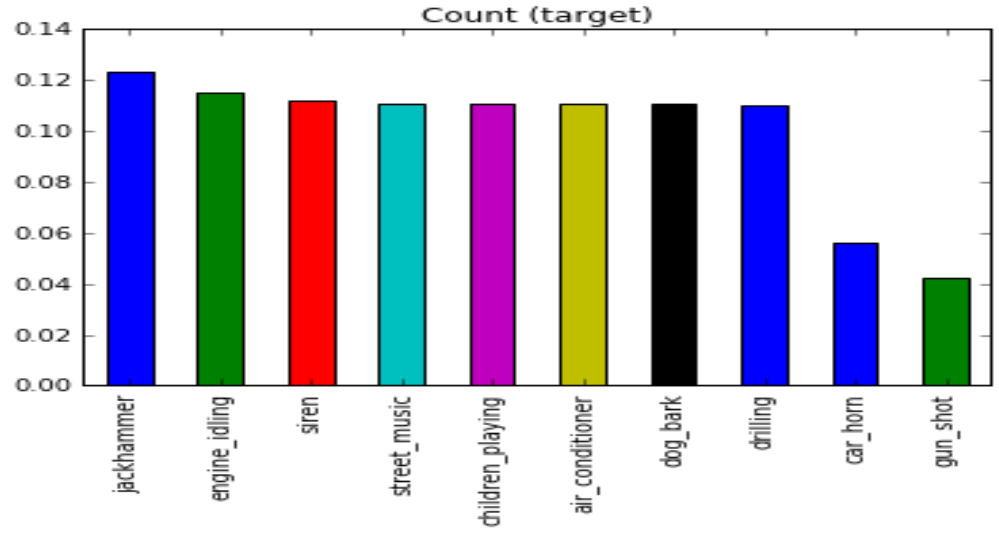
air conditioner car horn



Εικόνα 13: Μορφή κυματομορφών των κλάσεων

Όπως παρατηρούμε υπάρχουν κατηγορίες με μοναδική κυματομορφή όπως εκείνες του πυροβολισμού και της μουσικής του δρόμου. Υπάρχουν όμως άλλες που είναι δύσκολο να τις διαφοροποιήσουμε όπως την κυματομορφή του πριονιού από της γεώτρησης. Έτσι το μοντέλο μας είναι πιθανό να δυσκολευτεί να τις ξεχωρίσει και να κατατάξει σωστά τα δεδομένα σε αυτές.

Επόμενο βήμα είναι να ελέγξουμε για μη ισορροπία δειγμάτων ανάμεσα στις κλάσεις.



Εικόνα 14 :Έλεγχος για μη ισορροπία δειγμάτων

Παρατηρούμε λοιπόν ότι οι κατηγορίες car_horn και gun_shot έχουν λιγότερα δείγματα από τα υπόλοιπα που ίσως να μην είναι αρκετά για τη σωστή εκπαίδευση των μοντέλων μας.

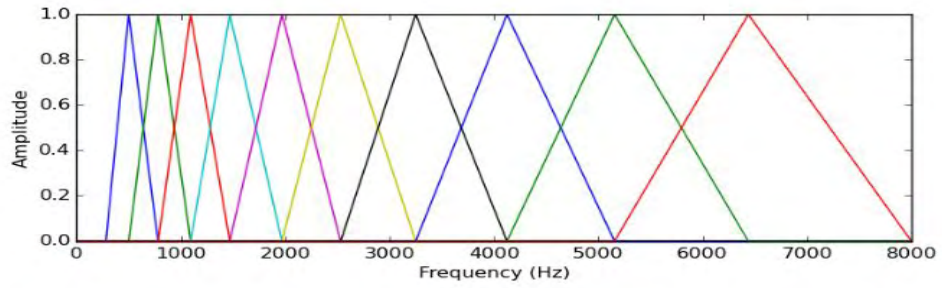
4.2 Feature Extraction

Έχοντας κάποιες σημαντικές πληροφορίες σχετικά με τα δείγματα μας, το επόμενο βήμα είναι να μετατρέψουμε τα αρχεία ήχων σε μια μορφή την οποία θα μπορούμε πιο εύκολα να τη διαχειριστούμε. Για αυτό το λόγο χρησιμοποιήθηκε η τεχνική Mel Frequency Cepstral

