ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Αλγόριθμοι Classification σε Big Data

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ελένη Κ. Τσιλιγιάννη

Επιβλέπων : Ιωάννης Βασιλείου
Καθηγητής Ε.Μ.Π.

Αθήνα, Σεπτέμβριος 2015

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Αλγόριθμοι Classification σε Big Data

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ελένη Κ. Τσιλιγιάννη

Επιβλέπων : Ιωάννης Βασιλείου
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή στις 17/09/2015

.....	.....	.....
Ι. Βασιλείου	Κ. Κοντογιάννης	Ν. Κοζύρης
Καθηγητής Ε.Μ.Π.	Αν. Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Σεπτέμβριος 2015

.....

Ελένη Κ. Τσιλιγιάννη

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Ευχαριστίες,

στην οικογένεια και στους φίλους μου για την υποστήριξη, σε όλα τα επίπεδα, στην ολοκλήρωση των ακαδημαϊκών μου στόχων,
στον Ντίνο Αρκουμάνη για τις συμβουλές και την καθοδήγησή του ως άμεσος επιβλέπων κατά τη διάρκεια της διπλωματικής μου εργασίας,
στον καθηγητή Βασιλείου Ιωάννη για την υποστήριξή του και την καθοδήγησή του τόσο ως καθηγητής στο προπτυχιακό μου στάδιο όσο και ως επιβλέπων στην διπλωματική μου εργασία.

Περίληψη

Στην παρούσα εργασία θα μελετήσουμε τις επιδόσεις δύο γνωστών αλγορίθμων που χρησιμοποιούνται για κατηγοριοποίηση κειμένου, του Naïve Bayes και των κ-εγγύτερων γειτόνων. Οι αλγόριθμοι αυτοί θα εφαρμοστούν πάνω σε μεγάλο όγκο δεδομένων που αντιπροσωπεύουν προϊόντα ηλεκτρονικών καταστημάτων. Η κατηγοριοποίηση βασίζεται στα χαρακτηριστικά των προϊόντων όπως αυτά προκύπτουν από τη γενικότερη περιγραφή. Μέσω κάποιων πειραμάτων προσπαθούμε να εξηγήσουμε τη συμπεριφορά των χρησιμοποιούμενων αλγορίθμων και τους τρόπους που θα μπορούσαν να βελτιώσουν την αποτελεσματικότητά τους.

Λέξεις κλειδιά: <<Κατηγοριοποίηση, Μηχανική Μάθηση, Mahout, Naïve Bayes, κ-εγγύτεροι γείτονες, ευρετηριοποίηση, Solr ,Big Data >>.

Abstract

The purpose of the present diploma thesis is study and performance evaluation of two well-known text classification algorithms, Naïve Bayes and k-Nearest-Neighbors. We will use these algorithms on big data that represent products retrieved from online shops. Classification is based on the characteristics of the products as resulting from the general description. Through some experiments we are trying to explain the behavior of the algorithms used and figure out some ways that they could improve their effectiveness.

Key words: <<Classification, Machine Learning, Mahout, Naïve Bayes k-Nearest-Neighbours, indexing, Solr, Big Data >>.

Πίνακας Περιεχομένων

ΚΕΦΑΛΑΙΟ 1

ΚΕΦΑΛΑΙΟ 2

ΚΕΦΑΛΑΙΟ 3

ΚΕΦΑΛΑΙΟ 4

ΚΕΦΑΛΑΙΟ 5

ΚΕΦΑΛΑΙΟ 6

ΚΕΦΑΛΑΙΟ 1

Μηχανική Μάθηση (Machine Learning)

1.1 Γενικά περί Μηχανικής Μάθησης

Η Μηχανική Μάθηση αποτελεί έναν επιστημονικό κλάδο που διερευνά την κατασκευή και τη μελέτη αλγορίθμων που μπορούν να "μάθουν" μέσω της τροφοδότησής τους από δεδομένα. Αυτοί οι αλγόριθμοι κατασκευάζουν μοντέλα που χρησιμοποιούν τα δεδομένα εισόδου για να κάνουν προβλέψεις ή να πάρουν αποφάσεις, αντί απλώς να ακολουθούν ρητά κάποιες εντολές προγράμματος. Η Μηχανική μάθηση μπορεί να θεωρηθεί ένα υποπεδίο της επιστήμης των υπολογιστών και της στατιστικής ανάλυσης. Συνδέεται στενά με την τεχνητή νοημοσύνη και τα προβλήματα βελτιστοποίησης από όπου αντλούνται μέθοδοι, θεωρίες και εφαρμογές, ενώ χρησιμοποιείται σε μια σειρά υπολογιστικών εργασιών όπου ο σχεδιασμός και ο προγραμματισμός αλγορίθμων που στηρίζονται σε κανόνες είναι ανέφικτος. Παράδειγματα εφαρμογών είναι το φιλτράρισμα ανεπιθύμητης αλληλογραφίας, η οπτική αναγνώριση χαρακτήρων (OCR), οι μηχανές αναζήτησης και η όραση υπολογιστών. Συχνά η Μηχανική μάθηση συγχέεται με την εξόρυξη δεδομένων, παρόλο που εστιάζει περισσότερο στη διερευνητική ανάλυση των δεδομένων. Η Μηχανική Μάθηση και η αναγνώριση προτύπων μπορούν να θεωρηθούν ως δύο όψεις του ίδιου τομέα.

1.2 Ορισμοί για τη Μηχανική Μάθηση

Τις τελευταίες έξι δεκαετίες έχουν δωθεί αρκετοί ορισμοί για τη Μηχανική Μάθηση από διάφορους πρωτοπόρους της βιομηχανίας που δούλεψαν ώστε να μας οδηγήσουν προς τη σωστή κατεύθυνση.

Το 1950 ο Alan Turing στη δημοσίευσή του "Computing Machinery and Intelligence" αναρωτήθηκε εάν "οι υπολογιστές είναι δυνατό να σκέφτονται". Η δημοσίευση περγράφει ένα παιχνίδι μίμησης στο οποίο συμμετέχουν 3 διαγωνιζόμενοι – ένας άνθρωπος που λειτουργεί ως κριτής ,ένας άλλος άνθρωπος κι ένας υπολογιστής που προσπαθεί να πείσει τον κριτή ότι είναι κι αυτός άνθρωπος. Ο κριτής μπορούσε να πληκτρολογήσει σε ένα τερματικό για να επικοινωνήσει με τους άλλους δύο συμμετέχοντες. Τόσο ο άλλος άνθρωπος όσο και ο υπολογιστής μπορούσαν να αποκριθούν και στο τέλος ο κριτής θα έπρεπε να αποφανθεί ποια απάντηση προήλθε από τον υπολογιστή. Στην περίπτωση που ο κριτής συστηματικά αποτύγχανε να βρει τη διαφορά μεταξύ των απαντήσεων του ανθρώπου και του υπολογιστή τότε ο υπολογιστής νικούσε το παιχνίδι.

Το πείραμα αυτό συνεχίζεται ακόμα και σήμερα με τη μορφή του βραβείου Loebner, ενός διαγωνισμού που διεξάγεται κάθε χρόνο στο πεδίο της Τεχνητής Νοημοσύνης. Ο σκοπός είναι το ίδιο απλός: να πειστούν οι κριτές ότι μιλούν με ανθρώπινο ον αντί με ένα υπολογιστικό σύστημα που μπορεί και στέλνει μηνύματα.

Το 1959 ο Arthur Samuel όρισε τη Μηχανική Μάθηση ως "Ένα πεδίο μελέτης που δίνει τη δυνατότητα στους υπολογιστές να μαθαίνουν χωρίς να ακολουθούν ρητές εντολές προγράμματος". Ο Samuel, κατά τη διάρκεια που εργαζόταν για την IBM του πιστώθηκε η δημιουργία ενός προγράμματος που μαθαίνει από μόνο του. Ο ίδιος επέλεξε ένα απλό παιχνίδι ,που ονομάζεται checkers , μέσω του οποίου το πρόγραμμα μπορεί να μάθει ,επειδή είναι απλό αλλά χειάζεται και στρατηγική. Με τη βοήθεια του αλγορίθμου alpha-beta με αποτίμηση-κλάδεμα(μείωση των κόμβων που χρειάζονται να αποτιμηθούν) και του αλγορίθμου minimax(ελαχιστοποίηση της απώλειας για τη χειρότερη περίπτωση) το πρόγραμμα μπορούσε να μειώσει τις κινήσεις ,συνεπώς να βελτιώσει την επίδοση λόγω κόστους σε μνήμη του προγράμματος.

Ο Tom M. Mitchell, επικεφαλής στον τομέα της Μηχανικής Μάθησης στο πανεπιστήμιο Carnegie Mellon και συγγραφέας του βιβλίου Machine Learning (McGraw-Hill) το 1997 έδωσε τον εξής ορισμό: "Ένα πρόγραμμα υποτίθεται ότι μαθαίνει από μια εμπειρία Ε, σύμφωνα με μια κλάση έργων Τ και ένα μέτρο της επίδοσής του Ρ, εάν η επίδοσή του στο έργο Τ, όπως έχει μετρηθεί από το Ρ, βελτιώνει την εμπειρία του κατά Ε ". Το σημαντικό σε αυτή την προσέγγιση είναι ότι έχουμε συγκεκριμένους όρους για να προσδιοριστεί η έννοια της Μηχανικής Μάθησης:

- Έργο (Τ), που μπορεί να είναι ένα η περισσότερα στον αριθμό
- Εμπειρία (Ε)
- Επίδοση (Ρ)

Έτσι όταν ένας υπολογιστής τρέχει μια σειρά από διαδικασίες η εμπειρία που θα αποκομίσει θα πρέπει να οδηγήσει σε αύξηση της απόδοσής του.

Η Μηχανική Μάθηση, με ένα συνολικό ορισμό, είναι ένας κλάδος της Τεχνητής Νοημοσύνης όπου χρησιμοποιώντας υπολογιστές σχεδιάζουμε συστήματα ικανά να μαθαίνουν μέσω των δεδομένων από τα οποία εκπαιδεύονται. Τα συστήματα μαθαίνουν και βελτιώνονται με βάση την εμπειρία και το πέρασμα του χρόνου οπότε τελικά κατασκευάζεται ένα μοντέλο που μπορεί να προβλέψει και να δώσει απάντηση σε ερωτήσεις στηριζόμενο στην προηγούμενη γνώση του.

1.3 Κατηγορίες προβλημάτων

Το πλήθος των αλγορίθμων που μπορούν να χρησιμοποιηθούν στα προβλήματα Μηχανικής Μάθησης είναι μεγάλο. Το είδος του προβλήματος και η απαιτούμενη έξοδος είναι οι παράγοντες που θα καθορίσουν ποιος αλγόριθμος τελικά θα χρησιμοποιηθεί. Τα προβλήματα κατηγοριοποιούνται τυπικά σε τρεις μεγάλες κατηγορίες μάθησης ανάλογα με το σήμα ή την ανατροφοδότηση του συστήματος: την επιβλεπόμενη μάθηση, τη μάθηση χωρίς επίβλεψη και την ενισχυτική μάθηση.

1.3.1 Επιβλεπόμενη μάθηση

Στην κατηγορία αυτή δίνονται στο πρόγραμμα κάποια παραδείγματα εισόδων και τις αντίστοιχες επιθυμητές εξόδους (ουσιασιαστικά αποτελούν ένα ζεύγος-διάνυσμα) από κάποιον "δάσκαλο" και ο σκοπός είναι να δημιουργηθεί ένας γενικός κανόνας που αντιστοιχίζει μελλοντικές εισόδους σε εξόδους. Οι αλγόριθμοι που χρησιμοποιούνται αναλύουν τα δεδομένα της εκπαίδευση του προγράμματος και προσπαθούν να συμπεράνουν κάποια συνάρτηση που θα χρησιμοποιηθεί στην αντιστοίχηση των μελλοντικών εισόδων και εξόδων.

Ιδανικό σενάριο θα επέτρεπε στον αλγόριθμο να προσδιορίσει σωστά την ταμπέλα ή κλάση που ανήκουν τα επόμενα άγνωστα στιγμιότυπα. Αυτό προϋποθέτει έναν αλγόριθμο

που γενικεύει από τα δεδομένα εκπαίδευσης σε άγνωστες καταστάσεις με κάποιο λογικό τρόπο.

Δύο υποκατηγορίες της επιβλεπόμενης μάθησης είναι η Κατηγοριοποίηση και η Παλινδρόμηση. Στην Κατηγοριοποίηση θα πρέπει να αναγνωριστεί σε ποια από τις διαθέσιμες κατηγορίες ανήκει μια καινούργια παρατήρηση με βάση φυσικά ένα σετ από δεδομένα εκπαίδευσης των οποίων οι κλάσεις ή κατηγορίες είναι γνωστές. Το πλήθος των κατηγοριών είναι μικρό και μάλιστα διακριτό, ενώ σε πολλές από τις περιπτώσεις δεν ξεπερνά τις δύο.

Από την άλλη στην Παλινδρόμηση προσπαθούμε να βρούμε κάποια εξίσωση ώστε να ταιριάξει με τα δεδομένα που έχουμε κι έτσι να γίνει πρόβλεψη για την τιμή των επόμενων εξόδων. Η πιο απλή μορφή της εξίσωσης είναι η γραμμική (y=m*x+b) στην οποία καθορίζονται οι τιμές m και b και με δεδομένη την είσοδο x προβλέπουμε την τιμή του y. Πιο προχωρημένες τεχνικές επιτρέπουν τη χρήση παραπάνω από μιας τιμής εισόδου x ώστε να γίνει ταίριασμα σε πιο πολύπλοκα μοντέλα όπως γίνεται με τετραγωνικές εξισώσεις. Στην Παλινδρόμηση οι τιμές της εξόδου είναι πολλές στον αριθμό και μπορεί να πάρουν οποιαδήποτε τιμή.

1.3.2 Μη επιβλεπόμενη μάθηση

Στη Μηχανική Μάθηση το πρόβλημα της Μη επιβλεπόμενης μάθησης είναι ότι προσπαθούμε να βρούμε το πως είναι δομημένα τα δεδομένα που έχουμε τα οποία όμως δεν ανήκουν σε κάποια κλάση. Το γεγονός λοιπόν ότι δεν έχουν ταμπέλα σημαίνει ότι δεν υπάρχει σφάλμα ή κάποια ένδειξη που θα αξιολογήσει κάπως μια πιθανή λύση. Σε αυτή την κατηγορία εντάσσονται και άλλες τεχνικές που βρίσκουν,συνοψίζουν και εξηγούν κάποια χαρακτηριστικά-κλειδιά των δεδομένων. Πολλές από αυτές τις μεθόδους ταυτίζονται με τις μεθόδους που χρησιμοποιούνται κατά την προεπεξεργασία των δεδομένων στον τομέα της εξόρυξης δεδομένων.

Υποκατηγορία της Μη επιβλεπόμενης μάθησης είναι η Συσταδοποίηση(Clustering). Πρόκειται για την ομαδοποίηση ανάμεσα σε ένα σύνολο αντικειμένων με τέτοιο τρόπο ώστε τα αντικείμενα που ανήκουν στην ίδια ομάδα (συστάδα) παρουσιάζουν μεγαλύτερη ομοιότητα μεταξύ τους παρά με οποιοδήποτε άλλο αντικείμενο που ανήκει σε διαφορετική συστάδα.Η Συσταδοποίηση από μόνη της δεν αποτελεί κάποιον αλγόριθμο αλλά τη γενική ιδέα του προβλήματος που πρέπει να επιλυθεί. Αυτό μπορεί να επιτευχθεί από διαφορετικούς αλγορίθμους που διαφέρουν αρκετά στο σκεπτικό μεταξύ τους ως προς το τι αποτελεί μια συστάδα και πώς μπορεί να βρεθεί μια τέτοια ,με αποδοτικό τρόπο.

Δημοφιλείς έννοιες συστάδων περιλαμβάνουν ομάδες με μικρές μεταξύ τους αποστάσεις, πυκνές περιοχές του χώρου των δεδομένων ή και κάποια συγκεκριμένη στατιστική κατανομή. Επιπλέον η συσταδοποίηση μπορεί να θεωρηθεί ένα πρόβλημα βελτιστοποίησης με πολλές παραμέτρους που πρέπει να ληφθούν υπόψιν. Κατά τη διαδικασία δοκιμής ή και αποτυχίας είναι συχνά απαραίτητο να τροποποιήσουμε τον τρόπο που προεπεξεργαζόμαστε τα δεδομένα και να βάλουμε παραμέτρους έτσι ώστε το αποτέλεσμα να επιτύχει τα επιθυμητά χαρακτηριστικά.

Εκτός από τον όρο “Συσταδοποίηση” υπάρχουν αρκετοί όροι με παρόμοια έννοια όπως αυτόματη κατηγοριοποίηση, αριθμητική ταξινόμηση και τυπολογική ανάλυση.

1.3.3 Ενισχυτική μάθηση

Η τρίτη κατηγορία έχει επηρρεαστεί από την επιστήμη της ψυχολογίας και εστιάζει στο πώς κάποιοι ευφυείς πράκτορες λειτουργούν σε ένα περιβάλλον έτσι ώστε να

μεγιστοποιήσουν μια έννοια συσωρευμένης ανταμοιβής. Λόγω ότι το πρόβλημα είναι τόσο γενικό έχει μελετηθεί από διάφορους επιστημονικούς κλάδους όπως η θεωρία παιγνίων, η θεωρία ελέγχου, η στατιστική , οι γενετικοί αλγόριθμοι και η βελτιστοποίηση που βασίζεται σε προσομοίωση. Στον τομέα του αυτόματου ελέγχου οι μέθοδοι που χρησιμοποιούνται και μελετώνται ονομάζονται δυναμικοί-προσεγγιστικοί(approximate dynamic programming).

Για τη Μηχανική Μάθηση οι περισσότεροι αλγόριθμοι που βρίσκουν εφαρμογή σε προβλήματα Ενισχυτικής μάθησης χρησιμοποιούν τεχνικές δυναμικού προγραμματισμού. Οι κλασσικοί αλγόριθμοι εδώ δε μπορούν να εφαρμοστούν διότι το περιβάλλον αποτελείται από καταστάσεις που οδηγούν σε μερικώς τυχαία αποτελέσματα ,τα οποία εν μέρει επηρρεάζονται από τη διαδικασία λήψης της απόφασης. Συνεπώς η γνώση για το περιβάλλον για τους αλγόριθμους που χρησιμοποιούνται στην Ενισχυτική μάθηση δεν είναι αναγκαία.

1.4 Τομείς που χρησιμοποιούν Μηχανική Μάθηση

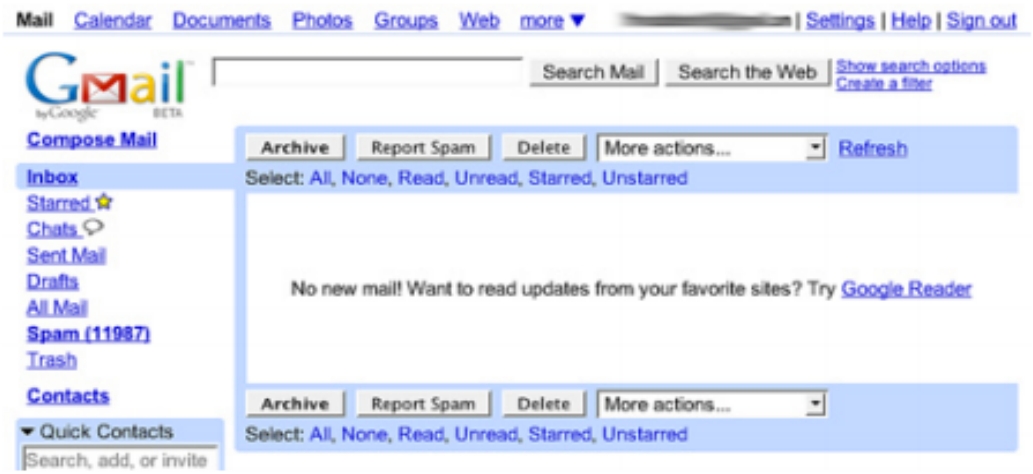
Η Μηχανική Μάθηση βρίσκει μεγάλη εφαρμογή σε επίλυση προβλημάτων που απασχολούν την καθημερινή μας ζωή όπως:

- **Λογισμικό**

Σκοπός της χρήσης είναι η βελτίωση της εμπειρίας έτσι ώστε το σύστημα να μαθαίνει μέσα από τη συμπεριφορά του χρήστη και τις προτιμήσεις του. Έπειτα από κάποιο χρόνο που το σύστημα χρησιμοποιείται καθίσταται ικανό να μαντέψει τις επόμενες επιλογές του χρήστη.

Ανεπιθύμητη αλληλογραφία

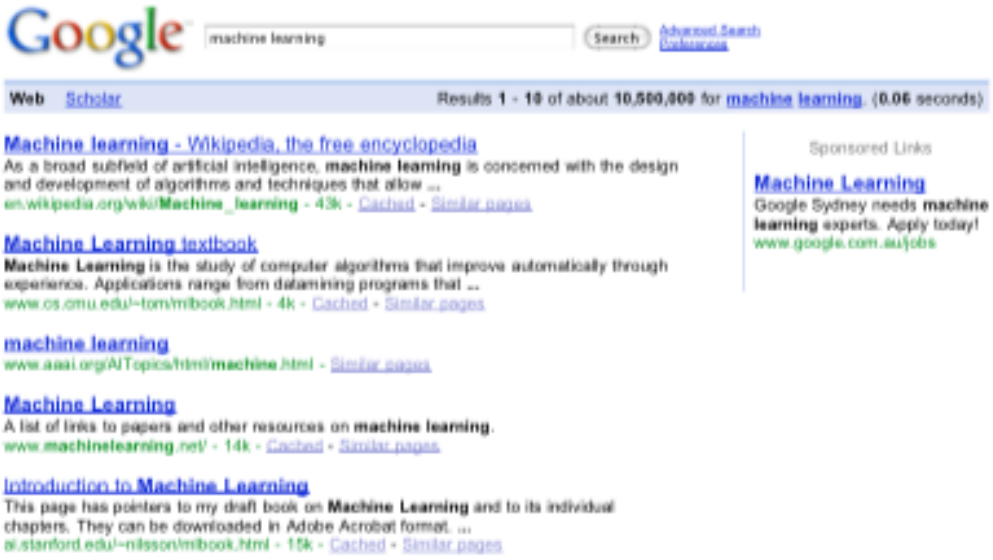
Η αναγνώριση της ανεπιθύμητης αλληλογραφίας είναι ένα κλασσικό παράδειγμα όπου με το πέρασμα του χρόνου οι υπολογιστές ανταποκρίνονται όλο και καλύτερα. Σε περίπτωση που το πρόγραμμα θεωρήσει ανεπιθύμητο ένα μήνυμα αλληλογραφίας ρωτά το χρήστη να επιβεβαιώσει ή να απορρίψει αυτή τη θεώρηση και με βάση την ενέργειά του υπάρχει μεγάλη πιθανότητα τα επόμενα μηνύματα να τα διαχειριστεί σωστά από εδώ και στο εξής.