Coefficients(MFCC) η οποία εισήχθη πρώτη φορά το 1980 από τους Davis and Mermelstein[31] και αποτελεί state-of-art τεχνική ακόμα και σήμερα. Χρησιμοποιεί μη γραμμική κλίμακα συχνότητας με βάση την ακουστική αντίληψη. Βασίζεται στην mel κλίμακα, όπου mel είναι η μονάδα μέτρησης της συχνότητας ενός τόνου. Μια δημοφιλής εξίσωση για την μετατροπή της κλίμακας συχνότητας σε mel κλίμακα είναι η εξής :

$fmel = 1127ln(1 + fhz/700)$ (1)

Τα MFCCs συχνά υπολογίζονται χρησιμοποιώντας ένα σύνολο(filterbank) από M φίλτρα(m=0,1,...,M-1), το κάθε ένα από τα οποία έχει τριγωνική μορφή σε ομοιόμορφο επίπεδο στην mel κλίμακα.



Εικόνα 15: Σύνολο φίλτρων και η μορφή τους

Κάθε φίλτρο ορίζεται από την εξής συνάρτηση :

$$H_m[k] = \begin{cases} 0 & k < f[m-1] \\ \frac{k-f[m-1]}{f[m]-f[m-1]} & f[m-1] < k \leq f[m] \\ \frac{f[m+1]-k}{f[m+1]-f[m]} & f[m] \leq k < f[m+1] \\ 0 & k \geq f[m+1] \end{cases} \quad (2)$$

Δοσμένου του DFT(discrete fourier transformation) ενός σήματος εισόδου $x[k] = \Sigma x[n]e^{-j2\pi n/N}$ (3) όπου N το μέγεθος δειγματοληψίας του DFT, ας ορίσουμε fmin την ελάχιστη, fmax τη μέγιστη συχνότητα στο σύνολο των φίλτρων και Fs τη συχνότητα

δειγματοληψίας. M+2 οριακά σημεία f[m] είναι ομοιόμορφα κατανεμημένα μεταξύ της fmin και της fmax στην mel κλίμακα :
$f[m] = N/FsB^{-1}(B(fmin) + m[B(fmax) - B(fmin)]/(M + 1)$
όπου B είναι η μετατροπή της κλίμακας συχνότητας σε mel κλίμακα που δόθηκε στην εξίσωση (1) και B^{-1} είναι η αντίστροφη αυτής :
$fhz = 700(exp(fmel/1127) - 1)$

Το φάσμα πυκνότητας υπολογίζεται ως εξής :
$S[m] = ln[\Sigma|x[k]|^2 Hm[k]]$ (5) με m=0,1,...,M-1 και x[k] να είναι το αποτέλεσμα του DFT της εξίσωσης (3).

Η μέθοδος αυτή συνήθως υλοποιείται χρησιμοποιώντας DCT-II(discrete cosine transformation μορφής II) μιας και το $S[m]$ είναι ομοιόμορφο.
$x[n] = \Sigma S[m] cos[(m + 1/2)\pi n/M]$ (6) με n=0,1,...,M-1

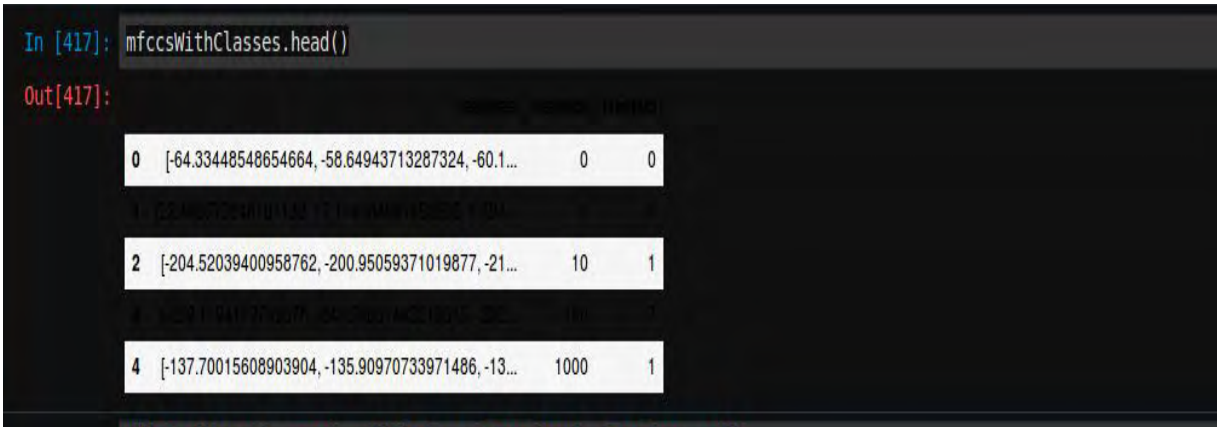
Συνήθως ο αριθμός των M φίλτρων κυμαίνεται από 20 εως 40 και ο αριθμός των coefficients που κρατάμε είναι 13. Αυτό γιατί η επιλογή μεγάλου αριθμού coefficients δίνει μεγαλύτερη πολυπλοκότητα στα μοντέλα. Οι χαμηλότεροι συντελεστές περιέχουν τις πληροφορίες σχετικά με το συνολικό φάσμα της συνάρτησης μεταφοράς. Ο συντελεστής μηδενικής τάξης δείχνει τη μέση ισχύ του σήματος εισόδου. Ο συντελεστής πρώτης τάξης δείχνει τη φασματική ενέργεια μεταξύ υψηλών και χαμηλών συχνοτήτων.

Με λίγα λόγια τα βήματα για την υλοποίηση της τεχνικής αυτής είναι τα εξής :

- Χωρισμός του σήματος σε μικρότερα frame
- Για κάθε frame υπολογισμός περιοδογράματος του φάσματος ισχύος

- Εφαρμογή του filterbank στο φάσμα ισχύος και υπολογισμός αθροίσματος της ενέργειας σε κάθε φίλτρο
- Υπολογισμός λογαρίθμου όλων των ενεργειών στο filterbank
- Υπολογισμός του DFT του λογαρίθμου όλων των ενεργειών
- Επιλογή των πρώτων 13 coefficients του DFT και απόρριψη των υπολοίπων

Έτσι λοιπόν με τη βοήθεια της βιβλιοθήκης librosa εφαρμόσαμε τη μέθοδο στα δεδομένα μας και τα αποτελέσματα είναι τα εξής :



Εικόνα 16: Μορφή των δεδομένων μετά το feature extraction

Το τελευταίο βήμα για να είναι έτοιμα τα δεδομένα μας να εφαρμοστούν στους αλγορίθμους είναι μετατροπή της ετικέτας κλάσης από string σε νούμερο και ο χωρισμός του dataset σε train set και test set με τη βοήθεια της train_test_split της βιβλιοθήκης scikit-learn της python. Το μέγεθος του train set είναι (2000,3588) .

4.3 Εφαρμογή SVM

Η πρώτη εφαρμογή των δεδομένων έγινε πάνω στον αλγόριθμο SVM με kernel rbf και τα αποτελέσματα που παίρνουμε είναι τα εξής :

- Accuracy : 0.11162079510703364
- F1 score : 0.11162079510703365

Ο επόμενος kernel που δοκιμάστηκε είναι ο sigmoid και τα αποτελέσματα που παίρνουμε είναι τα εξής :

- Accuracy : 0.10703363914373089
- F1 score : 0.10703363914373089

Ο τελευταίος kernel που δοκιμάστηκε είναι ο linear και τα αποτελέσματα που παίρνουμε είναι τα εξής :