Εικόνα 1.1. Παράδειγμα εφαρμογής ανεπιθύμητης αλληλογραφίας

Κατάταξη αποτελεσμάτων σε μηχανές αναζήτησης

Όλοι είμαστε εξοικειωμένοι με την κατάταξη των αποτελεσμάτων μετά από μια αναζήτηση στο διαδίκτυο. Η διαδικασία υποβολής ενός ερωτήματος προς μια μηχανή αναζήτησης μας επιστρέφει σελίδες σχετικές με το ερώτημα με μια συγκεκριμένη σειρά σχετικότητας. Για να επιτευχθεί αυτή η ταξινόμηση στα αποτελέσματα η μηχανή αναζήτησης χρειάζεται να "ξέρει" ποιες σελίδες είναι σχετικές και ποιες ταιριάζουν με το ερώτημα. Αυτή η γνώση μπορεί να αντληθεί απο διάφορες πηγές όπως η δομή του συνδέσμου που οδηγεί στη σελίδα, το περιεχόμενό της, τη συχνότητα που οι χρήστες επιλέγουν να ακολουθήσουν μια συγκεκριμένη πρόταση από τα αποτελέσματα. Έτσι με χρήση της Μηχανικής Μάθησης η διαδικασία αυτοματοποιείται και τελικά κατασκευάζουμε πιο έξυπνες μηχανές αναζήτησης που ανταποκρίνονται στις ανάγκες των χρηστών.



Εικόνα 1.2. Παράδειγμα των 5 πρώτων αποτελεσμάτων αναζήτησης του όρου "machine learning" στη μηχανή αναζήτησης Google

Συνεργατικό φιλτράρισμα/Συστάσεις

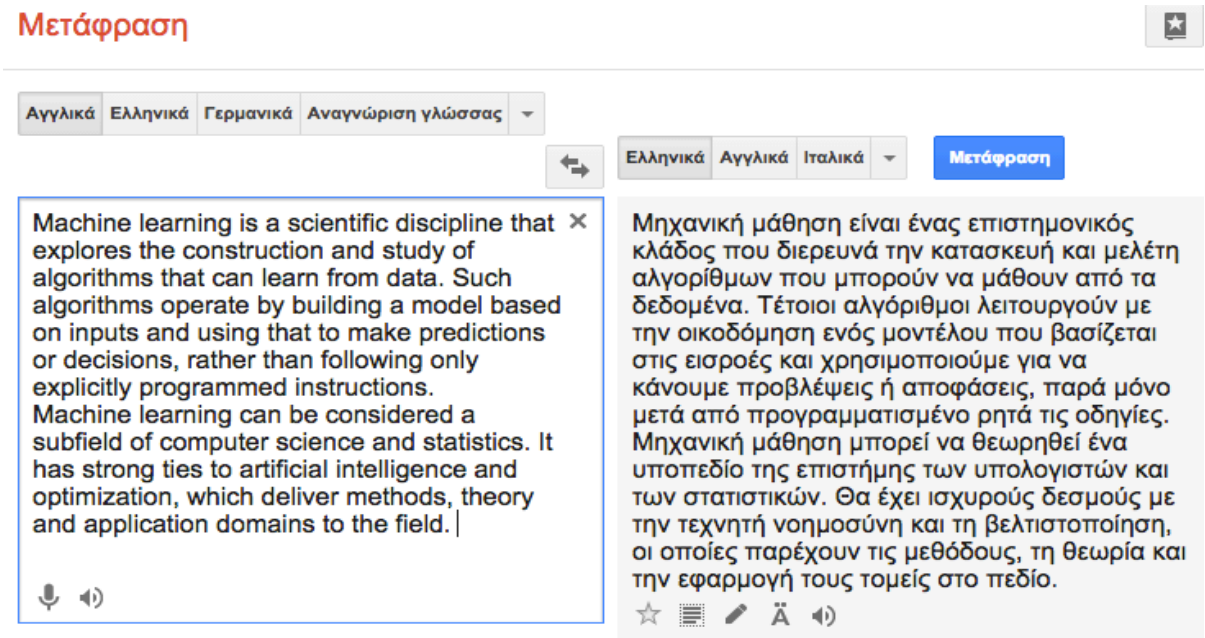
Μια ακόμη σχετική εφαρμογή είναι το συνεργατικό φιλτράρισμα που συνηθίζεται κυρίως σε ηλεκτρονικά μαγαζιά όπως το Amazon ,σε σελίδες ενοικίασης βίντεο ή ταινιών όπως το Netflix όπου χρησιμοποιούνται οι πληροφορίες που παρέχονται από τους χρήστες ,με την αναζήτηση συγκεκριμένων αντικειμένων,με τη βαθμολόγηση ή την αγορά αυτών, για να πεισουν τους καταναλωτές να αγοράσουν επιπλέον προϊόντα που προτείνονται ως σχετικά.

Το πρόβλημα μοιάζει αρκετά κι εδώ με την κατάταξη των αποτελεσμάτων στις μηχανές αναζήτησης. Όπως ακριβώς και πριν θέλουμε μια ταξινομημένη λίστα από αποτελέσματα μόνο που η διαφορά βρίσκεται ότι εδώ δεν θέσαμε συγκεκριμένο ερώτημα, παρά μόνο μπορούμε να χρησιμοποιήσουμε προηγούμενη συμπεριφορά του χρήστη(αγορές, προτιμήσεις,συνήθειες) ώστε να του προτείνουμε σχετικά αντικείμενα. Μια άλλη διαφορά είναι ότι βασιζόμαστε στην ομοιότητα των επιλογών των χρηστών για αυτό και η συνεργατική φύση της διαδικασίας.

Εικόνα 1.3. Παράδειγμα προτάσεων αγοράς βιβλίων σχετικών με το βιβλίο “Machine Learning” του Tom M. Mitchell στη σελίδα του Amazon

Αυτόματη μετάφραση και αναγνώριση λέξεων σε διάφορες γλώσσες

Ένα μη επαρκώς ορισμένο πρόβλημα που καλείται να λύσει η Μηχανική Μάθηση είναι η αυτόματη μετάφραση εγγράφων. Θα στοχεύαμε ιδανικά στο να καταλαβαίνουμε πλήρως ένα κείμενο πρωτού το μεταφράσουμε, χρησιμοποιώντας μια ομάδα από κανόνες που θα είχαν εξαχθεί από γλωσσολογική ανάλυση των δύο γλωσσών από και προς τις οποίες θέλουμε να μεταφράσουμε. Αυτή είναι μια επίπονη διαδικασία ,πολύ περισσότερο όταν το δοσμένο κείμενο δεν είναι συντακτικά σωστό ή υπάρχει δυσκολία στην κατανόηση μέρους αυτού. Μια τακτική που έχει αποδειχθεί επιτυχημένη στην πράξη είναι η χρήση παραδειγμάτων μεταφρασμένων κειμένων που υποδεικνύουν πως πρέπει να γίνει η μετάφραση.



Εικόνα 1.4. Παράδειγμα μετάφρασης τμήματος κειμένου από αγγλικά σε ελληνικά

Αναγνώριση προτύπων

Πολλές εφαρμογές που αφορούν την ασφάλεια όπως ο έλεγχος πρόσβασης σε ένα κτίριο χρησιμοποιούν την αναγνώριση προσώπων.Με δεδομένο λοιππόν μια φωτογραφία

ή ενός βίντεο του προσώπου καλούνται να αναγνωρίσουν την ταυτότητα του ατόμου. Με άλλα λόγια το σύστημα θα πρέπει να κατηγοριοποιήσει τα πρόσωπα σε μια ή περισσότερες κλάσεις ή να αποφανθεί ότι το συγκεκριμένο πρόσωπο είναι άγνωστο. Ένα παρόμοιο αλλά με κάποιες διαφορές είναι το πρόβλημα της επαλήθευσης. Ο σκοπός είναι να επαληθευτεί εάν ένα πρόσωπο είναι αυτό που λέει ότι είναι. Να σημειωθεί ότι η διαφορά με πριν έγγειται στο ότι η απάντηση είναι ναι ή όχι. Για να αντιμετωπίσουμε τις διαφορετικές συνθήκες φωτισμού, τις εκφράσεις του προσώπου, εάν το άτομο φοράει γυαλιά, το μήκος των μαλλιών κλπ. Είναι επιθυμητό να έχουμε ένα σύστημα που μαθαίνει ποια χαρακτηριστικά είναι σχετικά ώστε να αναγνωριστεί η ταυτότητα του συγκεκριμένου ατόμου.

Εικόνα 1.5. Παράδειγμα του ίδιου προσώπου. Η πρόκληση είναι να αναγνωριστεί πως πρόκειται για το ίδιο άτομο και στις 11 περιπτώσεις.

Εκτός από την αναγνώριση προσώπων η Μηχανική Μάθηση βοηθά και στην αναγνώριση οντοτήτων (named entity recognition), ένα πρόβλημα που απαιτεί την αναγνώριση κάποιας οντότητας όπως μια τοποθεσία, έναν τίτλο, ένα όνομα, μιας ενέργειας κλπ από έγγραφα. Τέτοια βήματα είναι καθοριστικά για την αυτόματη αναγνώριση εικόνας από κείμενα. Κάποια προγράμματα ηλεκτρονική αλληλογραφίας στην εποχή μας έχουν την ικανότητα να αναγνωρίζουν αυτόματα κάποιες διευθύνσεις στα μηνύματα αλληλογραφίας και να τις προσθέτουν σε ένα αρχείο διευθύνσεων.

```
HAVANA (Reuters) - The European Union's top development aid official
left Cuba on Sunday convinced that EU diplomatic sanctions against
the communist island should be dropped after Fidel Castro's
retirement, his main aide said.

<TYPE="ORGANIZATION">HAVANA</> (<TYPE="ORGANIZATION">Reuters</>) - The
<TYPE="ORGANIZATION">European Union</>'s top development aid official left
<TYPE="ORGANIZATION">Cuba</> on Sunday convinced that EU diplomatic sanctions
against the communist <TYPE="LOCATION">island</> should be dropped after
<TYPE="PERSON">Fidel Castro</>'s retirement, his main aide said.
```

Εικόνα 1.6. Παράδειγμα οντοτήτων σε άρθρο.Οι σχετικές τοποθεσίες, οργανισμοί και πρόσωπα έχουν ταμπέλα για εξαγωγή περισσότερων δεδομένων

Άλλες εφαρμογές περιλαμβάνουν την αναγνώριση φωνής και λόγου, όπως για παράδειγμα το Siri της Apple όπου ο χρήστης ρωτά το κινητό του τηλέφωνο κι αυτό επεξεργάζεται τα δεδομένα από το λόγο του.Το αποτέλεσμα μπορεί να είναι ένα γραπτό μήνυμα ή ένα μήνυμα στο Twitter ή η οργάνωση μιας συνάντησης. Εάν η εφαρμογή δεν μπορέσει να καταλάβει πραγματοποιεί μια αναζήτηση βασισμένη στα όσα κατέγραψε από το χρήστη.

- **Οικονομία – Χρηματιστήριο**

Υπάρχουν πολλές πλατφόρμες που στοχεύουν στο να βοηθήσουν τους χρήστες να κάνουν πιο επιτυχημένες συναλλαγές. Αυτές οι πλατφόρμες χρειάζεται να κάνουν πολύ

μεγάλη ανάλυση και υπολογισμούς για να είναι σε θέση να κάνουν προτάσεις και να δώσουν συστάσεις. Από τη σκοπιά της Μηχανικής Μάθησης οι αποφάσεις παίρνονται για λογαριασμό του χρήστη για το αν είναι συμφέρον να αγοράσει ή να πουλήσει μετοχές στην παρούσα τιμή. Λαμβάνονται υπ΄όψιν το ιστορικό, οι τιμές που άνοιξαν και έκλεισαν οι μετοχές αλλά και οι ποσότητα που αγοράζεται και πωλείται. Με δεδομένους αυτούς τους 4 παράγοντες ένας αλγόριθμος μπορεί να μάθει τις "συνήθειες" του χρηματιστηρίου οπότε εφαρμόζοντας αυτές τις αρχές στο σύνολο των μετοχών ενός χρήστη μπορεί να αποφανθεί εάν συμφέρει να πουλήσει ή να αγοράσει.

Τα Bitcoins είναι ένα καλό παράδειγμα όπου χρησιμοποιούνται τέτοιου είδους αλγόριθμοι στην πράξη. Τα εικονικά νομίσματα αγοράζονται και πωλούνται με βάση την τιμή όπου η αγορά είναι διαθέσιμη να πληρώσει και την τιμή που οι ιδιοκτήτες επιθυμούν να πουλήσουν. Μεγάλη σημασία έχει ο χρόνος υπολογισμού και η ικανότητα να γίνονται πολλές χιλιάδες συναλλαγές το δευτερόλεπτο που στηρίζονται στην πρόβλεψη που κάνουν οι αλγόριθμοι. Διακινείται τεράστιο ποσό χρημάτων και καθυστέρηση της τάξης των χιλιοστών του δευτερολέπτου μπορεί να οδηγήσει σε απώλεια εκατομμυρίων εάν δεν πραγματοποιηθούν εγκαίρως οι συναλλαγές. Σήμερα περίπου το 70% των συναλλαγών πραγματοποιούνται από μηχανές, αντί για ανθρώπους.

➢ Ρομποτική

Με χρήση της Μηχανική Μάθησης τα ρομπότ μπορούν να αποκτήσουν ικανότητες προσαρμογής στο περιβάλλον που παράγουν έργο. Για παράδειγμα μπορούν να τοποθετούν αντικείμενα στη σωστή θέση καθώς και να μετακινούνται είτε από μόνα τους είτε με ανθρώπινη παρέμβαση. Με την εξέλιξη της τεχνολογία των αισθητήρων πολλοί αλγόριθμοι χρησιμοποιούνται ως εξωτερικοί μηχανισμοί και μπορούν να στέλνουν ,να λαμβάνουν σήματα και να επικοινωνούν με τα ρομπότ.

➢ Υγεία – Ιατρική

Ένας μεγάλος αριθμός νεοφυών επιχειρήσεων επικεντρώνεται στα πλεονεκτήματα που παρέχουν οι αλγόριθμοι Μηχανικής Μάθησης σε συνεργασία με τη μεγάλη ποσότητα δεδομένων για να προσφέρουν στους επαγγελματίες υγείας να είναι καλύτερα ενημερωμένοι έτσι ώστε να παίρνουν καλύτερες αποφάσεις για τους ασθενείς τους. Ο Watson,ο γνωστός υπολογιστής που είχε κατασκευάσει η IBM και είχε νικήσει δύο διαγωνιζόμενους στο τελεοπτικό παιχνίδι ερωτήσεων Jeopardy βρίσκεται στην υπηρεσία της υγείας. Χρησιμοποιώντας τον σαν υπηρεσία στον νέφος οι γιατροί μπορούν να έχουν πρόσβαση και να πληροφορούνται από εκατομμύρια σελίδες ερευνητικού περιεχομένου και πολλές ακόμη χιλιάδες πληροφορίες από ιατρικά στοιχεία.

Με τον αριθμό των καταναλωτών που χρησιμοποιούν έξυπνα τηλέφωνα να έχει αυξηθεί και με τις συσκευές να μπορούν να καταγράψουν και να επεξεργαστούν μετρήσεις όπως το βάρος, ο καρδιακός ρυθμός, η πίεση του αίματος, ακόμη και τα επίπεδα γλυκόζης στο αίμα είναι πλέον εύκολο να ελέγχεται η κατάσταση τους υγείας κάποιου και να κρατιέται λεπτομερές ιστορικό. Μάλιστα μπορεί μέσω της συσκευής να προταθούν κάποιες εναλλακτικές στην αγωγή με χρήση φυσικά αλγορίθμων μάθησης.

Παρά το γεγονός ότι είναι εύκολο να αναλύσουμε τα δεδομένα που αφορούν την υγεία, η προστασία των προσωπικών δεδομένων είναι ένα άλλο θέμα. Προφανώς κάποιοι χρήστες να ενδιαφέρονται παραπάνω για τον τρόπο που τα δεδομένα τους χρησιμοποιούνται και ειδκά όταν αυτά πωλούνται σε τρίτους. Τελικά λοιπόν ενώ τα δεδομένα ολοένα αυξάνονται σε ποσότητα η συζήτηση για την ιδιωτικότητα μπορεί να επηρρεάσει τον τρόπο με τον οποίο τελικά οι αλγόριθμοι θα χρησιμοποιηθούν.

➢ Διαφήμιση

Ανέκαθεν οι εταιρείες προσπαθούσαν να επηρρεάσουν το καταναλωτικό κοινό με σκοπό να αγοράσουν τα προϊόντα που παρήγαγαν. Από το 1995 και μετά το διαδίκτυο έχει δώσει τη δυνατότητα σε όλες τις επιχειρήσεις να διαφημίζουν τα προϊόντα τους χωρίς να χρειάζονται την τηλεόραση ή τον τύπο. Μάλιστα είχε προκληθεί μεγάλη συζήτηση όταν ανακαλύφθηκε η προοπτική ενός υπολογιστή ο οποίος μπορεί να καταγράψει μέσω των cookies τις καθημερινές μας συνήθειες κι έτσι ακολούθησε μεγάλος αγώνας ννα απενεργοποιηθεί από τους πλοηγούς.

Τα αρχεία καταγραφής είναι μια ακόμη τακτική που οι διαφημιστές μπορούν να δουν τις προτιμήσεις του κοινού. Η ομαδοποίηση των χρηστών σύμφωνα με το ενδιαφέρον τους για συγκεκριμένο τύπο προϊόντων πραγματοποιείται και μέσω κινητών τηλεφώνων που στέλνουν με σήμα την τοποθεσία του χρήστη οπότε λαμβάνει εξαιρετικά στοχευμένες διαφημίσεις.Παλιότερα αυτού του είδους η διαφήμιση θεωρούται εισβολή στη ιδιωτικότητα του ατόμου, σταδιακά όμως η ιδέα έγινε συνήθεια και το ίδιο το άτομο δηλώνει από μόνο του την τοποθεσία του στα κοινωνικά δίκτυα.

➢ Ηλεκτρονικό Εμπόριο

Η Μηχανική Μάθηση χρησιμοποιείται σε μεγάλο βαθμό στο εμπόριο και κυριώς στο ηλεκτρονικό. Μεγάλης κλίμακας παράδειγμα είναι η κάρτες “αφοσίωσης” που δίνονται στους πελάτες κάποιων καταστημάτων. Από αυτές οι εταιρείες μπορούν να λάβουν πολλά δεδομένα τα οποία μεταφράζονται σε πληροφορία για τους καταναλωτές κι έτσι μπορούν να σχεδιάσουν τη στρατηγική τους στην αγορά. Επιπλέον από την καταγραφή των επισκέψεων είτε στο φυσικό κατάστημα είτε στο ηλεκτρονικό μπορεί να γίνει στοχευμένη διαφήμιση με αποστολή προσφορών μέσω ηλεκτρονικού ταχυδρομείου έτσι ώστε να αυξηθούν οι πωλήσεις. Παράδειγμα, που είναι αρκετά διαδεδομένο, και προκύπτει από ανάλυση αγορών με χρήση της κάρτας είναι η ταυτόχρονη αγορά χαρτικών ειδών και μπυρας από μεγάλο πλήθος καταναλωτών που έχει οδηγήσει τους εμπόρους να τοποθετούν αυτά τα δύο προϊόντα σχετικά κοντά το ένα με το άλλο.

➢ Ηλεκτρονικά Παιχνίδια

Η ψυχαγωγία και συγκεκριμένα τα ηλεκτρονικά παιχνίδια είναι ιδνικά για την εφαρμογή Μηχανικής Μάθησης. Οι εταιρείες έχουν καταναλώσει πολύ χρόνο και χρήμα στη μελέτη των κινήσεψν ενός παίχτη κάτω από δεδομένες συνθήκες για να συμπεράνουν αν ο συγκεκριμένους παίχτης έκλεψε στο παιχνίδι. Σε άλλα παιχνίδια γίνεται δειγματολειψία για κάποιες παρτίδες έτσι ώστε ο υπολογιστής να εκπαιδευτεί και να προσαρμόσει το επίπεδο στα χαρακτηριστικά του παίχτη. Πλέον ηλεκτρονικά παιχνίδια κατασκευάζονται και για τις φορητές συσκευές, κινητά τηλέφωνα και tablets, με τρόπο ώστε να προκαλούν ένα είδος εθισμού για αρκετό καιρό και να δημιουργούν ευχαρίστηση στο χρήστη.

➢ Διαδίκτυο των αντικειμένων (Internet of Things)

Διασυνδεδεμένες συσκευές που μπορούν να ανταλλάξουν κάθε τύπου δεδομένα μεταξύ τους έχουν ήδη κατακλείσει την αγορά. Εξαιτίας του χαμηλού κόστους παραγωγής και διανομής η επικοινωνία μεταξύ συσκευών διαδόθηκε τόσο στο σπίτι όσο και στη βιομηχανία. Πρόκειται για αυτοματισμούς αυτοματισμούς, έξυπνες αγορές αλλά και πρωτότυποι τρόποι μέτρησης της ενεργειακής κατανάλωσης. Τα δεδομένα που εξάγονται είναι τόσα πολλά που η Μηχανική Μάθηση μπορεί να βρει πρόσφορο έδαφος από την έξοδο που παράγουν οι συσκευές αυτές. Αν και είναι μια περίοδος σε αρχικό στάδιο για τη

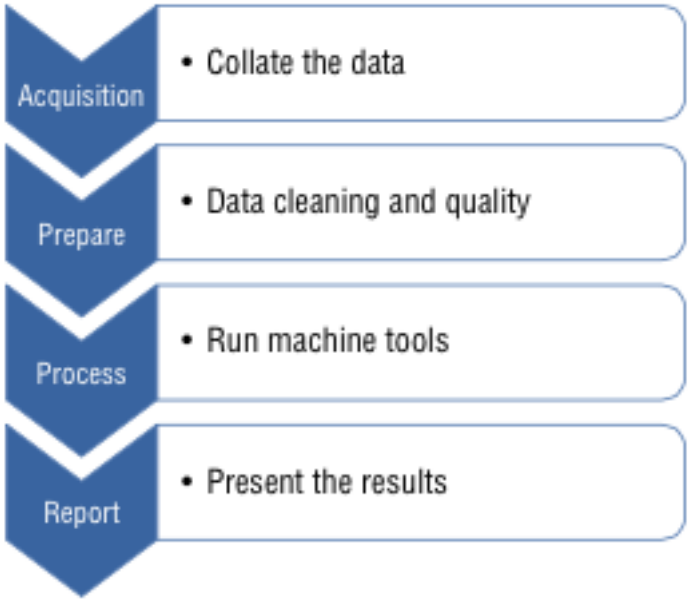
διασύνδεση των αντικειμένων παρόλα αυτά γίνεται τεράστια προσπάθεια με θεαματικά αποτελέσματα. Σε αυτό συμβάλλουν κάποιες ηλεκτρονικές πλακέτες όπως το Arduino και το Raspberry Pi οι οποίες ως φθηνά υπολογιστικά συστήματα μπορούν να χρησιμοποιηθούν για τη μέτρηση της θερμοκρασίας, την ανίχνευση ήχου και κίνησης και να εξαχθούν χρήσιμα συμπεράσματα σε πραγματικό χρόνο.