- Accuracy : 0.5856269113149847
- F1 score : 0.5856269113149847

Confusion Matrix

57	2	1	3	4	0	1	4	0	0
5	28	3	10	12	0	11	14	2	2
2	2	46	1	2	0	5	4	8	1
12	6	3	20	7	0	4	3	0	0
6	21	16	4	21	0	11	3	3	0
0	0	0	0	0	0	0	0	0	0
7	5	10	1	1	1	31	12	6	1
1	6	3	3	9	0	23	53	4	0
5	6	11	1	2	0	6	7	41	0
2	3	4	0	0	0	1	0	3	15

Classification Report

	precision	recall	F1 score	support
0	0.59	0.79	0.67	72

1	0.35	0.32	0.34	87
2	0.47	0.65	0.55	71
3	0.47	0.36	0.41	55
4	0.36	0.25	0.29	85
5	0.00	0.00	0.00	0
6	0.33	0.41	0.37	75
7	0.53	0.52	0.52	102
8	0.61	0.52	0.56	79
9	0.79	0.54	0.64	28

Ο πρώτος πίνακας που βλέπουμε ονομάζεται confusion matrix και μας δείχνει την επίδοση του αλγορίθμου. Κάθε σειρά αναπαριστά χαρακτηριστικά που προβλέφθηκαν να ανήκουν σε μια κλάση και κάθε στήλη αναπαριστά σε ποια κλάση πραγματικά ανήκουν τα χαρακτηριστικά . Ο επόμενος μας δείχνει τις βασικές μετρήσεις της κατηγοριοποίησης μας. Σε αυτό το σημείο καλό θα ήταν να αναφέρουμε ακριβώς την έννοια της κάθε μετρικής.

- Precision : είναι ο λόγος των true positive προς των συνολικών προβλεπόμενων positive και δείχνει πόσο ακριβές είναι το μοντέλο ως προς αυτά

- Recall : είναι ο λόγος των true positive προς των συνολικών πραγματικών positive και δείχνει ποιο θα είναι το βέλτιστο μοντέλο

- F1-score : είναι ο μέσος όρος των precision, recall

- Accuracy : δείχνει το ποσοστό επιτυχίας σωστών προβλέψεων

4.4 Εφαρμογή KNN

Ο δεύτερος αλγόριθμος πάνω στον οποίο εφαρμόστηκαν τα δεδομένα μας είναι ο KNN με κ=3 και τα αποτελέσματα που πήραμε είναι τα εξής :

- Accuracy : 0.6804281345565749
- F1 score : 0.6804281345565749

4.5 Εφαρμογή Naive Bayes

Με την εφαρμογή του Naive bayes αλγορίθμου το μοντέλο μας δίνει τα παρακάτω αποτελέσματα :

- Accuracy : 0.4831804281345566
- F1 score : 0.4831804281345566

Κεφάλαιο 5 Συμπεράσματα και Μελλοντική Εργασία

5.1 Συμπεράσματα

Συγκεντρωτικά τα αποτελέσματα μας από τις τρεις τεχνικές επίλυσης που μελετήθηκαν είναι τα εξής :

	Επίδοση
SVM rbf kernel	0.11
SVM sigmoid kernel	0.10
SVM linear kernel	0.58
KNN	0.68
Naive Bayes	0.48

Παρατηρούμε λοιπόν ότι για τον αλγόριθμο των support vector machine οι δυο πρώτοι kernel δίνουν πολύ χαμηλή επίδοση στο μοντέλο με αποτέλεσμα να έχουμε underfitting, ενώ ο γραμμικός kernel δίνει ένα μέτριο μοντέλο κατηγοριοποίησης. Να αναφερθεί ότι όσο πιο κοντά στην τιμή 1 βρίσκεται η τιμή της επίδοσης, τόσο καλύτερο είναι το μοντέλο αυτό στην κατηγοριοποίηση των ήχων μας. Έτσι παρατηρούμε ότι το βέλτιστο μοντέλο που προκύπτει για την αναγνώριση και κατηγοριοποίηση περιβαλλοντικών ήχων.