ΚΕΦΑΛΑΙΟ 2

Αλγόριθμοι Κατηγοριοποίησης (Classification)

2.1 Μοντέλα Μάθησης

Ένα πρόβλημα Μηχανικής Μάθησης περιλαμβάνει ουσιαστικά ένα κύκλο ενεργειών που χρειάζεται να γίνουν όπως φαινεται παρακάτω:



Εικόνα 2.1. Διαδικασία που ακολουθείται σε προβλήματα Μηχανικής Μάθησης

Τα δεδομένα μπορούν να αποκτηθούν από πολλές πηγές, είτε από διάφορους ιδιωτικούς οργανισμούς είτε ελεύθερα προσβάσιμα στο διαδίκτυο. Στη συνέχεια θα πρέπει να γίνει διαλογή ώστε να καθαριστούν ώστε να μπορούν να αξιολογηθούν σωστά, πριν υποστούν οποιαδήποτε επεξεργασία. Η φάση της επεξεργασίας είναι και η πιο σημαντική αφού ουσιαστικά εδώ μέσω κάποιων αλγορίθμων θα αναλυθούν και θα προκύψουν τα συμπεράσματα. Τέλος τα αποτελέσματα θα πρέπει να παρασταθούν είτε με διαγράμματα είτε με γραπτές αναφορές.

Οι αλγόριθμοι εκμάθησης, με πολλούς από τους οποίους θα ασχοληθούμε σε αυτό το κεφάλαιο, κατασκευάζονται με βάση το πρόβλημα που αντιμετωπίζουμε και την προσέγγιση που επιλέγουμε. Γενικά όμως η επιλογή του μοντέλου που θα χρησιμοποιηθεί θα πρέπει να είναι ξεκάθαρο στα παρακάτω σημεία:

- Μαθητευόμενος

Στην περίπτωσή μας ο μαθητευόμενος είναι ένας αλγόριθμος ή αλλιώς ένα πρόγραμμα υπολογιστή. Οι αλγόριθμοι εκμάθησης μπορούν να ενσωματωθούν σε πιο γενικευμένα συστήματα λογισμικού όπως για παράδειγμα οι ευφυείς πράκτορες. Επιπλέον μπορούν να ενσωματωθούν σε φυσικά αντικείμενα ,όπως σε ένα ρομπότ, είτε σε ad-hoc δίκτυα επεξεργαστών σε ευφυή υπολογιστικά περιβάλλοντα.

- Πεδίο ορισμού

Ουσιαστικά τι ακριβώς θα περιλαμβάνει η εκμάθηση. Μπορεί να περιγράφεται από μια συνάρτηση ή από ένα γενικότερο concept. Το τι περιλαμβάνει η εκμάθηση μπορεί να είναι,μεταξύ άλλων, η λειτουργία μιας συσκευής, ένα παιχνίδι, μια γλώσσα ,μια προτίμηση κλπ.

- Στόχος

Ο στόχος περιγράφει για ποιο λόγο λαμβάνει χώρα η εκμάθηση. Μπορεί να είναι η εξαγωγή κανόνων για αναγώριση πλαστών δεδομένων, η δημιουργία ενός προσομοιωτή μεγάλης ακρίβειας για φυσικά φαινόμενα ή για τον έλεγχο ενός συστήματος.

- Αναπαράσταση

Ο τρόπος με τον οποίο τα αντικείμενα προς εκμάθηση αναπαριστώνται από τον υπολογιστή. Η υπόθεση που αναπτύσσει το πρόγραμμα κατά τη διάρκεια της διαδικασίας μπορεί να παρασταθεί με τον ίδιο τρόπο ή με πιο αυστηρές/χαλαρές συνθήκες.

- Τεχνολογία όσον αφορά τους αλγόριθμους

Το καθαρά τεχνικό κομμάτι περιλαμβάνει το framework που θα χρησιμοποιηθεί. Μεταξύ των πολλών διαφορετικών τεχνολογιών μπορεί να είναι: τα νευρωνικά δίκτυα, τα δίκτυα πεποίθησης, τα δέντρα αποφάσεων, οι γραμματικές, ο λογισμός που βασίζεται σε περιπτώσεις, τα πιθανοτικά δίκτυα , η εκμάθηση βασισμένη σε κανόνες, οι συναρτήσεις κατωφλίου, οι μηχανές διανυσμάτων στήριξης. Κάθε μία από αυτές τις τεχνολογίες διαθέτει τη δική της στρατηγική μάθησης και το δικό της εύρος εφαρμογών, παρόλο που ένα πρόβλημα μπορεί να προσεγγιστεί με πολλαπλές στρατηγικές.

- Πηγή πληροφοριών

Η πληροφορία αποτελείται από τα δεδομένα που χρησιμοποιούνται για την εκπαίδευση των αλγορίθμων. Μπορεί να είναι διαφόρων μορφών όπως θετικά ή αρνητικά παραδείγματα(ονομάζονται παραδείγματα με ταμπέλα),απαντήσεις σε ερωτήματα, αναπληροφόρηση σε συγκεκριμένες ενέργειες κλπ. Μια πηγή πληροφοριών μπορεί να διαθέτει “θόρυβο”, δηλαδή σφάλματα ενώ τα παραδείγματα μπορεί να είναι ομαδοποιημένα πρωτού χρησιμοποιηθούν στη διαδικασία εκπαίδευσης.

- Σενάριο εκπαίδευσης

Το σενάριο εκπαίδευσης είναι πρακτικά η περιγραφή της διαδικασίας εκμάθησης. Τα σενάρια συνήθως είναι πεπερασμένα σε αριθμό, παρόλο που σε ένα επαγωγικό συμπέρασμα ένα πρόγραμμα μπορεί να τροφοδοτείται με απεριόριστη ποσότητα δεδομένων. Στην επιβλεπόμενη μάθηση το σενάριο λέει ότι το πρόγραμμα τροφοδοτείται

από παραδείγματα και θα πρέπει να προβλέψει την ετικέτα καθενός από τα επόμενα παραδείγματα πρωτού ο "δάσκαλος" αποκαλύψει την απάντηση. Στη μη επιβλεπόμενη μάθηση το σενάριο λέει ότι το πρόγραμμα θα πρέπει να εξάγει κάποιες ιδιότητες, του στιγμιοτύπου που λαμβάνει, από μόνο του χωρίς να υπάρχει κάποιος "δάσκαλος". Τέλος στην ενισχυτική μάθηση οι είσοδοι προέρχονται από απρόβλεπτο περιβάλλον και δίνεται θετική ή αρνητική ανάδραση-πληροφορία στο τέλος κάθε μικρής ακολουθίας εκμάθησης, όπως για παράδειγμα κατά την εκμάθηση της βέλτιστης στρατηγικής που θα ακολουθηθεί.

- Προηγούμενη γνώση

Η προηγούμενη γνώση έχει να κάνει με το τι γνωρίζουμε από πριν για το πεδίο ορισμού ,για παράδειγμα για συγκεκριμένες ιδιότητες του κόνσεπτ που πρόκειται να διδαχθεί. Όλο αυτό βοηθά στον περιορισμό των κλάσεων υποθέσεων που χρειάζονται να ληφθούν υπ΄όψιν από το πρόγραμμα κατά τη διάρκεια της εκμάθησης και συνεπώς τον περιορισμό της αβεβαιότητας για τα άγνωστα αντικείμενα που προκύπτουν και την ταχύτερη σύγκλιση. Τέλος το πρόγραμμα μπορεί να χρησιμοποιήσει την πρώτερη γνώση με σκοπό να προδιαθέσει την επιλογή της υπόθεσης που επρόκειτο να κάνει.

- Κριτήρια επιτυχίας

Τα κριτήρια για να θεωρηθεί η εκμάθηση επιτυχημένη ,για παράδειγμα για να καθορίσουμε πότε αυτή θα σταματήσει ή θα μεταβληθεί αναλόγως, εξαρτώνται από το σκοπό που έχουμε θέσει για την εκμάθηση και το αν το πρόγραμμα ταιριάζει στο έργο. Αν για παράδειγμα το πρόγραμμα χρησιμοποιείται σε κρίσιμα για την ασφάλεια περιβάλλοντα τότε η ακρίβεια θα πρέπει να φτάσει σε επαρκές επίπεδο στη διαδικασία της εκπαίδευσης έτσι ώστε να μπορεί να αποφανθεί ή να προβλέψει κατά τη χρήση. Ένα κριτήριο επιτυχίας μετριέται με γνώμονα τα σύνολα δοκιμών ή με θεωρητική ανάλυση.

- Επίδοση

Η επίδοση εξαρτάται από το χρόνο, το χώρο και την υπολογιστική ισχύ που χρειάζεται ώστε να ολοκληρωθεί η εκμάθηση ενός συγκεκριμένου κόνσεπτ, και φυσικά η ακρίβεια που επετεύχθη κατά τη διαδικασία. Υπάρχει μια ισορροπία (trade-off) μεταξύ του πλήθους των παραδειγμάτων που χρειάζονται για την εκπαίδευση ενός προγράμματος ,άρα των υπολογιστικών πόρων που χρησιμοποιούνται, και των μετέπειτα δυνατοτήτων του προγράμματος.

2.2 Το πρόβλημα της κατηγοριοποίησης

Η κατηγοριοποίηση ως πρόβλημα προκύπτει αρκετά συχνά στην πράξη. Για παράδειγμα, πως έχουμε προαναφέρει , όταν θέλουμε να φιλτράρουμε τα απεσταλμένα μηνύματα ηλεκτρονικού ταχυδρομείου από ανεπιθύμητη αλληλογραφία, ή όταν θέλουμε να κάνουμε διάγνωση για μια ασθένεια βασιζόμενοι σε ιστορικά στοιχεία ενός ασθενούς ώστε να αποφανθούμε εάν αυτός είναι υγιής ή όχι. Σε αυτές λοιπόν τις περιπτώσεις, όπως και σε πολλές παρόμοιες μας ενδιαφέρει μια απάντηση του τύπου ναι ή όχι οπότε μιλάμε και για δυαδικό πρόβλημα κατηγοριοποίησης. Ας σημειωθεί ότι οι κατηγορίες δεν είναι πάντοτε δύο, μπορεί να είναι και περισσότερες(multi-class classification).

Εικόνα 2.2. Δυαδικό πρόβλημα κατηγοριοποίησης, όπου μπορούμε να διαχωρίσουμε τα διαφορετικά σύνολα με γραμμικό ταξινομητή.

2.3 Αλγόριθμοι Κατηγοριοποίησης

Παρακάτω θα περιγραφούν οι αλγόριθμοι που χρησιμοποιούνται πιο συχνά στην επίλυση προβλημάτων κατηγοριοποίησης.

2.3.1 Κατηγοριοποίηση με κριτήρια Bayes

Ο κατηγοριοποιητής Bayes χρησιμοποιεί το γνωστό θεώρημα Bayes για να προβλέψουμε την κλάση αυτή που μεγιστοποιεί την μεταγενέστερη πιθανότητα. Ο κύριος στόχος είναι να εκτιμηθεί η συνδυαστική συνάρτηση πυκνότητας πιθανότητας για κάθε κλάση που μοντελοποιείται μέσω μιας πολυμεταβλητής κανονικής διανομής. Ο απλουστευμένος κατηγοριοποιητής Bayes υποθέτει ότι τα γνωρίσματα ,προς μελέτη, είναι ανεξάρτητα μεταξύ τους, παρόλα αυτά χρησιμοποιείται πολύ συχνά σε εφαρμογές.

Υποθέτουμε ότι έχουμε ένα σετ από δεδομένα D που αποτελείται από n σημεία x_i σε ένα d-διάστατο χώρο και έστω y_i η κλάση που ανήκει κάθε σημείο ,με $y_i \in \{c_1, c_{2,} \dots c_k\}$.Ο κατηγοριοποιητής Bayes απευθείας χρησιμοποιεί το θεώρημα Bayes για να προβλέψει την κλάση για το καινούργιο στιγμιότυπο x. Εκτιμά την μεταγενέστερη πιθανότητα $P(c_i \mid x)$ για κάθε κλάση c_i και διαλέγει την κλάση με τη μεγαλύτερη πιθανότητα. Η προβλεπόμενη κλάση για το x υπολογίζεται από τον τύπο:

$$\hat{y} = \arg\max\{ P(c_i|x)\}$$

Το θεώρημα του Bayes μας επιτρέπει να αντιστρέψουμε τη μεταγενέστερη πιθανότητα ως εξής:

$$P(c_i|x) = \frac{P(c_i|x) * P(c_i)}{P(x)}$$

Αφού ο παράγοντας P(x) είναι γνωστός για δεδομένο σημείο ο τύπος του Bayes μπορεί να γραφεί ξανά ως εξής:

$$\hat{y} = \arg\max\{ P(c_i|x)\} =$$

$$= \arg\max\left\{ \frac{P(x|c_i) * P(c_i)}{P(x)} \right\} = \arg\max\{ P(x|c_i) * P(c_i) \}$$

Με άλλα λόγια η προβλεπόμενη κλάση ουσιαστικά εξαρτάται από την πιθανότητα της ίδιας της κλάσης λαμβάνοντας υπ'όψιν και την προγενέστερη πιθανότητα που είχε.Για να κατηγοριοποιήσουμε τα σημεία θα πρέπει να εκτιμηθούν οι δύο όροι που χρειάζονται κατευθείαν από το σετ των δεδομένων D. Ας πούμε ότι D_i είναι ένα υποσύνολο του D που ανήκει στην κλάση c_i δηλαδή $D_i = \{x_j \in D \mid x_j$ ανήκει στην κλάση $y_j = c_i\}$. Το μέγεθος των δεδομένων είναι |D| = n και το μέγεθος της κάθε κλάσης-υποσυνόλου είναι $|D_i| = n_i$. Η προηγούμενη πιθανότητα για την κλάση c_i δίνεται από τον τύπο $\widehat{P}(c_i) = \frac{n_i}{n}$.

Για τον υπολογισμό της δεσμευμένης πιθανότητας $P(x|c_i)$ θα πρέπει να υπολογίσουμε πρώτα τη συνδιαστική πιθανότητα των x κατά μήκος όλων των d διαστάσεων δηλαδή των $P(x = (x_1,x_2,...x_i) \mid c_i)$.

Για ποσοτικά χαρακτηριστικά για να εκτιμηθεί η συνδιαστική πιθανότητα θα χρησιμοποιήσουμε μια παραμετρική προσέγγιση. Υποθέτουμε ότι κάθε κλάση c_i κατανεμηθεί με κανονικό τρόπο γύρω από κάποιο μέσο μ_i με τον αντίστοιχο πίνακα διακυμάνσεων Σ_i ,όπου και τα δύο προκύπτουν από το D_i. Για την κλάση c_i η κατανομή της πιθανότητας στο x θα δίνεται από τον τύπο:

$$f_i(\mathbf{x}) = f(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \frac{1}{(\sqrt{2\pi})^d \sqrt{|\boldsymbol{\Sigma}_i|}} \exp\left\{-\frac{(\mathbf{x}-\boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)}{2}\right\}$$

Αφού το c_i χαρακτηρίζεται από συνεχή κατανομή η πιθανότητα οποιουδήποτε σημείου θα πρέπει να ισούται με μηδέν δηλαδή $P(x \mid c_i)=0$. Παρόλα αυτά μπορούμε να μετρήσουμε τον όρο αυτό θεωρώντας ένα μικρό διάστημα ε>0 που έχει το κέντρο του στο x οπότε θα εχουμε:
$P(x \mid c_i)=2*\varepsilon*f_i(x)$. Η μεταγενέστερη πιθανότητα υπολογίζεται από τη σχέση:

$$P(c_i|\mathbf{x}) = \frac{2\epsilon \cdot f_i(\mathbf{x})P(c_i)}{\sum_{i=1}^{k} 2\epsilon \cdot f_i(\mathbf{x})P(c_i)} = \frac{f_i(\mathbf{x})P(c_i)}{\sum_{i=1}^{k} f_i(\mathbf{x})P(c_i)}$$

κι αφού το άθροισμα $\sum_{i=0}^{k} f_i(x) * P(c_i)$ παραμένει σταθερό για το x μπορούμε να προβλέψουμε την κλάση του x ως εξής:

$$\hat{y} = \arg\max\{f_i(x) * P(c_i)\}$$

Για την κατηγοριοποίηση αριθμητικών τιμών x ο κατηγοριοποιητής Bayes εκτιμά τις παραμέτρους μέσω των δειγμάτων του μέσου και του πίνακα διακυμάνσεων. Η μέση τιμή για την κλάση c_i ισούται με $\hat{\mu}_i = \frac{1}{n_i} * \sum_{x_j \in D_i} x_j$ και ο πίνακας διακυμάνσεων για τν κάθε κλάση υπολογίζεται από τη σχέση $\hat{\Sigma}_i = \frac{1}{n_i} * Z_i^T * Z_i$,όπου ο Z_i είναι ο κανονικοποημένος πίνακας για κάθε κλάση και δίνεται από τον τύπο $Z_i = D_i - 1 * \hat{\mu}_i^T$. Οι τιμές αυτές χρησιμοποιούνται στον υπολογισμό της πυκνότητας πιθανότητας ως εξής: $f_i(x) = f(x| \hat{\mu}_i, \hat{\Sigma}_i)$. Παρακάτω δίνεται σε ψευδοκώδικα ο αλγόριθμος κατηγοριοποίησης του Bayes.

BayesClassifier $(\mathbf{D} = \{(\mathbf{x}_j, y_j)\}_{j=1}^n)$:

for $i = 1, \cdots, k$ **do**
- $\mathbf{D}_i \leftarrow \{\mathbf{x}_j \mid y_j = c_i, j = 1, \cdots, n\}$ // class-specific subsets
- $n_i \leftarrow |\mathbf{D}_i|$ // cardinality
- $\hat{P}(c_i) \leftarrow n_i/n$ // prior probability
- $\hat{\boldsymbol{\mu}}_i \leftarrow \frac{1}{n_i} \sum_{\mathbf{x}_j \in \mathbf{D}_i} \mathbf{x}_j$ // mean
- $\mathbf{Z}_i \leftarrow \mathbf{D}_i - \mathbf{1}_{n_i} \hat{\boldsymbol{\mu}}_i^T$ // centered data
- $\widehat{\boldsymbol{\Sigma}}_i \leftarrow \frac{1}{n_i} \mathbf{Z}_i^T \mathbf{Z}_i$ // covariance matrix

return $\hat{P}(c_i), \hat{\boldsymbol{\mu}}_i, \widehat{\boldsymbol{\Sigma}}_i$ *for all* $i = 1, \cdots, k$

Testing (**x** and $\hat{P}(c_i)$, $\hat{\boldsymbol{\mu}}_i$, $\widehat{\boldsymbol{\Sigma}}_i$, **for all** $i \in [1, k]$):

$\hat{y} \leftarrow \arg\max_i \{ f(\mathbf{x} | \hat{\boldsymbol{\mu}}_i, \widehat{\boldsymbol{\Sigma}}_i) \cdot P(c_i) \}$

return $\hat{y}$

Με είσοδο ένα σετ δεδομένων D ο αλγόριθμος εκτιμά την προγενέστερη πιθανότητα, τον μέσο όρο και τον πίνακα διακύμανσης για την κάθε κλάση. Για επαλήθευση με δεδομένο ένα σημείο x απλά επιστρέφεται η κλάση με τη μεγαλύτερη μεταγενέστερη πιθανότητα. Η πολυπλοκότητα για την εκπαίδευση εξαρτάται από το βήμα υπολογισμού του πίνακα διακύμανσης το οποίο γίνεται σε χρόνο $O(nd^2)$.

Το πιο σημαντικό πρόβλημα που αντιμετωπίζουμε με τον κατηγοριοποιητή Bayes είναι η έλλειψη αρκετών δεδομένων ώστε να υπολογίσουμε με ακρίβεια τη συλλογική πυκνότητα πιθανότητας ειδικά σε μεγάλης διάστασης δεδομένα. Για παράδειγμα για αριθμητικά χαρακτηριστικά θα πρέπει να υπολογίσουμε $O(d^2)$ τιμές διακύμανσης και όσο οι διαστάσεις αυξάνονται απαιτείται να υπολογίσουμε πάρα πολλές παραμέτρους. Για μη αριθμητικά δεδομένα ακόμα κι αν οι δυνατές τιμές είναι μόνο δύο χρειάζεται να υπολογίσουμε την πιθανότητα για 2^d τιμές. Για να λύσουμε κάποια από αυτά τα προβλήματα μπορούμε να μειώσουμε τον αριθμό των παραμέτρων στην πράξη και να χρησιμοποιήσουμε τον απλοποιημένο κατηγοριοποιητή Bayes.

Η προσέγγιση του απλοποιημένου κατηγοριοποιητή Bayes κάνει την υπόθεση ότι όλα τα γνωρίσματα είναι μεταξύ τους ανεξάρητα. Αυτό οδηγεί στην πράξη σε πιο απλή αλλά και πιο αποτελεσματική κατηγοριοποίηση. Με την υπόθεση ανεξαρτησίας εννοείται ότι η δεσμευμένη πιθανότητα μπορεί να αποσυντεθεί ως εξής: $P(x \mid c_i) = P(x_1, x_2, \ldots, x_d \mid c_i) = \prod_{j=1}^{d} P(x_j|c_i)$. Για αριθμητικά γνωρίσματα μπορούμε να κάνουμε την υπόθεση ότι καθένα από αυτά είναι κανονικώς κατανεμημένο για κάθε κλάση c_i. Έστω ο μέσος μ_{ij} και η διασπορά σ_{ij}^2 για ένα χαρακτηριστικό X_j για την κλάση c_i. Η πιθανότητα για του x_j με δεδομένο ότι ανήκει στην κλάση c_i δίνεται από τη σχέση:

$$P(x_j|c_i) \propto f(x_j|\mu_{ij}, \sigma_{ij}^2) = \frac{1}{\sqrt{2\pi}\sigma_{ij}} \exp\left\{ -\frac{(x_j - \mu_{ij})^2}{2\sigma_{ij}^2} \right\}$$

Λόγω της απλοποιημένης υπόθεσης που κάναμε ο πίνακας των διακυμάνσεων θα είναι:

$$\boldsymbol{\Sigma}_i = \begin{pmatrix} \sigma_{i1}^2 & 0 & \ldots & 0 \\ 0 & \sigma_{i2}^2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \\ 0 & 0 & \ldots & \sigma_{id}^2 \end{pmatrix}$$

άρα και η ορίζουσα του πίνακα θα είναι $|\Sigma_i| = \det(\Sigma_i) = \sigma_{i1}^2 * \sigma_{ij2}^2 * \ldots * \sigma_{ijd}^2 = \prod_{j=1}^{d} \sigma_{ij}^2$, ενώ

$$\boldsymbol{\Sigma}_i = \begin{pmatrix} \sigma_{i1}^2 & 0 & \dots & 0 \\ 0 & \sigma_{i2}^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \\ 0 & 0 & \dots & \sigma_{id}^2 \end{pmatrix}$$

με $\sigma_{ij}^2 \neq 0$ για όλα τα j. Επίσης είναι:

$$(\mathbf{x} - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} (\mathbf{x} - \boldsymbol{\mu}_i) = \sum_{j=1}^{d} \frac{(x_j - \mu_{ij})^2}{\sigma_{ij}^2}$$

άρα τελικά έχουμε:

$$\begin{aligned} P(\mathbf{x}|c_i) &= \frac{1}{(\sqrt{2\pi})^d \sqrt{\prod_{j=1}^{d} \sigma_{ij}^2}} \exp\left\{ -\sum_{j=1}^{d} \frac{(x_j - \mu_{ij})^2}{2\sigma_{ij}^2} \right\} \\ &= \prod_{j=1}^{d} \left(\frac{1}{\sqrt{2\pi}\ \sigma_{ij}} \exp\left\{ -\frac{(x_j - \mu_{ij})^2}{2\sigma_{ij}^2} \right\} \right) \\ &= \prod_{j=1}^{d} P(x_j|c_i) \end{aligned}$$

Ο απλοποιημένος κατηγοριοποιητής Bayes χρησιμοποιεί ως μέσο τον $\hat{\mu}_i = (\hat{\mu}_{i1}, \dots, \hat{\mu}_{id})^T$ και ως πίνακα διακυμάνσεων τον $\hat{\Sigma}_i = \text{diag}(\sigma_{i1}^2, \dots, \sigma_{id}^2)$ για κάθε κλάση c_i. Στην περίπτωση αυτή λοιπόν το σύνολο των παραμέτρων που θα πρέπει να εκτιμηθούν θα είναι 2d αναφορικά με τον μέσο και τη διακύμανση για κάθε διάσταση X_j.

Παρακάτω φαίνεται σε ψευδοκώδικα ο αλγόριθμος. Έχοντας ένα σετ δεδομένων D, εκτιμάται η προγενέστερη πιθανότητα και ο μέσος για κάθε κλάση. Στη συνέχεια υπολογίζεται η διακύμανση $\widehat{\sigma_{ij}^2}$ για καθένα από τα γνωρίσματα X_j με όλες τις d διακυμάνσεις για την κλάση c_i καταχωρημένη στο διάνυσμα $\hat{\sigma}_i$. Η διακύμανση για το X_j ακολουθεί αφού πρώτα κανονικοποιήσουμε τα δεδομένα μέσω της σχέσης: $Z_i = D_i - 1 * \hat{\mu}_i^T$, υποδηλώνοντας ως Z_{ij} τα κανονικοποιημένα δεδομένα για την κλάση c_i που ανταποκρίνονται στο γνώρισμα X_j. Τελικά η διακύμανση δίνεται από τον τύπο : $\hat{\sigma} = \frac{1}{n_i} * Z_{ij}^T * Z_{ij}$. Η εκπαίδευση του απλοποιημένου αλγορίθμου Bayes είναι πολύ γρήγορη με υπολογιστική πολυπλοκότητα O(nd). Για την επαλήθευση με δεδομένο ένα σημείο x μας επιστρέφεται η κλάση με τη μεγαλύτερη μεταγενέστερη πιθανότητα εκφρασμένη ως το γινόμενο της δεσμευμένης πιθανότητας για κάθε διάσταση και της προγενέστερης πιθανότητας της συγκεκριμένης κλάσης.

NAIVEBAYES $(\mathbf{D} = \{(\mathbf{x}_j, y_j)\}_{j=1}^{n})$:

for $i = 1, \cdots, k$ **do**
- $\mathbf{D}_i \leftarrow \{\mathbf{x}_j \mid y_j = c_i, j = 1, \cdots, n\}$ // class-specific subsets
- $n_i \leftarrow |\mathbf{D}_i|$ // cardinality
- $\hat{P}(c_i) \leftarrow n_i/n$ // prior probability
- $\hat{\boldsymbol{\mu}}_i \leftarrow \frac{1}{n_i} \sum_{\mathbf{x}_j \in \mathbf{D}_i} \mathbf{x}_j$ // mean
- $\mathbf{Z}_i = \mathbf{D}_i - \mathbf{1} \cdot \hat{\boldsymbol{\mu}}_i^T$ // centered data for class c_i
- **for** $j = 1, .., d$ **do** // class-specific variance for X_j
  - $\hat{\sigma}_{ij}^2 \leftarrow \frac{1}{n_i} Z_{ij}^T Z_{ij}$ // variance
- $\hat{\boldsymbol{\sigma}}_i = \left(\hat{\sigma}_{i1}^2, \ldots, \hat{\sigma}_{id}^2\right)^T$ // class-specific attribute variances

return $\hat{P}(c_i), \hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\sigma}}_i$ *for all* $i = 1, \cdots, k$

TESTING (**x and** $\hat{P}(c_i)$, $\hat{\boldsymbol{\mu}}_i$, $\hat{\boldsymbol{\sigma}}_i$, **for all** $i \in [1, k]$):

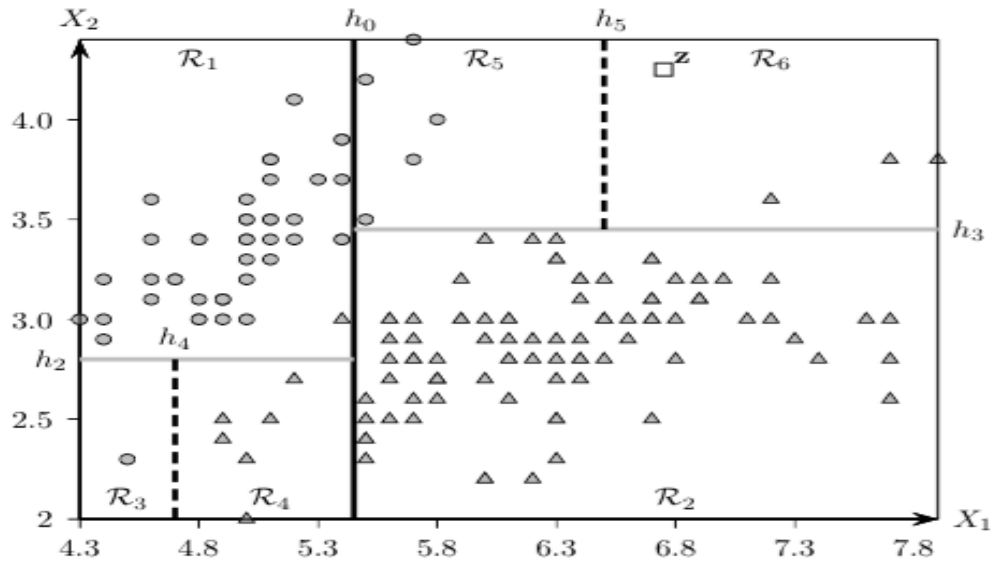
$$\hat{y} \leftarrow \arg\max_i \left\{ \hat{P}(c_i) \prod_{j=1}^{d} f(x_j | \hat{\mu}_{ij}, \hat{\sigma}_{ij}^2) \right\}$$

return $\hat{y}$

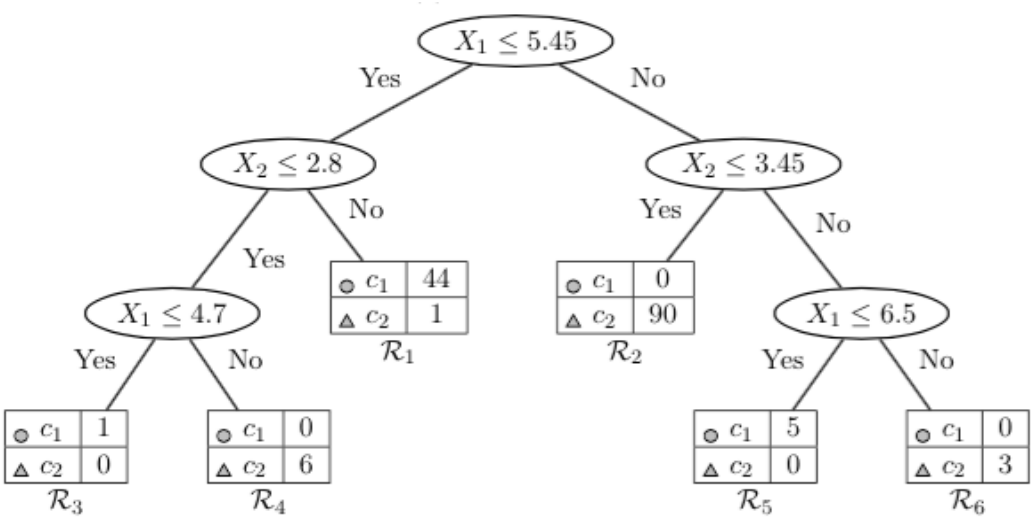
2.3.2 Κατηγοριοποίηση με Δένδρα Αποφάσεων

Έστω ότι έχουμε ένα σετ δεδομένων προς εκπαίδευση D= $\{x_i, y_i\}_{i=1}^{n}$ που περιλαμβάνει n σημεία σε ένα d-δίαστατο χώρο, με y_i την ετικέτα της κλάσης για το σημείο x_i . Υποθέτουμε ότι οι διαστάσεις ή τα γνωρίσματα X_j είναι είτε αριθμητικά είτε κατηγοριακά κι ότι υπάρχουν k διακριτές κλάσεις έτσι ώστε $y_i \in \{c_1, c_2, \ldots, c_k\}$. Ο κατηγοριοποιητής που χρησιμοποιεί δέντρα αποφάσεων είναι αναδρομικός και προβλέπει την κλάση $\hat{y}_i$ για κάθε σημείο x_i. Αν R είναι ο χώρος δεδομένων που περικλείει το σετ των σημείων του D, τότε ένα δέντρο αποφάσεων χρησιμοποιεί ένα υπερεπίπεδο παράλληλων αξόνων για να διαχωρίσει τον R σε δύο ημιχώρους-περιοχές τις R_1 και R_2, που συνεπάγονται με τη σειρά τους το διαχωρισμό των σημείων εισόδων σε D_1 και D_2 . Καθεμία από αυτές τις περιοχές μπορεί αναδρομικά να διαχωριστεί με υπερεπίπεδο παράλληλων αξόνων μέχρι τα σημεία που ανήκουν στην ίδια περιοχή να έχουν την ίδια ετικέτα,δηλαδή τα περισσότερα σημεία να ανήκουν στην ίδια κλάση. Η προκύπτουσα ιεραρχία των αποφάσεων διαχωρισμού συνιστά ένα δέντρο αποφάσεων, με τους κόμβους φύλλα να έχουν την ετικέτα της επικρατέστερης κλάσης μεταξύ αυτών των σημείων σε αυτές τις περιοχές. Για την κατηγοριοποίηση ενός καινούργιου σημείου θα πρέπει να επανεκτιμήσουμε αναδρομικά σε ποιο υπερεπίπεδο ανήκει μέχρι να φτάσουμε σε κόμβο φύλλο του δέντρου, οπότε προβλέπουμε την κλάση που ανήκει ως την ετικέτα του φύλλου.

Ένα δέντρο αποφάσεων αποτελείται από εσωτερικούς κόμβους που αντιπροσωπεύουν τις αποφάσεις που ανταποκρίνονται στα υπερεπίπεδα ή τα σημεία διαχωρισμού και τους κόμβους φύλλα που αντιπροσωπεύουν περιοχές του χώρου των δεδομένων και έχουν την ετικέτα της επικρατούσας κλάσης. Μια περιοχή χαρακτηρίζεται από το υποσύνολο των σημείων που βρίσκονται στην περιοχή αυτή. Ένα υπερεπίπεδο h(x) ορίζεται από όλα αυτα τα σημεία x που ικανοποιούν την εξίσωση: $h(x) : w^T x + b = 0$. Το w $\in \mathcal{R}^d$ είναι το διάνυσμα βάρους και η σταθερά b λειτουργεί ως αντιστάθμιση του ημιεπιπέδου από την αρχή των αξόνων.



Εικόνα 2.3. Αναδρομικός διαχωρισμός



Εικόνα 2.4. Δένδρο αποφάσεων

Αν τώρα θεωρήσουμε ότι το διάνυσμα w δεσμεύεται εκ των προτέρων με κάποιο από τα βασικά μοναδιαία διανύσματα $\{e_1, e_2, \dots, e_d\}$ όπου $e_i \in \mathcal{R}^d$ ισούται με 1 για την j-διάσταση και 0 για τις άλλες. Εάν $x=(x_1, x_2, \dots, x_d)^T$ και $w=e_j$ τότε η εξίσωση h(x) ξαναγράφεται h(x): e_j^T x + b=0 , δηλαδή h(x): x_j + b =0, με το b να παράγει διαφορετικά υπερεπίπεδα κατά τη διάσταση X_j. Το υπερεπίπεδο ορίζει μια απόφαση ή ενα σημείο διαχωρισμού αφού διαχωρίζει τα δεδομένα του χώρου $\mathcal{R}$ σε δύο ημιχώρους. Όλα τα σημεία όπου ισχύει $h(x) \leq 0$ είτε βρίσκονται πάνω στο υπερεπίπεδο είτε από τη μια πλευρά του, ενώ όλα τα σημεία που ισχύει $h(x) > 0$ βρίσκονται στην άλλη πλευρά. Εάν λοιπόν $h(x) \leq 0$ χρησιμοποιώντας την παραπάνω εξίσωση έχουμε ότι $x_i + b \leq 0$, άρα $x_i \leq -b$, κι αφού το x_i είναι κάποια τιμή από τη διάσταση X_j και η σταθερά b μπορεί να πάρει οποιαδήποτε τιμή η γενική μορφή ενός σημείου διαχωρισμού για ένα γνώρισμα X_j δίνεται από τη σχέση $X_j \leq \upsilon$, όπου $\upsilon = -b$ είναι μια τιμή στο πεδίο ορισμού του γνωρίσματος X_j. Η απόφαση $X_j \leq \upsilon$ διαχωρίζει τα δεδομένα εισόδου του $\mathcal{R}$ σε δύο περιοχές $\mathcal{R}_Y$ και $\mathcal{R}_N$ που υποδεικνύουν όλα τα δυνατά σημεία που ικανοποιούν την απόφαση και αυτά που δεν την ικανοποιούν.

Κάθε διαχωρισμός του $\mathcal{R}$ σε $\mathcal{R}_Y$ και $\mathcal{R}_N$ συνεπάγεται επίσης και σε δυαδική διαίρεση των αντίστοιχων σημείων εισόδου του D. Έτσι από ένα σημείο της εξίσωσης $X_j \leq v$ συνεπάγεται και διαχωρισμός των δεδομένων: $D_Y = \{x \mid x \in D,\ x_j \leq v\}$ και $D_N = \{x \mid x \in D,\ x_j > v\}$, όπου D_Y είναι το υποσύνολο των σημείων που βρίσκονται στην περιοχή $\mathcal{R}_Y$ και D_N το υποσύνολο των σημείων που βρίσκονται στην περιοχή $\mathcal{R}_N$.Η καθαρότητα(purity) μιας περιοχής $\mathcal{R}_Y$ καθορίζεται από το μίγμα των κλάσεων για τα σημεία στα αντίστοιχα δεδομένα που διαχωρίζονται D_j. Η καθαρότητα ορίζεται ως το κλάσμα του πλήθους των σημείων με την επικρατούσα ετικέτα στο D_j ,δηλαδή $\text{purity}(D_j) = max_i\left\{\frac{n_{ji}}{n_j}\right\}$, όπου $n_j = |D_j|$ είναι ο συνολικός αριθμός των σημείων στην περιοχή R_j και n_{ji} ο αριθμός των σημείων D_j με ετικέτα κλάσης c_i.

Εκτός από τα αριθμητικά γνωρίσματα ένα δέντρο αποφάσεων μπορεί ακόμα να χειριστεί και κατηγοριακά δεδομένα. Για ένα τέτοιο γνώρισμα X_j τα σημεία διαχωρισμού ή αλλιώς οι αποφάσεις είναι $X_j \in V$, όπου $V \subset \text{dom}(X_j)$, όπου $\text{dom}(X_j)$ είναι το πεδίο ορισμού για το X_j .Συνεπώς αυτός ο διαχωρισμός μπορεί να θεωρηθεί ότι είναι ανάλογος του υπερεπιπέδου για τα κατηγοριακά όμως δεδομένα. Ως αποτέλεσμα έχουμε δύο ημιχώρους , μια περιοχή $\mathcal{R}_Y$ που αποτελείται από τα σημεία x που ικανοποιούν τη συνθήκη $x_i \in V$ και στην άλλη περιοχή $\mathcal{R}_N$ με τα σημεία που ικανοποιούν τη συνθήκη $x_i \notin V$.

Ένα από τα πλεονεκτήματα των δένδρων αποφάσεων είναι ότι παράγουν κάποια μοντέλα που είναι εύκολο να ερμηνευτούν. Συγκεκριμένα ένα δέντρο μπορεί να διαβαστεί ως ένα σετ κανόνων ,με κάθε προγενέστερο κανόνα να περιλαμβάνει τις αποφάσεις των εσωτερικών κόμβων και να δημιουργεί ένα μονοπάτι μέχρι να φτάσουμε στα φύλλα, όπου και δίνεται μια ετικέτα. Επιπλέον από τη στιγμή που οι περιοχές είναι τεμαχισμένες και καλύπτουν όλο το χώρο, όλοι αυτοί οι κανόνες μπορούν να ερμηνευτούν ως εναλλακτικές ή διαζεύξεις. Παρακάτω περιγράφεται σε ψευδοκώδικα ο αλγόριθμος.

```
DecisionTree (D, η, π):
1  n ← |D| // partition size
2  n_i ← |{x_j | x_j ∈ D, y_j = c_i}| // size of class c_i
3  purity(D) ← max_i { n_i / n }
4  if n ≤ η or purity(D) ≥ π then // stopping condition
5      c* ← arg max_i { n_i / n } // majority class
6      create leaf node, and label it with class c*
7      return
8  (split-point*, score*) ← (∅, 0) // initialize best split-point
9  foreach (attribute X_j) do
10     if (X_j is numeric) then
11         (v, score) ← Evaluate-Numeric-Attribute(D, X_j)
12         if score > score* then (split-point*, score*) ← (X_j ≤ v, score)
13     else if (X_j is categorical) then
14         (V, score) ← Evaluate-Categorical-Attribute(D, X_j)
15         if score > score* then (split-point*, score*) ← (X_j ∈ V, score)
   // partition D into D_Y and D_N using split-point*, and call
      recursively
16 D_Y ← {x ∈ D | x satisfies split-point*}
17 D_N ← {x ∈ D | x does not satisfy split-point*}
18 create internal node split-point*, with two child nodes, D_Y and D_N
19 DecisionTree(D_Y); DecisionTree(D_N)
```

Ο αλγόριθμος παραπάνω παίρνει ως είσοδο κάποια δεδομένα D, δύο παραμέτρους η και π, όπου το η είναι το μέγεθος του φύλλου και π το κατώφλι του μεγέθους της καθαρότητας για κάθε φύλλο. Διαφορετικά σημεία διαχωρισμού αποτιμώνται για κάθε γνώρισμα που ανήκει στο D. Οι αποφάσεις για δεδομένα αριθμητικά είναι της μορφής $X_j \leq$ υ για κάποια τιμή υ στο εύρος του γνωρίσματος X_j και για κατηγοριακά δεδομένα της μορφής $X_j \in V$ για κάποιο υποσύνολο τιμών στο πεδίο ορισμού του X_j .Το καλύτερο σημείο διαχωρισμού επιλέγεται για να διαιρέσει τα δεδομένα σε δύο υποσύνολα D_Y και D_N όπου το D_Y περιλαμβάνει όλα τα σημεία $x \in D$ που ικανοποιούν την απόφαση διαχωρισμού και το D_N που περιλαμβάνει όλα εκείνα που δεν ικανοποιούν την απόφαση διαχωρισμού. Η μέθοδος αυτή καλείται αναδρομικά στα D_Y και D_N .

Μπορούν να χρησιμοποιηθούν αρκετές συνθήκες διακοπής για να σταματήσουν την αναδρομική διαίρεση. Η πιο απλή εξαρτάται από το μέγεθος της διαίρεσης του D. Εάν ο αριθμός των σημείων n στο D πέσει πιο κάτω από τον αριθμό που όρισε ο χρήστης και ισούται με το κατώφλι η, τότε σταματά η διαδικασία της διαίρεσης και θέτει το D ως φύλλο. Αυτή η διαδικασία αποτρέπει την υπερπροσαρμογή (over-fitting) του μοντέλου στα δεδομένα εκπαίδευσης, αποτρέποντας να μοντελοποιήσει πολύ μικρά υποσύνολα των δεδομένων. Για το πρόβλημα της υπερπροσαρμογής θα αναφερθούμε πιο αναλυτικά στη συνέχεια του κεφαλαίου. Το να ελέγξουμε το μέγεθος της διαίρεσης όμως, από μόνο του, δεν είναι αρκετό αφού κάποια στιγμή η διαίρεση δεν θα έχει πλέον νόημα.Συνεπώς η αναδρομική διαίρεση μπορεί επιπλέον να τερματιστεί εάν η καθαρότητα του D είναι πιο πάνω του κατωφλίου καθαρότητας που είναι π.