5.2 Μελλοντική εργασία

Τα μοντέλα της μηχανικής μάθησης επιδέχονται σε πολύ υψηλές επιδόσεις που κυμαίνονται σε τιμές από 70% και άνω. Έτσι

καταλαβαίνουμε ότι το βέλτιστο μοντέλο μας επιδέχεται βελτίωση. Θα μπορούσαμε όμως να εξετάσουμε το ενδεχόμενο βελτίωσης και των υπόλοιπων τριών μοντέλων. Έτσι ο πρώτος μελλοντικά στόχος είναι η περαιτέρω έρευνα πάνω στις τεχνικές που παρέχονται ώστε να βελτιστοποιήσουμε τα μοντέλα μας καθώς και η δοκιμή νέων αλγορίθμων από τον τομέα του deep learning όπως είναι τα Recurrent Neural Network και τα Long Short Term Memory.

Όταν μελετηθούν και εφαρμοστούν όλα τα παραπάνω θα είμαστε έτοιμοι να παρουσιάσουμε ένα μοντέλο με πολύ υψηλότερη επίδοση και λιγότερες πιθανότητες λαθών. Στη συνέχεια, θα μπορούσαμε να μελετήσουμε την εφαρμογή του μοντέλου σε μια κινητή συσκευή ή ένα smart watch δημιουργώντας μια εφαρμογή. Η εφαρμογή θα διαθέτει το εκπαιδευμένο μοντέλο, θα αναγνωρίζει τους ήχους του περιβάλλοντος και θα τους στέλνει με τη μορφή μηνύματος στη συσκευή των χρηστών. Έτσι οι άνθρωποι που δεν ακούνε θα έχουν τη δυνατότητα να αντιλαμβάνονται τι διαδραματίζεται στον περίγυρό τους.

Βιβλιογραφία

[1] M.Ausloos, R. Lambiotte (2007). *Clusters or networks of economies? A macroeconomy study through Gross Domestic Product.* Physica A, 382, 16-21,https://doi.org/10.1016/j.physa.2007.02.005

[2] K. Kalpakis, D. Gada, V. Puttagunda (2001). *Distance measures for effective clustering of ARIMA time series*. Proceedings of IEEE International Conference on Data Mining, 10.1109/ICDM.2001.989529

[3] Anonymous URL. Retrieved 07/02/2019 https://machinelearningmastery.com/how-to-prepare-data-for-machine-learning/

[4] E. A. Maharaj, A. M. Alonso (2007). *Discrimination of locally stationary time series using wavelets*. Computational Statistics and Data Analysis, 52, 879-895, https://doi.org/10.1016/j.csda.2007.05.010

[5] C. Chatfield(1996). *The analysis of time series: An introduction,Sixth Edition* ,Chapman and Hall_CRC.

[6] A.M. Alonso, C. Garcia-Martos(2012). *Time series Analysis, Intergrated and long memory processes*.Retrieved 07/02/2019 from: http://www.etsii.upm.es/ingor/estadistica/Carol/TSAtema5petten.pdf

[7]Anonymous. Types of time series filter. Retrieved 07/02/2019 from: https://socialresearchmethods.net/kb/measlevl.php

[8] Andrew Ng. Machine Learning Course at Coursera .

[9] A. Li, C. Wang(2003). *An Industrial-Strength Audio Search Algorithm.*Retrieved 08/02/2019 from: http://www.ee.columbia.edu/~dpwe/papers/Wang03-shazam.pdf

[10] URL. Retrieved 10/02/2019 from: https://assistant.google.com/explore?hl=en_us

[11] E. Sadun, S. Sande(2011). *Talking to Siri: Learning the Language of Apple's Intelligent Assistant*, *2rd Edition*. Retrieved 10/02/2019 from: http://ptgmedia.pearsoncmg.com/images/9780789750693/samplepages/0789750694.pdf

[12] G. Lopez, L. Quesada, L.A. Guerrero(2017). *Alexa vs. Siri vs. Cortana vs. Google Assistant. A comparison of Speech-Based Natural User Interfaces. DOI:10.1007/978-3-319-60366-7_23*