Με δεδομένο ένα σημείο διαχωρισμού της μορφής $X_j \leq$ υ ή $X_j \in V$ για αριθμητικό ή κατηγοριακό γνώρισμα χρειαζόμαστε ένα κριτήριο για την αξιολόγηση του σημείου. Θα θέλαμε να διαλέξουμε το σημείο αυτό που δίνει τον καλύτερο διαχωρισμό ή διαφοροποίηση μεταξύ δύο διαφορετικών ετικετών κλάσεων. Η εντροπία γενικά μετρά το μέγεθος της αταξίας ή της αβεβαιότητας σε ένα σύστημα. Όσον αφορά την καηγοριοποίηση μια διαίρεση έχει χαμηλότερη εντροπία (ή χαμηλότερη αταξία) εάν είναι σχετικά καθαρή, για παράδειγμα εάν τα περισσότερα σημεία έχουν την ίδια ετικέτα. Από την άλλη μια διαίρεση έχει υψηλότερη εντροπία(ή μεγαλύτερη αταξία) εάν οι ετικέτες σε μια κλάση είναι ανομοιογενείς, οπότε και δεν υπάρχει καμία κλάση που να επικρατεί.

Η εντροπία ενός σετ από επισημασμένα σημεία D ορίζεται απότην παρακάτω σχέση: $H(D) = -\sum_{i=1}^{k} P(c_i|D) * log_2 P(c_i|D)$ όπου $P(c_i \mid D)$ είναι η πιθανότητα της κλάσης c_i στο D και k είναι το πλήθος των κλάσεων. Αν μια περιοχή είναι καθαρή, δηλαδή τα σημεία της ανήκουν στην ίδια κλάση, τότε η εντροππία είναι μηδέν. Στην περίπτωση που όλες οι κλάσεις είναι ανομοιογενείς και καθεμιά από αυτές εμφανίζεται με ίση πιθανότητα $P(c_i \mid D) = \frac{1}{k}$ τότε η εντροπία έχει την υψηλότερη τιμή $H(D) = \log_2 k$. Αν υποθέσουμε ότι ένα σημείο διαχωρισμού διαιρεί το D σε D_Y και D_N. Η εντροπία διαχωρισμού ως σταθμισμένη σε καθένα από τα δύο μέρη δίνεται ως εξής: $H(D_Y, D_N) = \frac{n_Y}{n} H(D_Y) + \frac{n_N}{n} H(D_N)$, όπου $n=|D|$ είναι ο αριθμός των σημείων στο D και $n_Y=|D_Y|$, $n_N=|D_N|$ είναι ο αριθμός των σημείων στα D_Y και D_Y αντίστοιχα. Για να δούμε εάν το σημείο διαχωρισμού οδηγεί σε ελάχιστη συνολική εντροπία ορίζουμε το κέρδος πληροφορίας για το δεδομένο σημείο ως: $Gain(D, D_Y, D_N) = H(D) - H(D_Y, D_N)$. Όσο υψηλότερο το κέρδος οληροφορίας τόσο μικρότερη η εντροπία και τόσο καλύτερο θεωρείται τη σημείο διαχωρισμού. Έτσι με δοσμένα τα σημεία διαχωρισμού και τις αντίστοιχες διαιρέσεις μπορούμε να αξιολογήσουμε κάθε σημείο και να διαλέξουμε αυτό που δίνει την υψηλότερη τιμή στο κέρδος πληροφορίας.

2.3.3 Κατηγοριοποίηση με Νευρωνικά Δίκτυα

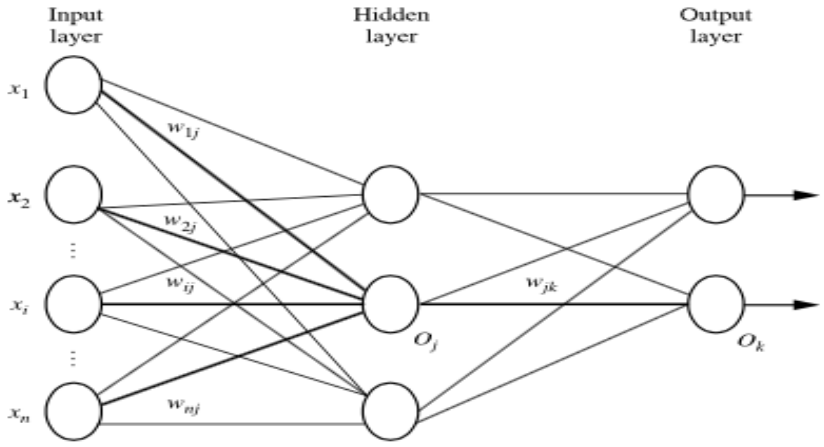
Τα νευρωνικά δίκτυα (ΤΝΔ) παρέχουν μια γενική, πρακική μέθοδο για την εκμάθηση πραγματικών,χαρακτηριστικά διακριτών και διανυσματικών τιμών συναρτήσεων μέσα από παραδείγματα. Αλγόριθμοι όπως η "προς-τα-πίσω-διάδοση"(backpropagation)

διαμορφώνουν κατάλληλα τις παραμέτρους του δικτύου ώστε να προσαρμόζονται καλύτερα στα δεδομένα εκπαίδευσης και να προκύπτουν τα καλύτερα δυνατά ζεύγη εισόδων-εξόδων. Η μελέτη για τα ΤΝΔ ξεκίνησε ως έμπνευση από τη παρατήρηση ότι τα συστήματα βιολογικής εκμάθησης μπορούν και κατασκευάζουν πολύπλοκους ιστούς διασυνδεδεμένων νευρώνων. Έτσι κατά αυτή την αναλογία τα ΤΝΔ κατασκευάζονται από πυκνά διασυσνδεδεμένα σύνολα απλών μονάδων οι οποίες παίρνουν ένα αριθμό πραγματικών τιμών ως είσοδο(πιθανώς την έξοδο άλλων μονάδων) και παράγουν μια μοναδική πραγματική τιμή ως έξοδο(πιθανώς η είσοδος πολλών άλλων μονάδων). Οι συνδέσεις μεταξύ των μονάδων έχουν ένα βάρος, μια στάθμη δηλαδή που σχετίζεται με την κάθε μονάδα.

Κατά τη διάρκεια της φάσης εκπαίδευσης το δίκτυο μαθαίνει να ρυθμίζει τα βάρη έτσι ώστε να είναι δυνατό να προβλεφθεί η σωστή ετικέτα κλάσης για τις πλειάδες(ζεύγη εισόδων - εξόδων). Η εκμάθηση με χρήση νευρωνικών δικτύων είναι χρονοβόρα και για αυτό το λόγο είναι κατάλληλη για εφαρμογές όπου αυτό είναι εφικτό. Απαιτείται ένας αριθμός παραμέτρων που τυπικά καθορίζονται καλύτερα εμπειρικά ανάλογα με την τοπολογία του δικτύου ή αλλιώς τη δομή του. Η προσέγγιση με χρήση των ΤΝΔ δέχτηκε αρκετή κριτική λόγω του ότι δεν μπορούν να ερμηνευτούν με ευκολία. Για παράδειγμα είναι δύσκολο για τους ανθρώπους να ερμηνεύσουν το συμβολικό νόημα πίσω από τις τιμές των βαρών και των κρυφών μονάδων του δικτύου. Αυτά τα μειονεκτήματα έκαναν τα ΤΝΔ αρχικά λιγότερο επιθυμητά για την εξόρυξη δεδομένων.

Από την άλλη πλευρά πλεονεκτήματα των ΤΝΔ περιλαμβάνουν την υψηλή αντοχή σε δεδομένα με θόρυβο όπως και τη δυνατότητα να κατηγοριοποιούν κάποια μοτίβα στα οποία δεν έχουν εκπαιδευτεί. Μπορούν να χρησιμοποιηθούν όταν έχουν πολύ μικρή γνώση των σχέσεων μεταξύ γνωρισμάτων και κλάσεων. Επίσης είναι κατάλληλα για συνεχούς τιμής εισόδους και εξόδους αντιθέτως δηλαδή με τους αλγορίθμους δένδρων αποφάσεων. Χρησιμοποιούνται με μεγάλη επιτυχία σε μεγάλης κλίμακας πραγματικών δεδομένων συμπεριλαμβανομένων της αναγνώρισης χειρόγραφων χαρακτήρων, εργαστηριακής ιατρικής και εκπαίδευσης ενός υπολογιστή στο να προφέρει σωστά ένα κείμενο στην αγγλική γλώσσα. Οι αλγόριθμοι ΤΝΔ είναι παραλληλοποιήσιμοι, που σημαίνει ότι μπορούν να χρησιμοποιθούν στο να επιταχυνθεί μια υπολογιστική διαδικασία. Πρόσφατα έχουν βρεθεί διάφορες τεχνικές για εξαγωγή κανόνων από εκπαίδευση ΤΝΔ. Όλοι αυτοί οι παράγοντες που προαναφέρθηκαν συμβάλλουν στην χρησιμότητα των ΤΝΔ στα προβλήματα κατηγοριοποιήσης και στην αριθμητική πρόβλεψη στην εξόρυξη δεδομένων.

Ο αλγόριθμος της “προς-τα-πισω-διάδοσης” εφαρμόζει εκμάθηση σε μια πολυεπίπεδη έμπροσθεν τροφοδότηση του νευρωνικού δικτύου. Μαθαίνει επαναληπτικά ένα σύνολο από βάρη για την πρόβλεψη της ετικέτας των πλειάδων. Μια έμπροσθεν τροφοδότηση των ΤΝΔ αποτελείται από ένα στρώμα εισόδου, ένα ή περισσότερα ενδιάμεσα κρυμμένα στρώματα και ένα στρωμα εξόδου.



Εικόνα 2.5. Νευρωνικό δίκτυο

Κάθε επίπεδο του δικτύου αποτελέιται από μονάδες. Οι είσοδοι στο δίκτυο αντιστοιχούν στα γνωρίσματα που μετρώνται για κάθε πλειάδα εκπαίδευσης. Οι μονάδες - είσοδοι που τροφοδοτούνται ταυτόχρονα αποτελούν το επίπεδο εισόδου. Αυτές περνώντας από το επίπεδο εισόδου σταθμίζονται και στη συνέχεια τροφοδοτούν ταυτόχρονα ένα δεύτερο στρώμα μονάδων, όπως ακριβώς κάνει ένας νευρώνας, γνωστό ως κρυμμένο στρώμα. Οι έξοδοι του κρυμμένου στρώματος μπορεί να είναι είσοδοι ενός άλλου κρυμμένου στρώματος κ.ο.κ. Ο αριθμός των κρυμμένων στρωμάτων είναι αυθαίρετος , παρόλα αυτά στην πράξη συνήθως χρησιμοποιείται μόνο ένα. Οι σταθμισμένοι έξοδοι του τελευταίου κρυμμένου στρώματος είναι οι είσοδοι των μονάδων που συνιστούν το στρώμα εξόδου, από όπου προκύπτει και η πρόβλεψη του δικτύου για τις δοσμένες πλειάδες.

Πριν ξεκινήσει η εκπαίδευση ο χρήστης θα πρέπει να αποφασίσει για την τοπολογία του δικτύου διευκρινίζοντας τον αριθμό των μονάδων του στρώματος εισόδου, τον αριθμό των κρυμμένων στρωμάτων, αν είναι μεγαλύτερος από ένα, τον αριθμό των μονάδων σε κάθε κρυμμένο στρώμα και τον αριθμό των μονάδων στο στρώμα εξόδου. Κανονικοποιώντας τις τιμές εισόδου για κάθε μετρούμενο γνώρισμα στις πλειάδες θα βοηθήσει στην επιτάχυνση της διαδικασίας μάθησης. Τυπικά οι τιμές εισόδου κανονικοποιούνται έτσι ώστε να πέφτουν μεταξύ του 0.0 και του 1.0 .Τα γνωρίσματα διακριτής τιμής μπορούν να προσαρμοστούν έτσι ώστε να υπάρχει μία μονάδα εισόδου ανά τιμή στο πεδίο ορισμού. Για παράδειγμα εάν ένα γνώρισμα Α έχει τρεις πιθανές τιμές $\{\alpha_0 , \alpha_1 , \alpha_2\}$ τότε μπορούμε να θέσουμε τρεις μονάδες εισόδου για να αναπαραστήσουμε το Α, ας πούμε I_0 , I_1 , I_2. Καθεμία από αυτές αρχικοποιείται στο 0. Αν $A= \alpha_0$ τότε το I_0 τίθεται στο 1 και τα υπόλοιπα στο 0 κ.ο.κ.

Τα νευρωνικά δίκτυα μπορούν να χρησιμοποιηθούν τόσο στην κατηγοριοποίηση όσο και στην πρόβλεψη αριθμητικών τιμών. Για την κατηγοριοποίηση μία μονάδα εξόδου μπορεί να χρησιμοποιηθεί για την αναπαράσταση δύο κλάσεων, όπου η τιμή 1 αναπαριστά τη μία και η τιμή 0 την άλλη. Αν υπάρχουν παραπάνω από δύο κλάσεις τότε χρησιμοποιείται μία μονάδα εξόδου για την καθεμία. Δεν υπάρχουν ξεκάθαροι κανόνες για τον καλύτερο δυνατό αριθμοό κρυμμένων στρωμάτων από μονάδες. Ο σχεδιασμός του δικτύου είναι μια διαδικασία δοκιμής και σφάλματος και μπορεί να επηρρεάσει πολύ την ακρίβεια του τελικού εκπαιδευμένου δικτύου. Οι αρχικές τιμές των βαρών μπορούν επίσης να επηρρεάσουν την ακρίβεια. Εάν ένα δίκτυο έχει εκπαιδευτεί και η ακρίβειά του δεν θεωρείται αποδεκτή τότε συνήθως επαναλαμβάνεται η εκπαίδευση με διαφορετική τοπολογία ή διαφορετικές αρχικές τιμές στα βάρη. Αρκετές αυτοματοποιημένες τεχνικές έχουν προταθεί για την αναζήτηση καλής δομής του δικτύου. Αυτές συνήθως χρησιμοποιούν τον αλγόριθμο hill-climbing που ξεκινά από μια αρχική δομή και μεταβάλλεται επιλεκτικά.

Με τον αλγόριθμο backpropagation που αναφέρθηκε το ΤΝΔ μαθαίνει με την επαναλαμβανόμενη επεξεργασία ενός σετ δεδομένων που αποτελείται από πλειάδες, συγκρίνοντας τις προβλέψεις του δικτύου για κάθε πλειάδα με την πραγματική γνωστή τιμή στην οποία στοχεύουμε. Αυτή η τιμή στόχος μπορεί να είναι μια γνωστή ετικέτα κλάσης για προβλήματα κατηγοριοποίησης ή μια συνεχής τιμή για προβλήματα αριθμητικών προβλέψεων. Για κάθε πλειάδα προς εκπαίδευση τα βάρη μεταβάλλονται έτσι ώστε να ελαχιστοποιηθεί το μέσο τετραγωνικό σφάλμα μεταξύ της πρόβλεψης του δικτύου και της τιμής στόχου. Αυτές οι μεταβολές γίνονται στην προς-τα-πίσω κατεύθυνση μέσω κάποιων κρυμμένων στρωμάτων που καταλήγουν στο πρώτο κρυμμένο στρώμα. Παρόλο που δεν είναι εγγυημένο γενικά τα βάρη τελικά συγκλίνουν στις τιμές και η διαδιακασία εκμάθησης σταματά.Παρακάτω φαίνεται ο ψευδοκώδικα που περιγράφει τα βήματα του αλγορίθμου:

(1) Initialize all weights and biases in *network*;
(2) **while** terminating condition is not satisfied {
(3) **for** each training tuple $\boldsymbol{X}$ in D {
(4) // Propagate the inputs forward:
(5) **for** each input layer unit j {
(6) $O_j = I_j$; // output of an input unit is its actual input value
(7) **for** each hidden or output layer unit j {
(8) $I_j = \sum_i w_{ij} O_i + \theta_j$; //compute the net input of unit j with respect to the previous layer, i
(9) $O_j = \frac{1}{1+e^{-I_j}}$; } // compute the output of each unit j
(10) // Backpropagate the errors:
(11) **for** each unit j in the output layer
(12) $Err_j = O_j(1 - O_j)(T_j - O_j)$; // compute the error
(13) **for** each unit j in the hidden layers, from the last to the first hidden layer
(14) $Err_j = O_j(1 - O_j) \sum_k Err_k w_{jk}$; // compute the error with respect to the next higher layer, k
(15) **for** each weight w_{ij} in *network* {
(16) $\Delta w_{ij} = (l) Err_j O_i$; // weight increment
(17) $w_{ij} = w_{ij} + \Delta w_{ij}$; } // weight update
(18) **for** each bias θ_j in *network* {
(19) $\Delta \theta_j = (l) Err_j$; // bias increment
(20) $\theta_j = \theta_j + \Delta \theta_j$; } // bias update
(21) } }

Αρχικά τα δεομένα μας είναι ένα σετ δεδομένων D που αποτελείται από πλειάδες και τις αντίστοιχες τιμές στόχους, τον ρυθμό εκμάθησης l και το πολυεπίπεδο, έμπροσθεν τροφοδοτούμενο δίκτυο. Τα βάρη στις ακμές του δικτύου αρχικοποιούνται σε μικρούς τυχαίους αριθμούς, με εύρος για παράδειγμα -1.0 ως 1.0 ή -0.5 ως 0.5. Κάθε μονάδα έχει ένα συστηματικό σφάλμα (bias) που αντιστοιχίζεται σε αυτή. Τα σφάλματα αυτά όμοια αρχικοποιούνται σε μικρούς τυχαίους αριθμούς. Κάθε πλειάδα Χ επεξεργάζεται σύμφωνα με τα παρακάτω στάδια: Η πλειάδα προς εκπαίδευση τροφοδοτεί το στρώμα εισόδου του δικτύου. Οι είσοδοι περνούν μέσω των μονάδων εισόδου αμετάβλητες. Αυτό σημαίνει ότι για μια μονάδα εισόδου j ,η έξοδος O_j ισούται με την τιμή εισόδου I_j. Στη συνέχεια η καθαρά είσοδος και έξοδος της κάθε μονάδας στα κρυμμένα στρώματα αλλά και στο στρώμα εξόδου υπολογίζεται ως γραμμικός συνδυασμός των εισόδων.

Κάθε τέτοια μονάδα έχει κάποιο αριθμό από εισόδους οι οποίες στην πραγματικότητα είναι οι έξοδοι των μονάδων που συδέονται από το προηγούμενο στρώμα. Κάθε σύνδεση έχει το δικό της βάρος και για υπολογίσουμε την καθαρά είσοδο στη μονάδα κάθε μία από τις εισόδους που συνδέεται σε αυτή πολλαπλασιάζεται με το βάρος της και αθροίζονται όλα μαζί. Με δεδομένη μια μονάδα j σε ένα κρυμμένο στρώμα ή εξόδου η καθαρά είσοδος I_j ισούται με:
$I_j = \sum_i w_{ij} O_i + \theta_j$ όπου w_{ij} είναι το βάρος του κάθε συνδέσμου από τη μονάδα i στη μονάδα j του προηγούμενου στρώματος, O_i είναι η έξοδος της μονάδας i από το προηγούμενο στρώμα και θ_j το συστηματικό σφάλμα της μονάδας.

Κάθε μονάδα στο κρυμμένο ή στο εξωτερικό στρώμα παίρνει την καθαρά είσοδο και στη συνέχεια εφαρμόζει μια συνάρτηση ενεργοποίησης. Η συνάρτηση συμβολίζει την ενεργοποίηση ενός νευρώνα από τη μονάδα. Η συνάρτηση που χρησιμοποιείται είναι η λογιστική ή αλλιώς σιγμοειδής συνάρτηση(logistic or sigmoid function) και δεδομένης της καθαράς εισόδου I_j στη μονάδα j τότε η O_j ως έξοδος της μονάδας j υπολογίζεται από τη σχέση: $O_j = \frac{1}{1+e^{-I_j}}$.Η συνάρτηση αυτή αναφέρεται επίσης ως συνάρτηση

σύνθλιψης(squashing function) διότι αντιστοιχίζει ένα μεγάλο πεδίο ορισμού που έχει ως είσοδο σε ένα κατά πολύ μικρότερο εύρους από 0 έως 1. Η συνάρτηση αυτή δεν είναι γραμμική αλλά είναι παραγωγίσιμη, οπότε επιτρέπει στον αλγόριθμο να μοντελοποιεί προβλήματα κατηγοριοποίησης που δεν διαχωρίζεται με γραμμικό τρόπο. Υπολογίζουμε τις τιμές εξόδου O_j για κάθε κρυμμένο στρώμα και έτσι έχουμε την πρόβλεψη του δικτύου. Στην πράξη είναι καλή ιδέα να αποθηκεύσουμε τις ενδιάμεσες τιμές εξόδου σε κάθε μονάδα αφού αυτές θα ξαναχρειαστούν στη συνέχεια όταν θα γίνει οπισθοχώρηση για τον υπολογισμό του σφάλματος. Αυτό το απλό κόλπο μπορεί να μειώσει ουσιωδώς τον όγκο των απαραίτητων υπολογισμών.

Το σφάλμα υπολογίζεται με οπισθοχώρηση όπως αναφέρθηκε, με το να ενημερώνονται οι τιμές των βαρών και των συστηματικών σφαλμάτων ώστε να αντικατοπτρίζεται το σφάλμα ποθ προέκυψε από την πρόβλεψη του δικτύου. Για μια μονάδα j στο στρώμα εξόδου το σφάλμα Err_j υπολογίζεται από τον τύπο: $Err_j = O_j (1- O_j)(T_j - O_j)$, όπου O_j είναι η πραγματική έξοδος της μονάδας και T_j είναι η γνωστή τιμή στόχος της δεδομένης πλειάδας προς εκπαίδευση. Ο παράγοντας $O_j (1- O_j)$ είναι η παράγωγος της σιγμοειδούς συνάρτησης. Για τον υπολογισμό του σφάλματος μιας μονάδας j του κρυμμένου στρώματος λαμβάνεται υπόψιν το σταθμισμένο άθροισμα των σφαλμάτων των μονάδων των επόμενων στρωμάτων που συνδέονται με τη μονάδα j. Το σφάλμα της μονάδας j του κρυμμένου στρώματος δίνεται από τον τύπο: $Err_j = O_j (1- O_j)\sum_k Err_k w_{jk}$,όπου w_{jk} είναι τα βάρη της σύδεσης από τη μονάδα j στη μονάδα k του επόμενου υψηλότερου στρώματος και Err_k είναι το σφάλμα της μονάδας k. Τα βάρη και τα συστηματικά σφάλματα ενημερώνονται για να αντικατοπτρίσουν τα σφάλματα της οπισθοχώρησης και υπολογίζονται: $\Delta w_{ij} = (l)Err_j O_i$ και $w_{ij} = w_{ij} + \Delta w_{ij}$ όπου Δw_{ij} είναι η αλλαγή που υφίσταται ο όρος w_{ij}.

Η μεταβλητή l είναι ο ρυθμός εκμάθησης, μια σταθερά τυπικά που παίρνει τιμή από 0.0 ως 1.0 . Ο αλγόριθμος μαθαίνει χρησιμοποιώντας μια μέθοδο βελτιστοποίησης πρώτης τάξης που ονομάζεται gradient descent για την αναζήτηση του συνόλου των βαρών που ταιριάζουν στα δεδομένα εκπαίδευσης έτσι ώστε να επιτευχθεί η ελαχιστοποίηση της μέγιστης τετραγωνικής απόστασης μεταξύ της πρόβλεψης κλάσης του δικτύου και της γνωστής τιμής στόχου των πλειάδων. Ο ρυθμός εκμάθησης βοηθά στην αποφυγή τοπικού ελαχίστου στο χώρο αποφάσεων, δηλαδή το να υπάρξη σύγκλιση της τιμής των βαρών αλλά να μην είναι η βέλτιστη λύση, και κατά συνέπεια στην εύρεση ολικού ελαχίστου. Εάν ο ρυθμός εκμάθησης είναι πολύ μικρός τότε η μάθηση διεξάγεται με πολύ αργό ρυθμό, ενώ αν είναι πολύ μεγάλος τότε μπορεί να προκύψει ταλάντευση μεταξύ ανεπαρκών λύσεων. Ένας εμπειρικός κανόνας είναι να θέσουμε το ρυθμό εκμάθησης στο $\frac{1}{t}$,όπου t είναι ο αριθμός των επαναλήψεων του συνόλου εκπαίδευσης μέχρι στιγμής.

Τα συστηματικά σφάλματα ενημερώνονται με βάση τις ακόλουθες εξισώσεις, όπου $\Delta\theta_j$ είναι η αλλαγή του συστηματικού σφάλματος θ_j : $\Delta\theta_j = (l)\ Err_j$ και $\theta_j = \theta_j + \Delta\theta_j$. Να σημειωθεί ότι τα βάρη και τα συστηματικά σφάλματα ενημερώνονται μετά την παρουσίαση της κάθε πλειάδας κι αυτό ονομάζεται ενημέρωση κατά περίπτωση. Εναλλακτικά η αύξηση των βαρών και των σφαλμάτων θα μπορούσαν να συσσωρευτούν σε μεταβλητές έτσι ώστε να ενημερώνονται αφότου όλες οι πλειάδες στο σύνολο εκπαίδευσης έχουν παρουσιαστεί. Η τελευταία αυτή στρατηγική ονομάζεται εποχιακή ενημέρωση, όπου κάθε επανάληψη στο εκπαιδευόμενο σετ συνιστά μια εποχή. Στη θεωρία η μαθηματική παραγώγιση της οπισθοχώρησης απασχολεί εποχιακή ενημέρωση, στην πράξη είναι πιο κοινή η ενημέρωση κατά περίπτωση γιατί αποφέρει πιο ακριβή αποτελέσματα.

Η εκπαίδευση σταματά όταν: όλα τα Δw_{ij} της προηγούμενης εποχής είανι τόσο μικρά έτσι ώστε να είναι κάτω από ένα συγκεκριμένο κατώφλι, αν το ποσοστό των πλειάδων που κατηγοριοποιήθηκαν λάθος στην προηγούμενη εποχή είναι πιο κάτω από ένα κατώφλι και όταν ένας προκαθορισμένος αριθμός από εποχές έχει λήξει. Πρακτικά αρκετές εκατοντάδες χιλιάδες εποχές μπορεί να απαιτούνται πριν οι τιμές των βαρών συγκλίνουν. Η υπολογιστική αποδοτικότητα του αλγορίθμου της οπισθοχώρησης εξαρτάται από το χρόνο που καταναλώνεται στην εκπαίδευση του δικτύου. Με δεδομένο τον αριθμό των πλειάδων

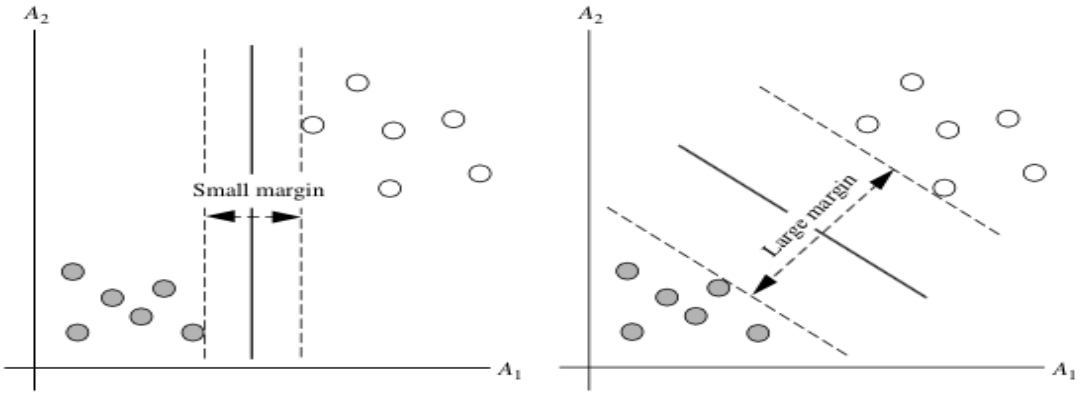
|D| και w των βαρών κάθε εποχή απαιτεί O(|D| x w) χρόνο.Παρόλα αυτά στη χειρότερη περίπτωση ο αριθμός των εποχών μπορεί να είναι εκθετικός του αριθμού των εισόδων n. Στην πράξη ο απαιτούμενος χρόνος για τη σύγκλιση των δικτύων ποικίλλει πολύ. Ένας αριθμός τεχνικών είναι διαθέσιμος που μπορούν να επιταχύνουν τη διαδικασία εκπαίδευσης. Για παράδειγμα μια τεχνική γνωστή ως προσομοιωμένη ανόπτηση(simulated annealing) μπορεί να χρησιμοποιηθεί και να εγγυηθεί τη σύγκλιση σε ολική βέλτιστη τιμή.

2.3.4 Κατηγοριοποίηση με Μηχανές Διανυσμάτων Στήριξης (Support Vector Machines)

Η μέθοδος των ΜΔΣ χρησιμοποιείται για κατηγοριοποίηση γραμμικών και μη γραμμικών δεδομένων. Συνοπτικά ο αλγόριθμος δουλεύει ως εξής: χρησιμοποιεί μια μη γραμμική τεχνική αντιστοίχησης για το μετασχηματισμό του πρωτότυπου συνόλου των δεδομένων προς εκπαίδευση σε μια πιο υψηλή διάσταση, Σε αυτή την καινούργια διάσταση αναζητά το βέλτιστο γραμμικό διαχωριστικό υπερεπίπεδο,δηλαδή το όριο που διαχωρίζει τις πλειάδες που ανήκουν σε μια κλάση από αυτές μιας διαφορετικής κλάσης. Με κατάλληλη μη γραμμική αντιστοίχηση σε μια επαρκώς υψηλή διάσταση τα δεδομένα που ανήκουν σε δύο διαφορετικές κλάσεις μπορούν πάντα να διαχωριστούν με ένα υπερεπίπεδο. Η μέθοδος των ΜΔΣ βρίσκει αυτό το υπερεπίπεδο χρησιμοποιώντας διανύσματα στήριξης, ουσιώδεις δηλαδή πλειάδες προς εκπαίδευση, και περιθώρια, που ορίζονται από τα διανύσματα στήριξης.

Οι πρώτες μελέτες παρουσιάστηκαν το 1992 από τον Vladimir Vapnik όμως ήδη από το 1960 είχαν αρχίσει να γίνεται ουσιαστική έρευνα από τους Vapnik και Chervonenkis στη θεωρία της στατιστική μάθησης. Παρόλο που ο χρόνος εκπαίδευσης ακόμα και της πιο γρήγορης μεθόδου ΜΔΣ μπορεί να είναι εξαιρετικά αργός, τα αποτελέσματα είναι πολύ ακριβή, διότι στηρίζονται στη δυνατότητα να μοντελοποιούν πολύπλοκα μη γραμμικά όρια απόφασης. Σε σύγκριση με άλλες μεθόδους ρέπουν πολύ λιγότερο προς την υπερπροσαρμογή(overfitting), ενώ τα διανύσματα στήριξης μπορούν να δώσουν μια συμπαγή περιγραφή του μοντέλου εκμάθησης. Οι ΜΔΣ χρησιμοποιούνται τόσο για προβλήματα κατηγοριοποίησης όσο και για προβλήματα αριθμητικών προβλέψεων. Έχουν βρει εφαρμογή σε πολλούς τομείς όπως την αναγνώριση χαρακτήρων γραπτού λόγου, αναγνώριση αντικειμένων, αναγνώριση φωνής και πρόβλεψη χρονοσειρών.

Ας υποθέσουμε ότι έχουμε ένα πρόβλημα κατηγοριοποίησης με δύο κλάσεις το οποίο είναι γραμμικώς διαχωρίσιμο. Έστω D το σύνολο των δεδομένων που δίνεται στη μορφή (X_1, y_1), (X_2, y_2), … , $(X_{|D|}, y_{|D|})$, όπου X_i το σετ των πλειάδων προς εκπαίδευση με αντίστοιχες ετικέτες κλάσης y_i. Κάθε y_i μπορεί να πάρει μία τιμή είτε +1 είτε -1. Μπορεί να υπάρχει άπειρος αριθμός από διαχωριστικές γραμμές που μπορούμε να τραβήξουμε, όμως θέλουμε να βρούμε την “καλύτερη”, η οποία είναι αυτή που θα έχει το μικρότερο σφάλμα κατηγοριοποίησης σε άγνωστες, μέχρι τώρα, πλειάδες. Γενικεύοντας στις n διαστάσεις θέλουμε να βρούμε το καλύτερο υπερεπίπεδο, που αποτελεί το όριο απόφασης ανεξάρτητα από τον αριθμό των χαρακτηριστικών εισόδων. Μια προσέγγιση λοιπόν με ΜΔΣ είναι να ψάξουμε να βρούμε το μέγιστο οριακό υπερεπίπεδο και τα αντίστοιχα περιθώρια.



Το παραπάνω σχήμα δείχνει δύο, από τους πολλούς, πιθανούς τρόπους διαχωρισμού. Και τα δύο υπερεπίπεδα διαχωρίζουν σωστά όλες τις δεδομένες πλειάδες. Όμως περιμένουμε το υπερεπίπεδο με το μεγαλύτερο περιθώριο να είναι πιο ακριβές στην κατηγοριοποίηση μελλοντικών πλειάδων από ότι αυτό με το μικρότερο περιθώριο. Για αυτό λοιπόν το λόγο κατά τη διάρκεια της εκμάθησης ή της διαδικασίας εκπαίδευσης η ΜΔΣ αναζητά για το υπερεπίπεδο με το μέγιστο περιθώριο, οπότε κατά συνέπεια δίνει και τον καλύτερο διαχωρισμό μεταξύ κλάσεων.

Ορίζοντας το περιθώριο μπορούμε να πούμε ότι η μικρότερη απόσταση από ένα υπερεπίπεδο ως τη μια πλευρά του περιθωρίου του ισούται με τη μικρότερη απόσταση από το υπερεπίπεδο ως την άλλη πλευρά του περιθωρίου του, όπου οι "πλευρές" του περιθωρίου είναι παράλληλες με το υπερεπίπεδο. Όταν έχουμε να κάνουμε με το μέγιστο περιθώριο του υπερεπιπέδου αυτή η απόσταση είναι στην πραγματικότητα η μικρότερη απόσταση από το μέγιστο περιθώριο του υπερεπιπέδου ως την πλησιέστερη εκπαιδευόμενη πλειάδα, ή κλάση.

Ένα διαχωριστικό υπερεπίπεδο μπορεί μαθηματικά να περιγραφεί με την ακόλουθη εξίσωση: $W * X + b = 0$, όπου $W=\{w_1, w_2, \dots, w_n\}$ είναι το διάνυσμα των βαρών, n ο αριθμός των χαρακτηριστικών και το b αναπαριστά το συστηματικό σφάλμα. Αν θεωρήσουμε την απλή περίπτωση 2 χαρακτηριστικών εισόδου A_1, A_2 και θεωρήσουμε κι ότι το b είναι ένα επιπλέον βάρος w_0 μπορούμε να ξαναγράψουμε την εξίσωση ως: $w_0 + w_1 x_1 + w_2 x_2 = 0$. Κάθε σημείο που βρίσκεται πάνω από το διαχωριστικό υπερεπίπεδο ικανοποιεί τη σχέση $w_0 + w_1 x_1 + w_2 x_2 > 0$ και όμοια κάθε σημείο που βρίσκεται κάτω από το διαχωριστικό υπερεπίπεδο ικανοποιεί τη σχέση $w_0 + w_1 x_1 + w_2 x_2 < 0$. Τα βάρη μπορούν να προσαρμοστούν έτσι ώστε τα υπερεπίπεδα που ορίζουν τις άκρες των περιθωρίων να γράφονται: $H_1 : w_0 + w_1 x_1 + w_2 x_2 \geq 1$ για $y_i = +1$ και $H_2 : w_0 + w_1 x_1 + w_2 x_2 \leq -1$ για $y_i = -1$. Αυτό σημαίνει ότι όποια πλειάδα βρίσκεται στο επίπεδο H_1 ή πάνω από αυτό ανήκει στην κλάση +1 ,ενώ όποια βρίσκεται στο επίπεδο H_2 ή κάτω από αυτό ανήκει στην κλάση -1.

Συνδιάζοντας τις δύο παραπάνω ανισότητες παίρπνουμε: $y_i (w_0 + w_1 x_1 + w_2 x_2) \geq 1$,$\forall\, i$. Οποιαδήποτε πλειάδα που βρίσκεται πάνω στα υπερεπίπεδα H_1 ή H_2 και ικανοποιούν την τελευταία ανισότητα καλείται διάνυσμα υποστήριξης. Τα διανύσματα αυτά είναι το ίδιο κοντά με το (διαχωριστικό) υπερεπίπεδο μέγιστου περιθωρίου. Ουσιαστικά τα διανύσματα υποστήριξης ορίζονται από τις πιο δύσκολα κατηγοριοποιήσιμες πλειάδες, οι οποίες δίνουν τις περισσότερες πληροφορίες για την κατηγοριοποίηση. Από όλα αυτά μπορούμε να εξάγουμε μια φόρμουλα για το μέγεθος του μέγιστου περιθωρίου. Η απόσταση από το διαχωριστικό υπερεπίπεδο σε οποιοδήποτε σημείο πάνω στο H_1 είναι $\frac{1}{||W||}$, όπου ||W|| είναι η ευκλείδια νόρμα του W, η οποία ισούται με $\sqrt{W * W}^2$. Εξ ορισμού αυτό είναι ίσο με την απόσταση από οποιοδήποτε σημείο πάνω στο H_2 ως το διαχωριστικό υπερεπίπεδο. Το μέγιστο όμως περιθώριο ισούται με $\frac{2}{||W||}$.

Χρησιμοποιώντας κάποια προχωρημένα μαθηματικά κόλπα, που δεν θα περιγραφούν εδώ, μπορούμε να ξαναγράψουμε την τελευταία εξίσωση έτσι ώστε να μετατραπεί σε αυτό

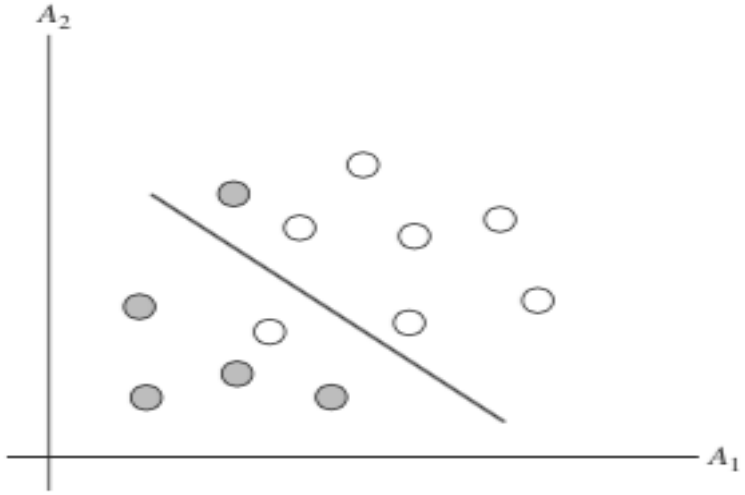
που είναι γνωστό ως τετραγωνικό πρόβλημα βελτιστοποίησης με περιορισμούς. Ενδεικτικά ένα από τα κόλπα θα ήταν η χρησιμοποίηση του μετασχηματισμού Lagrange και μετά η επίλυση χρησιμοποιώντας τις συνθήκες Karush-Kuhn-Tucker (ΚΚΤ). Εάν τα δεδομένα είναι λίγα(ας πούμε λιγότερες από 2000 πλειάδες) οποιοδήποτε λογισμικό πακέτο για επίλυση τετραγωνικών προβλημάτων βελτιστοποίησης με περιορισμούς μπορεί να χρησιμοποιηθεί ωστε να βρεθούν τα διανύσματα υποστήριξης και το υπερεπίπεδο με το μέγιστο περιθώριο.Για περισσότερα δεδομένα θα πρέπει να χρησιμοποιηθούν πιο ειδικοί και αποδοτικοί αλγόριθμοι για την εκπαίδευση των ΜΔΣ.

Από τη στιγμή που θα βρούμε τα διανύσματα στήριξης καθώς και το υπερεπίπεδο με το μέγιστο περιθώριο η ΜΔΣ έχει εκπαιδευτεί. Το υπερεπίπεδο με το μέγιστο περιθώριο είναι ένα γραμμικό όριο οπότε και η αντίστοιχη ΜΔΣ μπορεί να χρησιμοποιηθεί ώστε να διαχωρίσει γραμμικά διαχωρίσιμα δεδομένα. Μια τέτοια ΜΔΣ που έχει εκπαιδευτεί την ονομάζουμε γραμμική. Με βάση το μετασχηματισμό Lagrange το υπερεπίπεδο μέγιστου περιθωρίου μπορεί να ξαναγραφτεί ως το όριο της απόφασης: $d(X^T) = \sum_{i=1}^{l} y_i a_i X_i X^T + b_0$, όπου y_i είναι η ετικέτα της κλάσης του διανύσματος στήριξης X_i , X^T είναι μια πλειάδα για δοκιμή , α_i και b_0 είναι αριθμητικοί παράμετροι που καθορίζονται αυτόματα από τη βελτιστοποίηση ή τον αλγόριθμο των ΜΔΣ που αναφέραμε πιο πριν, και l είναι ο αριθμός των διανυσμάτων στήριξης. Αν προσέξουμε περισσότερο θα δούμε ότι οι α_i είναι οι πολλαπλασιαστές Lagrange. Για γραμμικώς διαχωρίσιμα δεδομένα τα διανύσματα στήριξης είναι υποσύνολο των πραγματικών εκπαιδευόμενων πλειάδων.

Με δεδομένη μια πλειάδα για δοκιμή X^T,την αντικαθιστούμε στην παραπάνω εξίσωση και περιμένουμε να δούμε ποιο είναι το πρόσημο του αποτελέσματος. Αυτό μας λέει σε ποια πλευρά του υπερεπιπέδου πέφτει η εξεταζόμενη πλειάδα. Εάν το πρόσημο είναι θετικό τότε το X^T βρίσκεται στο υπερεπίπεδο μέγιστου περιθωρίου ή πάνω από αυτό οπότε η ΜΔΣ προβλέπει ότι το X^T ανήκει στην κλάση +1. Αντίθετα εάν το πρόσημο είναι αρνητικό τότε το X^T βρίσκεται στο υπερεπίπεδο μέγιστου περιθωρίου ή κάτω από αυτό οπότε η ΜΔΣ προβλέπει ότι το X^T ανήκει στην κλάση -1. Να σημειωθεί ότι ο μετασχηματισμός Lagrange περιέχει ένα εσωτερικό γινόμενο μεταξύ του διανύσματος στήριξης X_i και της πλειάδας για δοκιμή X^T, το οποίο αποδεικνύεται πολύ χρήσιμο για την εύρεση του υπερεπιπέδου μέγιστου περιθωρίου και των διανυσμάτων στήριξης για την περίπτωση που τα δεδομένα δεν είναι γραμμικώς διαχωρίσιμα.

Η πολυπλοκότητα του κατηγοριοποιητή που εκπαιδεύεται χαρακτηρίζεται από τον αριθμό των διανυσμάτων στήριξης παρά από τις διαστάσεις που έχουν τα δεδομένα. Οι ΜΔΣ γενικά ρέπουν λιγότερο προς την υπερπροσαρμογή από ότι άλλες μέθοδοι. Τα διανύσματα στήριξης είναι ,όπως είπαμε προηγουμένως, οι κρίσιμες εκπαιδευόμενες πλειάδες που βρίσκονται πιο κοντά στο όριο απόφασης. Εάν όλες οι άλλες πλειάδες απομακρύνονταν και η διαδικασία εκπαίδευσης επαναλαμβανόταν θα βρίσκαμε και πάλι το ίδιο διαχωριστικό υπερεπίπεδο. Επιπλέον ο αριθμός των διανυσμάτων στήριξης που βρέθηκε μπορεί να χρησιμοποιηθεί για τον υπολογισμό ενός άνω ορίου του αναμενόμενου σφάλματος κατηγοριοποίησης, το οποίο είναι ανεξάρτητο από τις διαστάσεις των δεδομένων. Τέλος μια ΜΔΣ με μικρό αριθμό από διανύσματα στήριξης μπορεί να πραγματοποιεί καλύτερη γενίκευση, ακόμα και αν οι διαστάσεις των δεδομένων είναι πολλές.