[13]Anonymous. Retrieved 10/02/2019 from: https://techxplore.com/news/2017-11-google-pixel-buds-earphones-languages.html

[14]Anonymous. Retrieved 10/02/2019 from:https://link.springer.com/article/10.1186/s13634-015-0277-z#Sec5

[15] M. Jin ,Y. Song, I. McLoughlin, L. Rong-Dai(2017). *LID-Senones and Their Statistics for Language Identification*, IEEE 17410590,10.1109/TASLP.2017.2766023

[16] J. Glass(2007). *A brief introduction to Automatic Speech Recognition*, MIT Computer Science and Artificial Intelligence Laboratory. Retrieved 10/02/2019 from http://www.cs.columbia.edu/~mcollins/6864/slides/asr.pdf

[17]Anonymous.URL. Retrieved 11/02/2019 from:https://blog.algorithmia.com/introduction-natural-language-processing-nlp/

[18] S. King. *A beginner's guide to statistical parametric speech synthesis*, University of Edinburgh. Retrieved 11/02/2019 from: http://www.cstr.ed.ac.uk/downloads/publications/2010/king_hmm_tutorial.pdf

[19] L.R. Rabiner(1989). *A tutorial on hidden Markov models and selected applications in speech recognition*, IEEE 3424909, 10.1109/5.18626

[20] Y. Miao, M. Gowayyed, F. Metze(2015). *EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding*, IEEE 15789169,10.1109/ASRU.2015.7404790

[21] T. Winograd(1972). *Understanding natural language*, https://doi.org/10.1016/0010-0285(72)90002-3

[22] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa(2011), *Natural Language Processing (Almost) from Scratch*. Retrieved 11/02/2019 from: http://www.jmlr.org/papers/volume12/collobert11a/collobert11a.pdf

[23] Braci. URLRetrieved 11/02/2019 from: http://www.braci.co/

[24] S. Chu, S. Narayanan, C. Jay Kuo(2009). *Environmental Sound Recognition With Time–Frequency Audio Features*, IEEE Signal Processing Society, 10.1109/TASL.2009.2017438

[25] J. Leskovec, A. Rajaraman, J.D. Ullman(2014). *Mining of massive datasets*, Chapter 12 Large Scale Machine Learning p.p 479-490

[26] V. Hlavac. *Nonlinear classifiers, kernel methods and SVM* ,Czech Technical University in Prague.Retrieved 12/02/2019 from:http://people.ciirc.cvut.cz/~hlavac/TeachPresEn/31PattRecog/35KernelMeth-and-SVM.pdf

[27] K.P Murphy. *Machine learning: a probabilistic perspective*. MIT Press, 2012. Retrieved 12/02/2019 from: https://doc.lagout.org/science/Artificial%20Intelligence/Machine%20learning/Machine%20Learning_%20A%20Probabilistic%20Perspective%20%5BMurphy%202012-08-24%5D.pdf

[28] G. James, D.Witten, T. Hastie, R. Tibshirani. *An Introduction to Statistical Learning with Applications in R*.Retrieved 12/02/2019 from: http://www-bcf.usc.edu/~gareth/ISL/ISLR%20Seventh%20Printing.pdf

[29] Urban Sound Datasets. Retrieved 10/09/2018 from: https://urbansounddataset.weebly.com/urbansound8k.html

[30] J. Salamon, C. Jacoby, J.P. Bello(2014). A Dataset and Taxonomy for Urban Sound Research. Retrieved 10/09/2018 from: http://www.justinsalamon.com/uploads/4/3/9/4/4394963/salamon_urbansound_acmmm14.pdf

[31]Urban Sound Classification, Kaggle. Retrieved 13/09/2018 from: https://www.kaggle.com/pavansanagapati/urban-sound-classification

[32] S.B. Davis, P. Mermelstein(1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences, IEEE Trans. Acoust Speech Signal Processing 28(4), 357-366,10.1109/TASSP.1980.1163420

[33] Anonymous. MFCC tutorial. Retrieved 07/11/2018 from: http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/

[34] URL https://www.wikipedia.org/