Στην περίπτωση που τα δεδομένα δεν διαχωρίζονται με γραμμικό τρόπο δεν μπορούν να εφαρμοστούν όσα προαναφέρθηκαν. Αφού λοιπόν δε μπορούμε να τραβήξουμε κάποια ευθεία γραμμή ώστε να διαχωριστούν οι κλάσεις θα πρέπει να επεκτείνουμε την προσέγγισή μας και να δημιουργήσουμε μη γραμμικές ΜΔΣ. Τέτοιες ΜΔΣ είναι ικανές να βρίσκουν μη γραμμικά όρια αποφάσεων, όπως για παράδειγμα μη γραμμικές υπερεπιφάνειες, στο χώρο εισόδου.Η μέθοδος επέκτασης περιλαμβάνει 2 κύρια βήματα.Το πρώτο βήμα περιλαμβάνει το μετασχηματισμό των αρχικών δεδομένων εισόδων σε υψηλότερης διάστασης χώρο χρησιμοποιώντας μη γραμμική αντιστοίχιση.Πολλές κοινές μη γραμμικές αντιστοιχίσεις



Εικόνα 2.6. Περίπτωση 2-διαστάσεων μη διαχωρήσιμων δεδομένων

μπορούν να χρησιμοποιηθούν σε αυτό το βήμα και μερικές θα περιγραφούν παρακάτω.Μόλις τα δεδομένα μετασχηματιστούν κατά το δεύτερο βήμα αναζητάμε για ένα γραμμικώς διαχωρίσιμο υπερεπίπεδο στο νέο χώρο. Τελικά καταλήγουμε με ένα τετραγωνικό πρόβλημα βελτιστοποίησης που μπορεί να επιλυθεί με χρήση γραμικών ΜΔΣ. Το υπερεπίπεδο μέγιστου περιθωρίου που βρέθηκε στον καινούργιο χώρο αντιστοιχεί στη μη γραμμική διαχωριστική υπερεπιφάνεια στον πραγματικό αρχικό χώρο.

Ας υποθέσουμε ότι έχουμε ένα 3-διάστατο διάνυσμα εισόδου X=(x_1, x_2, x_3) που αντιστοιχίζεται στον 6-διάστατο χώρο Z χρησιμοποιώντας τις συναρτήσεις $\varphi_1(X)= x_1$, $\varphi_2(X)= x_2$, $\varphi_3(X)= x_3$, $\varphi_4(X)= (x_1)^2$, $\varphi_5(X)= x_1 x_2$, $\varphi_6(X)= x_1 x_3$. Το υπερεπίπεδο απόφασης στο νέο χώρο είναι d(Z)=WZ+b, όπουW και Z διανύσματα. Η γραμμική προσέγγιση είναι να λύσουμε για το W και το b και στη συνέχεια να αντικαταστήσουμε πίσω έτσι ώστε το γραμμικό υπερεπίπεδο απόφασης στον καινούργιο χώρο (Z) να ανταποκρίνεται σε μη γραμμικό πολυώνυμο δεύτερης τάξης στον 3-διάστατο χώρο, άρα: $d(Z) = w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 (x_1)^2 + w_5 x_1 x_2 + w_6 x_1 x_3 + b = w_1 z_1 + w_2 z_2 + w_3 z_3 + w_4 z_4 + w_5 z_5 + w_6 z_6 + b$. Όμως υπάρχουν κάποια προβλήματα όπως το πώς θα διαλέξουμε τη μη γραμμική αντιστοίχιση-συνάρτηση σε χώρο με πιο υψηλή διάσταση αλλά και το υψηλό κόστος αυτού του υπολογισμού.Από την εξίσωση $d(X^T) = \sum_{i=1}^{l} y_i a_i X_i X^T + b_0$,που είδαμε, με δεδομένη τη δοκιμαστική πλειάδα X^T θα πρέπει να υπολογίσουμε το εσωτερικό γινόμενό της με το κάθε διάνυσμα στήριξης. Στην εκπαίδευση θα πρέπει να υπολογίσουμε ένα παρόμοιο εσωτερικό γινόμενο αρκετές φορές έτσι ώστε να βρούμε το υπερεπίπεδο μέγιστου περιθωρίου. Η διαδικασία αυτή είναι εξαιρετικά ακριβή, αφού ο υπολογισμός του εσωτερικού γινομένου και μόνο είναι βαρύς και ακριβός.

Για καλή μας τύχη όμως μπορούμε να χρησιμοποιήσουμε ένα άλλο μαθηματικό κόλπο. Στην επίλυση του τετραγωνικού προβλήματος βελτιστοποίησης της γραμμικής ΜΔΣ οι εκπαιδευόμενες πλειάδες εμφανίζονται μόνο στη μορφή εσωτερικών γινομένων, $\varphi(X_i)$ $\varphi(X_j)$, όπου $\varphi(X)$ είναι απλά η μη γραμμική συνάρτηση αντιστοίχησης που εφαρμόζεται στο μετασχηματισμό των εκπαιδευόμενων πλειάδων.Αντί λοιπόν να υπολογίσουμε το εσωτερικό γινόμενο των μετασχηματισμένων πλειάδων δεδομένων προκύπτει ότι είναι μαθηματικά ισοδύναμο να εφαρμόσουμε μια συνάρτηση πυρήνα(kernel function) $K(X_i, X_j)$ στα αρχικά δεδομένα εισόδου,δηλαδή $K(X_i, X_j)= \varphi(X_i)\ \varphi(X_j)$.Με άλλα λόγια όπου εμφανίζεται το γίνομενο $\varphi(X_i)\ \varphi(X_j)$ στον αλγόριθμο εκπαίδευσης μπορούμε να το αντικαταστήσουμε με το $K(X_i, X_j)$. Με αυτό τον τρόπο όλοι οι υπολογισμοί γίνονται στον αρχικό χώρο ο οποίος το πιο πιθανό είναι να είναι μικρότερης διάστασης. Έτσι μπορούμε ασφαλώς να αποφύγουμε την αντιστοίχηση, ενώ προκύπτει ότι δεν χρειάζεται καν να ξέρουμε ποια είναι η αντιστοίχηση.

Αφού λοιπόν εφαρμόσουμε τα παραπάνω μπορούμε να προχωρήσουμε στην εύρεση του μέγιστου διαχωριστικού υπερεπιπέδου. Η διαδικασία είναι παρόμοια με αυτή που τα δεδομένα είναι γραμμικώς διαχωρίσιμα,παρόλο που εμπεριέχει την τοποθέτηση ενός άνω ορίου C ,ορισμένο από το χρήστη, στους πολλαπλασιαστές Lagrange α_i .Αυτό το άνω όριο προσδιορίζεται πειραματικά. Έχουν μελετηθεί ιδιότητες από πολλού είδους συναρτήσεων πυρήνα που θα μπορούσαν να χρησιμοποιηθούν για να αντικαταστήσουν το εσωτερικό γινόμενο, 3 από τις οποίες είναι πιο γνωστές και φαίνονται παρακάτω. Καθεμιά από αυτές τις

$$\textbf{Polynomial kernel of degree } h\textbf{:} \quad K(\boldsymbol{X}_i, \boldsymbol{X}_j) = (\boldsymbol{X}_i \cdot \boldsymbol{X}_j + 1)^h$$

$$\textbf{Gaussian radial basis function kernel:} \quad K(\boldsymbol{X}_i, \boldsymbol{X}_j) = e^{-\|\boldsymbol{X}_i - \boldsymbol{X}_j\|^2/2\sigma^2}$$

$$\textbf{Sigmoid kernel:} \quad K(\boldsymbol{X}_i, \boldsymbol{X}_j) = \tanh(\kappa \boldsymbol{X}_i \cdot \boldsymbol{X}_j - \delta)$$

Εικόνα 2.7. Συνηθισμένες kernel functions

συναρτήσεις έχει ως αποτέλεσμα διαφορετικό μη γραμμικό κατηγοριοποιητή στον αρχικό χώρο εισόδου. Σε αντιπαραβολή με τα νευρωνικά δίκτυα σημειώνουμε ότι η τελική απόφαση των υπερεπιπέδων που βρέθηκαν για τις ΜΔΣ είναι του ίδιου τύπου με αυτές που βρέθηκαν με χρήση άλλων γνωστών κατηγοριοποιητών νευρωνικών δικτύων. Για παράδειγμα μια ΜΔΣ με συνάρτηση Gauss ακτινικής βάσης δίνει την ίδια απόφαση υπερεπιπέδου που δίνει ο τύπος νευρωνικού δικτύου που είναι γνωστός ως συνάρτηση δικτύου ακτινικής βάσης. Επιπλέον μια ΜΔΣ με σιγμοειδή συνάρτηση πυρήνα είναι ισοδύναμη με ένα απλό νευρωνικό δίκτυο δύο στρωμάτων γνωστό και ως multilayer perceptron.

Δεν υπάρχουν κανόνες για τον προσδιορισμό του παραδετού πυρήνα που θα οδηγήσει στην πιο ακριβή ΜΔΣ. Στην πράξη ο πυρήνας που επιλέγεται δεν κάνει και μεγάλη διαφορά στην τελική ακρίβεια. Η τεχνική εκπαίδευσης των ΜΔΣ πάντα βρίσκει μια ολική λύση, σε αντίθεση με τα νευρωνικά δίκτυα και την τεχνική της οπισθοχώρησης όπου προκύπτουν πολλά τοπικά ελάχιστα. Οι κατηγοριοποιητές ΜΔΣ εκτός από δυαδική κατηγοριοποίηση μπορούν να συνδιαστούν και στην περίπτωση που έχουμε πολλές κλάσεις. Ένας μεγάλος στόχος σχετικά με τις ΜΔΣ έχει να κάνει με τη βελτίωση της ταχύτητας στην εκπαίδευση και τη δοκιμή έτσι ώστε οι ΜΔΣ να γίνουν πιο εφικτή επιλογή και για μεγάλα σετ δεδομένων. Άλλα θέματα περιλαμβάνουν τον προσδιορισμό του καλύτερου πυρήνα για δεδομένο σετ και την εύρεση πιο αποτελεσματικών μεθόδων για την περίπτωση των πολλαπλών κλάσεων.

2.3.5 Κατηγοριοποίηση με βάση τους k εγγύτερους γείτονες (k-Nearest Neighbors)

Τα μοντέλα κατηγοριοποίησης που έχουμε περιγράψει μέχρι στηγμής είναι όλα μοντέλα πρόθυμης όπως λέγεται μάθησης. Αυτό σημαίνει ότι όταν δίνεται ένα σύνολο πλειάδων για εκπαίδευση θα κατασκευαστεί ένα γενικευμένο μοντέλο πρωτού γίνουν δεκτές νέες πλειάδες προς κατηγοριοποίηση. Μπορούμε να θεωρήσουμε ότι το μοντέλο είναι έτοιμο και πρόθυμο να κατηγοριοποιήσει τις άγνωστες μέχρι πριν πλειάδες. Αντιθέτως στην οκνηρή προσέγγιση περιμένουμε μέχρι την τελευταία στιγμή ,πρωτού γίνει κατασκευή οποιουδήποτε μοντέλου, να κατηγοριοποιηθεί το σύνολο των δοκιμαστικών πλειάδων. Ένας οκνηρός λοιπόν μαθητευόμενος με δεδομένη μια εκπαιδευόμενη πλειάδα απλά την αποθηκεύει, ή κάνει μια πολύ μικρή επεξεργασία της, και περιμένει μέχρι να λάβει μια δοκιμαστική πλειάδα. Μόνο μόλις δει τη δοκιμαστική εφαρμόζει γενίκευση για να

κατηγοριοποιηθεί με βάση την ομοιότητά της με τις αποθηκευμένες εκπαιδευμένες πλειάδες.

Ανόμοια λοιπόν με τις πρόθυμες μεθόδους μάθησης, οι οκνηροί μαθητευόμενοι κάνουν λιγότερη δουλειά όταν εμφανίζεται μια πλειάδα προς εκπαίδευση και περισσότερη όταν είναι να γίνει κατηγιριοποίηση ή αριθμητική πρόβλεψη. Επειδή λοιπόν οι οκνηροί μαθητευόμενοι αποθηκεύουν τις πλειάδες ή αλλιώς στιγμιότυπα αναφέρονται και ως μαθητευόμενοι βασιμένοι σε στιγμιότυπα,διότι σε αυτά βασίζεται όλη η διαδικασία της μάθησης. Για να γίνει μια κατηγοριοποίηση ή μια αριθμητική πρόβλεψη η χρήση οκνηρών μαθητευόμενων μπορεί να είναι υπολογιστικά ακριβή. Απαιτούν αποδοτικές τεχνικές αποθήκευσης και θα πρέπει να προσαρμόζονται σε εφαρμογές με υλικό που λειτουργεί με παράλληλοποίηση, ενώ προσφέρουν πολύ λίγη εξήγηση για την εσωτερική δομή των δεδομένων. Παρόλα αυτά όμως υποστηρίζουν από τη φύση τους τη στοιχειώδη εκπαίδευση. Μπορούν να μοντελοποιήσουν πολύπλοκους χώρους αποφάσεων που έχουν υπερπολυγωνικά σχήματα ,τα οποία δεν είναι εύκολα περιγράψιμα από άλλους αλγόριθμους.

Η πιο κοινή βασισμένη σε στιγμιότυπα μέθοδος χρησιμοποιεί τον αλγόριθμο των k-εγγύτερων γειτόνων. Ο αλγόριθμος αυτός υποθέτει ότι όλα τα στιγμιότυπα απεικονίζονται σε σημεία του n-διάστατου χώρου $\mathcal{R}^n$. Οι εγγύτεροι γείτονες ενός στιγμιοτύπου ορίζονται με την έννοια της Ευκλείδιας απόστασης. Πιο συγκεκριμένα ένα αυθαίρετο σημείο x μπορεί να περιγραφεί από ένα διάνυσμα $\langle a_1(x), a_2(x), \dots\ a_n(x)\rangle$ όπου το $a_r(x)$ δείχνει την τιμή του r-ιοστού χαρακηριστικού του σημείου x. Η απόσταση δύο σημείων x_i , x_j ορίζεται ως $d(x_i , x_j)$ και ισούται με $d(x_i , x_j) = \sqrt{\sum_{r=1}^{n}(a_r(x_i) - a_r(x_j))^2}$. Τυπικά κανονικοποιούμε τις τιμές κάθε χαρακτηριστικού διότι βοηθά στην αποτροπή κάποιων χαρακτηριστικών με μεγάλο εύρος να υπερτερούν κάποιων άλλων με μικρότερο εύρος.Η κανονικοποίηση ελαχίστου-μεγίστου για παράδειγμα (min-max) μπορεί να χρησιμοποιηθεί για να μετασχηματίσει μια τιμή ν ενός αριθμητικού χαρακτηριστικού Α σε ν' με εύρος [0,1] ως εξής: $\nu' = \frac{\nu - min_A}{max_A - min_A}$, όπου min_A και max_A είναι η ελάχιστη και μέγιστη τιμή του χαρακτηριστικού Α. Να σημειωθεί εδώ ότι η επιλογή της μετρικής απόστασης μπορεί να είναι κρίσιμη. Η απόσταση Manhattan ή και άλλες μετρικές χρησιμοποιούνται κι αυτές αρκετά συχνά.

Στη μέθοδο των εγγύτερων γειτόνων η συνάρτηση στόχος μπορεί να είναι είτε διακριτής τιμής είτε πραγματικής τιμής. Ας υποθέσουμε μια συνάρτηση στόχο διακριτής τιμής της μορφής $f: \mathcal{R}^n \rightarrow V$, όπου V είναι το πεπερασμένο σύνολο $\{v_1, v_2, \dots v_s\}$. Ο αλγόριθμος των k-εγγύτερων γειτόνων φαίνεται παρακάτω σε ψευδοκώδικα. Η τιμή $f(x_q)$ που επιστρέφεται από

Training algorithm:

- For each training example $\langle x, f(x)\rangle$, add the example to the list *training_examples*

Classification algorithm:

- Given a query instance x_q to be classified,
  - Let $x_1 \dots x_k$ denote the k instances from *training_examples* that are nearest to x_q
  - Return

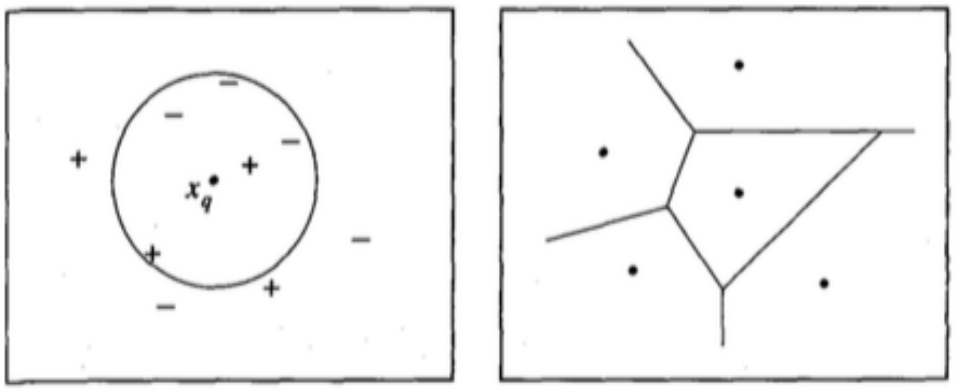
$$\hat{f}(x_q) \leftarrow \underset{v \in V}{\operatorname{argmax}} \sum_{i=1}^{k} \delta(v, f(x_i))$$

where $\delta(a, b) = 1$ if $a = b$ and where $\delta(a, b) = 0$ otherwise.

τον αλγόριθμο ως η εκτίμηση του $f(x_q)$ είναι απλά η πιο κοινή τιμή του f ανάμεσα στα k παραδείγματα που χρησιμοποιήθηκαν για εκπαίδευση και είναι κοντινά στο x_q. Αν διαλέξουμε k=1 τότε ο αλγόριθμος του 1-εγγύς γείτονα παραθέτει στο $\hat{f}(x_q)$ την τιμή $f(x_i)$, όπου x_i είναι το στιγμιότυπο που εκπαιδεύτηκε και βρίσκεται πιο κοντά στο x_q . Για

μεγαλύτερες τιμές του k ο αλγόριθμος αναθέτει την πιο κοινή τιμή ανάμεσα στα k κοντινότερα εκπαιδευόμενα παραδείγματα.

Στην εικόνα παρακάτω φαίνεται η λειτουργία του αλγορίθμου των k-εγγύτερων γειτόνων για την περίπτωση που τα στιγμιότυπα είναι σημεία του 2-διάστατου χώρου και η συνάρτηση στόχος παίρνει τιμές boolean. Τα θετικά και τα αρνητικά εκπαιδευόμενα παραδείγματα φαίνονται με "+" και "-". Να σημειωθεί ότι ο αλγόριθμος 1-εγγύς γείτονα κατηγοριοποιεί το x_q ως θετικό παράδειγμα ενώ των 5-εγγύτερων γειτόνων ως αρνητικό.



Εικόνα 2.8. Παράδειγμα κατηγοριοποίησης με τον αλγόριθμο k-εγγύτερων γειτόνων σε 2-διάστατο χώρο

Ο αλγόριθμος των k-εγγύτερων γειτόνων ποτέ δεν διαμορφώνει μια ρητή γενική υπόθεση $\hat{f}$ σχετική με τη συνάρτηση στόχο f. Απλά υπολογίζει την κατηγοριοποίηση κάθε νέου στιγμιοτύπου-ερωτήματος όταν χρειαστεί. Ακόμη μπορούμε να ρωτήσουμε ποια είναι η γενική συνάρτηση που υπονοείται, ή ποιες από τις κατηγοριοποιήσεις θα αναθέτονταν εάν κρατούσαμε τα εκπαιδευόμενα παραδείγματα σταθερά και κάναμε ερώτηση στον αλγόριθμο με κάθε πιθανό στιγμιότυπο στο X. Ο αλγόριθμος υιοθετείται εύκολα και για την προσέγγιση συναρτήσεων στόχων συνεχών τιμών.Για να επιτευχθεί αυτό θα πρέπει να υπολογίσουμε τη μέση τιμή των των k εγγύτερων εκπαιδευόμενων παραδειγμάτων, αντί να υπολογίσουμε την πιο κοινή τιμή τους. Πιο συγκεκριμένα για μια συνάρτηση πραγματικής τιμής της μορφής f: $\mathcal{R}^n \rightarrow \mathcal{R}$ αντικαθιστούμε την τελική γραμμή με $\hat{f}(x_q) \leftarrow \frac{\sum_{i=1}^{k} f(x_i)}{k}$.

Μια προφανής βελτίωση στον αλγόριθμο είναι η στάθμιση της συνεισφοράς του καθενός από τους k γείτονες σύμφωνα με την απόσταση από το σημείο αξιολόγησης x_q ,δίνοντας μεγαλύτερο βάρος στους πιο κοντινούς.Για παράδειγμα στον ψευδοκώδικα που δόθηκε παραπάνω και προσεγγίζει συναρτήσεις στόχους διακριτών τιμών μπορούμε να σταθμίσουμε την ψήφο του κάθε γείτονα ανάλογα με το αντίστροφο τετράγωνο της απόστασής του από το x_q. Αυτό επιτυγχάνεται με αντικατάσταση της τελευταίας γραμμής του αλγορίθμου με $\hat{f}(x_q) \leftarrow argmax_{v \in V} \sum_{i=1}^{k} w_i \delta(v, f(x_i))$, όπου $w_i = \frac{1}{d(x_q, x_i)^2}$. Για να καλυφθεί και η περίπτωση όπου το σημείο x_q ταιριάζει ακριβώς σε ένα από τα εκπαιδευόμενα στιγμιότυπα x_i και ο παρονομαστής είναι $d(x_q, x_i)^2$ είναι μηδέν τότε αναθέτουμε στο $\hat{f}(x_q)$ να ισούται με το $f(x_i)$. Αν υπάρχουν πολλά τέτοια εκπαιδευόμενα παραδείγματα αναθέτουμε την πλειοψηφική κατηγοριοποίηση μεταξύ αυτών.

Με παρόμοιο τρόπο μπορούμε να σταθμίσουμε τα στιγμιότυπα ανάλοα με την απόστασή τους και για συναρτήσεις στόχους πραγματικών τιμών αντικαθιστώντας ξανά την τελευταία γραμμή του ψευδοκώδικα του αλγορίθμου με $\hat{f}(x_q) \leftarrow \frac{\sum_{i=1}^{k} w_i f(x_i)}{\sum_{i=1}^{k} w_i}$, όπου το w_i είναι όπω ορίστηκε πιο πάνω. Να σημειωθεί ότι ο παρονομαστής είναι μια σταθερά που κανονικοποιεί τους γείτονες που συμβάλλουν με τα διάφορα βάρη. Όλες αυτές οι παραλλαγές του αλγορίθμου των k-εγγύτερων γειτόνων λαμβάνουν υπ'όψιν μόνο τους k

κοντινότερους γείτονες-σημεία ώστε να γίνει η κατηγοριοποίηση. Μόλις προστεθούν τα βάρη με βάση τις αποστάσεις δεν υπάρχει βλάβη στο να αφήσουμε όλα τα εκπαιδευόμενα παραδείγματα να έχουν επιρροή στην κατηγοριοποίηση του x_q ,αφού τα πολύ μακρινά παραδείγματα θα έχουν ελάχιστη επιρροή στον παράγοντα $\hat{f}(x_q)$.Το μόνο μειονέκτημα είναι ότι αν λάβουμε υπ'όψιν όλα τα δείγματα τότε ο κατηγοριοποιητής μας θα είναι πιο αργός. Εάν λάβουμε υπ'όψιν όλα τα δείγματα κατά την κατηγοριοποίηση ενός νέου στιγμιοτύπου ονομάζουμε τη μέθοδο καθολική, ενώ αν λάβουμε υπ'όψιν μόνο τα πιο κοντινά την ονομάζουμε τοπική.

Στις προηγούμενες περιπτώσεις θεωρήσαμε ότι τα δεδομένα που έχουμε είναι αριθμητικά. Όταν τα δεδομένα μας είναι ονομαστικά ή κατηγοριακά μια απλή μέθοδος είναι να συγκρίνουμε απευθείας τις τιμές του χαρακτηριστικού που ζητάμε της πλειάδας X_1 με αυτήν της πλειάδας X_2. Αν αυτές είναι ίδιες τότε η διαφορά μεταξύ των δύο είναι 0 αλλιώς αν είναι διαφορετικές ισούται με 1. Γενικά αν η τιμή ενός δεδομένου χαρακτηριστικού Α λείπει από μια πλειάδα X_1 ή/και από μια πλειάδα X_2 υποθέτουμε τη μέγιστη πιθανή διαφορά. Αν πούμε ότι κάθε ονομαστικό χαρακτηριστικό αντιστοιχίζεται με μια τιμή με εύρος [0,1] τότε παίρνουμε τη διαφορά ίση με 1 εάν κάποια ή και οι δύο τιμές των πλειάδων X_1 και X_2 λείπουν. Αν το Α είναι αριθμητικό και λείπει κι από τις δύο πλειάδες τότε ξανά η διαφορά ισούται με 1. Αν όμως μόνο μία από τις τιμές λείπει και η άλλη είναι παρούσα και κανονικοποιημένη (ν') τότε η διαφορά ισούται είτε με |1-ν'| είτε |0-ν'|, όποια από αυτές είναι μεγαλύτερη.

Ο αριθμός k των κοντινότερων γειτόνων προσδιορίζεται πειραματικά. Ξεκινάμε με k=1 και χρησιμοποιούμε ένα δοκιμαστικό σύνολο για την εκτίμηση του ποσοστού σφάλματος του κατηγοριοποιητή. Η διαδικασία επαναλαμβάνεται κάθε φορά αυξάνοντας το k επιτρέποντας έναν ακόμα γείτονα. Τελικά επιλέγεται αυτή η τιμή που δίνει το μικρότερο ποσοστό σφάλματος.Γενικά όσο μεγαλύτερος είναι ο αριθμός των δοκιμαστικών πλειάδων τόσο μεγαλύτερο θα είναι το k.Όσο ο αριθμός των πλειάδων τείνει στο άπειρο και k=1 το ποσοστό σφάλματος δε μπορεί να είναι χειρότερο από το διπλάσιο του ποσοστού σφάλματος Bayes,όπου το τελευταίο είναι το θεωρητικό ελάχιστο.Εάν επιλέον και το k τείνει στο άπειρο τότε το ποσοστό σφάλματος πλησιάζει αυτό του Bayes.

Ο σταθμισμένος με βάση την απόσταση αλγόριθμος των k-εγγύτερων γειτόνων είναι μια πολύ αποδοτική και επαγωγικού συμπερασμού μέθοδος για πολλά πρακτικά προβλήματα. Είναι συμπαγής σε δεδομένα που περιέχουν θόρυβο και αρκετά αποτελεσματική όταν προσφέρεται ένα επαρκώς μεγάλο σύνολο δεδομένων. Παίρνοντας το σταθμισμένο μέσο των k γειτόνων κοντά στο ζητούμενο σημείο μπορεί να εξομαλυνθεί ο αντίκτυπος των απομονωμένων και θορυβωδών δειγμάτων. Το επαγωγικό συμπέρασμα του αλγορίθμου προκύπτει από την υπόθεση ότι η κατηγοριοποίηση ενός δείγματος x_q θα είναι πιθανώς παρόμοια με την κατηγοριοποίηση των άλλων δειγμάτων που είναι κοντά του με βάση την ευκλείδια απόσταση.

Ένα πρακτικό ζήτημα που εφαρμόζεται στον αλγόριθμο είναι πως η απόσταση μεταξύ των δειγμάτων υπολογίζεται βασιζόμενη σε όλα τα χαρακτηριστικά του δείγματος, σε αντίθεση με μεθόδους που επιλέγουν μόνο ένα υποσύνολο των χαρακτηριστικών.Κατά συνέπεια η μετρική ομοιότητας που χρησιμοποιείται στους k-εγγύτερους γείτονες εξαρτάται κι από άσχετα χαρακτηριστικά οπότε μπορεί να είναι παραπλανητική.Η δυσκολία λοιπόν που προκύπτει λόγω του ότι αυτά τα χαρακτηριστικά είναι παρόντα αποκαλείται "κατάρα της διάστασης" και η τεχνική των εγγύτερων γειτόνων είναι ευαίσθητη σε αυτό το πρόβλημα.

Μια ενδιαφέρουσα προσέγγιση για να ξεπεραστεί αυτό το θέμα είναι να σταθμιστεί το κάθε χαρακτηριστικό διαφορετικά όταν υπολογίζεται η απόσταση μεταξύ δειγμάτων.Αυτό περιλαμβάνει το τέντωμα των αξόνων στον ευκλείδιο χώρο, με το να κοντύνουμε τους άξονες που αντιστοιχούν σε λιγότερο σχετικά χαρακτηριστικά και να επιμηκύνουμε τους άξονες για τα πιο σχετικά χαρακτηριστικά.Το κατά πόσο θα πρέπει ο κάθε άξονας να επιμηκυνθεί μπορεί να καθοριστεί αυτόματα με μια προσέγγιση διασταυρωμένης επικύρωσης (cross-validation). Αρχικά θέλουμε να επιμηκύνουμε-πολλαπλασιάσουμε τον j

άξονα κατά κάποιο παράγοντα z_j , όπου οι τιμές z_1 … z_n επιλέγονται για να ελαχιστοποιήσουν το πραγματικό σφάλμα κατηγοριοποίησης του αλγόριθμου εκμάθησης. Αυτό το σφάλμα μπορεί να εκτιμηθεί χρησιμοποιώντας διασταυρωμένη επικύρωση.

Έτσι ένας αλγόριθμος θα ήταν να επιλεγεί τυχαίο υποσύνολο των διαθέσιμων δεδομένων για να χρησιμοποιηθεί ως εκπαιδευόμενα δείγματα και μετά να καθοριστούν οι τιμές z_1 … z_n που οδηγούν στο ελάχιστο σφάλμα κατά την κατηγοριοποίηση των εναπομείναντων δειγμάτων. Επαναλαμβάνοντας αυτή τη διαδικασία αρκετές φορές η εκτίμηση για τους σταθμισμένους παράγοντες μπορεί να γίνει πιο ακριβής. Αυτός ο τρόπος λοιπόν με την επιμήκυνση των αξόνων με σκοπό τη βελτιστοποίηση της απόδοσης των k-εγγύτερων γειτόνων δίνει ένα μηχανισμό καταστολής στον αντίκτυπο των άσχετων χαρακτηριστικών.

Μια ακόμη πιο δραστική εναλλακτική είναι να αποκλείσουμε εντελώς τα λιγότερο σχετικά χαρακτηριστικά από το χώρο. Αυτό ισοδυναμεί με το να θέσουμε κάποια από τα z_i ίσα με το μηδέν. Οι Moore και Lee συζήτησαν το 1994 για αποδοτικές μεθόδους διασταυρωμένης επικύρωσης για την επιλογή σχετικών υποσυνόλων χαρακτηριστικών για τον αλγόριθμο που μελετάμε. Συγκεκριμένα ανακάλυψαν μεθόδους που βασίζονται στην αφήνω-ένα-εκτος διασταυρωμένη επικύρωση,όπου το σύνολο των m εκπαιδευόμενων δειγμάτων επανειλλημένα διαιρείται σε ένα σύνολο μεγέθους m-1 προς εκπαίδευση και σε ένα σύνολο μεγέθους 1 για δοκιμή με όλους τους δυνατούς τρόπους.Η τεχνική αυτή μπορεί να επεκτείνει εύκολα τον αλγόριθμο των k-εγγύτερων γειτόνων αφού δεν χρειάζεται επιπλέον εκπαίδευση και προσπάθεια κάθε φορά που το εκπαιδευόμενο σύνολο επαναπροσδιορίζεται.

Οι δύο τελευταίες προσεγγίσεις μπορούν να συμπεριληφθούν στην επιμήκυνση του κάθε άξονα κατά ένα σταθερό παράγοντα. Εναλλακτικά θα μπορούσαμε να επιμηκύνουμε κάθε άξονα κατά μια τιμή που ποικίλλει μέσα στο χώρο.Όμως όσο οι βαθμοί ελευθερίας ,που είναι διαθέσιμοι στον αλγόριθμο, αυξάνονται για τον επαναπροσδιορισμό της μετρικής της απόστασης με τον ίδιο τρόπο αυξάνεται και το ρίσκο της υπερπροσαρμογής.

Ένα επιπλέον ζήτημα πρακτικής σημασίας είναι η αποδοτική δεικτοδότηση(indexing) της μνήμης. Αφού ο αλγόριθμος καθυστερεί όλη την επεξεργασία μέχρι να ληφθεί ένα το νέο ερώτημα απαιτούνται σημαντικοί υπολογισμοί για αυτό.Πολλοί μέθοδοι αναπτυχθηκαν για τη δεικτοδότηση των αποθηκευμένων εκπαιδευόμενων δειγμάτων έτσι ώστε οι κοντινότεροι γείτονες να αναγνωρίζονται πιο αποδοτικά με λίγο επιπλέον κόστος σε μνήμη. Μια τέτοια μέθοδος είναι και τα kd-δέντρα στα οποία τα στιγμιότυπα αποθηκεύονται στα φύλλα ενός δέντρου με κοντινά στιγμιότυπα να αποθηκεύονται στο ίδιο ή σε κοντινούς κόμβους.Οι εσωτερικοί κόμβοι του δέντρου ταξινομούν το νέο στιγμιότυπο x_q σε ένα σχετικό φύλλο με το να τεστάρουν επιλεγμένα χαρακτηριστικά του x_q.

Όσον αφορά τη χρονική πολυπλοκότητα ο αλγόριθμος των k-εγγύτερων γειτόνων μπορεί να είνα εξαιρετικά αργός όταν κατηγοριοποιεί δοκιμαστικές πλειάδες. Αν D είναι μια εκπαιδευόμενη βάση με |D| πλειάδες και k=1 τότε απαιτούνται O(|D|) συγκρίσεις για να κατηγοριοποιηθεί μια δοκιμαστική πλειάδα.Με το να προταξινομήσουμε και να τακτοποιήσουμε τις αποθηκευμένες πλειάδες σε δέντρα αναζήτησης ο αριθμός των συγκρίσεων μπορεί να μειωθεί σε O(log|D|).Με παράλληλη επέκταση μπορούμε να μειώσουμε το χρόνο εκτέλεσεσης σε σταθερό,δηλαδή σε O(1) ανεξάρτητο από το |D|. Άλλες τεχνικές για να επιταχύνουμε το χρόνο κατηγοριοποίησης περιλαμβάνουν τη χρήση μερικού υπολογισμού της απόστασης και επεξεργασία των αποθηκευμένων πλειάδων. Στη μέθοδο της μερικής απόστασης υπολογίζουμε την απόσταση με βάση ένα υποσύνολο n των χαρακτηριστικών. Αν αυτή η απόσταση ξεπερνά ένα κατώφλι τότε σταματά ο επιπλέον υπολογισμός για τη δεδομένη πλειάδα και η διαδικασία προχωρά στην επόμενη. Η μέθοδος της επεξεργασίας απομακρύνει αυτές τις πλειάδες που αποδεικνύονται αχρείαστες. Η τελευταία μέθοδος ονομάζεται κλάδεμα ή συμπύκνωση γιατί ελαττώνει το συνολικό αριθμό των αποθηκευμένων πλειάδων.

2.4 Συνδιαστικοί Αλγόριθμοι και Μετα-αλγόριθμοι(Ensembles)

Ένας κατηγοριοποιητής ονομάζεται ασταθής αν μικρές διαταραχές στο σετ δεδομένων έχουν ως αποτέλεσμα μεγάλες αλλαγές στην πρόβλεψη ή στο σύνορο της απόφασης. Οι κατηγοριοποιητές υψηλής διακύμανσης είναι εγγενώς ασταθείς αφού τείνουν στο να υπερπροσαρμόζονται στα δεδομένα.Από την άλλη οι μέθοδοι με υψηλό συστηματικό σφάλμα τυπικά υποεφαρμόζουν στα δεδομένα και συνήθως έχουν χαμηλή διακύμανση. Και στις δύο περιπτώσεις ο σκοπός της εκμάθησης είναι η μείωση του σφάλματος κατηγοριοποίησης με τη μείωση της διακύμανσης ή του συστηματικού σφάλματος, ιδανικά και των δύο. Οι μέθοδοι ensemble δημιουργούν ένα συνδυαστικό κατηγοριοποιητή χρησιμοποιώντας την έξοδο πολλών βασικών κατηγοριοποιητών,οι οποίοι εκπαιδεύονται πάνω σε διαφορετικά σετ δεδομένων.Με εξάρτηση στο πως τα σετ δεδομένων επιλέγονται και στη σταθερότητα των βασικών κατηγοριοποιητών,οι ensemble κατηγοριοποιητές μπορούν να βοηθήσουν στη μείωση της διακύμανσης και του συστηματικού σφάλματος και να οδηγήσουν σε καλύτερη συνολική απόδοση.

Στην εξόρυξη δεδομένων ένα μοντέλο που παράγεται από αλγόριθμους μηχανικής μάθησης μπορεί να θεωρηθεί ως ειδικό. Το πόσο ειδικό όμως εξαρτάται από την ποσότητα και την ποιότητα των εκπαιδευόμενων δεδομένων και από το εάν ο αλγόριθμος εκμάθησης είναι κατάλληλος για το πρόβλημα, ειδάλλως ο ειδικός μπορεί στην πραγματικότητα να αποδειχθεί αδαής. Πολλά υποσχόμενες τέτοιες μέθοδοι είναι το bagging και το boosting. Αυτά αναπτύχθηκαν τις τελευταίες δεκαετίες και η απόδοσή τους ήταν εξαιρετική. Οσο οι ερευνητές στον τομέα της μηχανικής μάθησης πάσχιζαν να καταλάβουν το λόγο καινούργιες μέθοδοι προέκυψαν που πολλές φορές απέδιδαν ακόμη καλύτερα. Με πιο κοντινή ανάλυση αποκαλύπτεται ότι η τεχνική της ενίσχυσης(boosting),ίσως η πιο αποτελεσματική από τις 2,συνδέεται στενά με την καθιερωμένη στατιστική τεχνική των προσθετικών μοντέλων και οδηγεί σε βελτιωμένες διαδικασίες.

Όλα αυτά τα συδιαστικά μοντέλα μοιράζονται το μειονέκτημα ότι είναι πολύ δύσκολο να αναλυθούν. Είναι δυνατόν να περιλαμβάνουν μέχρι και χιλιάδες μεμονωμένα μοντέλα και παρόλο που λειτουργούν καλά δεν είναι εύκολο να γίνει κατανοητό σε βάθος ποιοι παράγοντες συμβάλλουν στις βελτιωμένες αποφάσεις.Τα τελαυταία χρόνια έχουν αναπτυχθεί μέθοδοι που συνδιάζουν τα οφέλη των επιδόσεων με κατανοητά μοντέλα. Καποια παράγουν συγκεκριμένα δενδρικά μοντέλα ενώ άλλα εισάγουν παραλλαγές δέντρων που προσφέρουν εναλλακτικά μονοπάτια.

2.4.1 Bagging

Με συνδιασμό των αποφάσεων από διαφορετικά μοντέλα σημαίνει ότι γίνεται συγχώνευση των διάφορων εξόδων σε μια μοναδική πρόβλεψη. ο πιο απλός τρόπος να γινει αυτό στην περίπτωση κατηγοριοποίησης είναι να γίνει συμψηφισμός, πιθανόν σταθμισμένος, ή στην περίπτωση αριθμητικής πρόβλεψης να υπολογιστεί ο μέσος όρος, πιθανόν κι εδώ σταθμισμένος. Κια οι δυο τεχνικές που θα περιγράψουμε και το bagging και το boosting υιοθετούν αυτή την προσέγγιση αλλά εξάγουν τα μεμονωμένα μοντέλα με διαφορετικό τρόπο. Στο bagging τα μοντέλα έχουν τα ίδια βάρη, ενώ στο boosting η στάθμιση χρησιμοποιείται για να δοθεί μεγαλύτερη επιρροή από το πιο επιτυχημένο από αυτά.

Ας υποθέσουμε ότι μερικά σετ δεδομένων του ίδιου μεγέθους επιλέγονται τυχαία από το πεδίο ορισμού που ορίζεται στο πρόβλημα. Χρησιμοποιώντας μια τεχνική μηχανικής μάθησης κατασκευάζουμε το δέντρο αποφάσεων του προβλήματος. Θα περιμέναμε τα δέντρα να είναι πρακτικά όμοια και ν ακάνουμε την ίδια πρόβλεψη για το καθένα καινούργιο στιγμιότυπο.Όμως η υπόθεση αυτή είναι συνήθως λανθασμένη ειδικά όταν τα εκπαιδευόμενα σετ δεδομένων είναι μικρά. Ο λόγος έγγυται στο ότι ο συμπερασμός μέσω των δέντρψν αποφάσεων,που είναι top-down μέθοδος, είναι μια ασταθής διαδικασία.

Μικρές αλλαγές στα δεδομένα μπορεί εύκολα να αποφέρει την επιλογή άλλου χαρακτηριστικού σε ένα κόμβο με σημαντικές προεκτάσεις για τη δομή του υποδέντρου κάτω από αυτό τον κόμβο.Αυτό αυτομάτως υπονοεί ότι υπάρχουν δοκιμαστικά στιγμιότυπα για τα οποία κάποια από τα δέντρα καταλήγουν σε σωστή πρόβλεψη και κάποια όχι.

Για να εκφράσουμε μέσω των μαθηματικών την τεχνική του bagging, θα υποθέσουμε ότι από τα δεδομένα εισόδου D δημιουργούμε πολλά ελαφρώς διαφορετικά εκπαιδευόμενα σετ D_i ,i=1,2,…,k με πολλά δείγματα το καθένα. Επίσης έχουμε διαφορετικούς βασικούς μαθητευόμενους κατηγοριοποιητές M_i ,με τον M_i να εκπαιδεύεται στο D_i. Με δεδομένο ένα δοκιμαστικό σημείο x αυτό κατηγοριοποιείται πρώτα με χρήση των k βασικών κατηγοριοποιητών M_i. Ο αριθμός των κατηγοριοποιητών που προβλέπουν την κλάση του x να είναι c_j δίνεται από τη σχέση: $v_j(x) = |\{M_i(x) = c_j \,|i = 1, \dots, k\}|$. Ο συνδυαστικός κατηγοριοποιητής που συμβολίζεται με M^K προβλέπει την πλειοψηφία της ψηφοφορίας μεταξύ των k κλάσεων και υπολογίζεται από τον τύπο: $M^K(x) = argmax_{c_j}\{v_j(x)|j = 1, \dots, k\}$. Για δυαδική κατηγοριοποίηση υποθέτουμε ότι οι κλάσεις δίνονται ως {+1,-1} και ο συνδυαστικός κατηγοριοποιητής M^K εκφράζεται πιο απλά ως: $M^K(x) = sign(\sum_{i=1}^{K} M_i(x))$.

Το αποτέλεσμα του συνδιασμού πολλαπλών υποθέσεων μπορεί να μελετηθεί από θεωρητική σκοπιά και συχνά αποκαλέιται αποσύνθεση πόλωσης-μεταβλητότητας(bias-variance decomposition).Ας υποθέσουμε ότι μπορούμε να έχουμε απεριόριστο αριθμό ανεξάρτητων σετ δεδομένων για εκπαίδευση του ίδιου μεγέθους και να τα χρησιμοποιούμε ώστε να φτιάχνουμε απεριόριστο αριθμό κατηγοριοποιητών. Ένα δοκιμαστικό στιγμιότυπο περνά από επεξεργασία από όλους τους κατηγοριοποιητές και η μοναδική απάντηση καθορίζεται από την πλειοψηφία της ψηφοφορίας.Σε αυτή την ιδανική κατάσταση τα σφάλματα μπορούν ακόμα να προκύψουν διότι κανένα σχήμα εκμάθησης δεν είναι τέλειο.Το ποσοστό σφάλματος εξαρτάται από το πόσο καλά η μέθοδος μηχανικής μάθησης ταιριάζει στο πρόβλημα, καθώς υπάρχει και η επιρροή του θορύβου στα δεδομένα που πιθανότατα δεν μπορεί να εκμαθευτεί.

Υποθέτουμε ότι το αναμενόμενο σφάλμα εκτιμάται με το μέσο όρο των σφαλμάτων του συνδιαστικού κατηγοριοποιητή πάνω στον απεριόριστο αριθμό των ανεξάρτητα επιλεγμένων δοκιμαστικών δειγμάτων. Το ποσοστό σφάλματος για συγκεκριμένο αλγόριθμο εκμάθησης ονομάζεται πόλωση(ή συστηματικό σφάλμα) αυτού για το πρόβλημα εκμάθησης και μετρά πόσο καλά η μέθοδος εκμάθησης ταιριάζει στο πρόβλημα.Ο θόρυβος συμπεριλαμβάνεται στον όρο αυτό αφού έτσι κι αλλιώς είναι άγνωστος στην πράξη.Αυτός ο τεχνικός ορισμός είναι ένας τρόπος ποσοτικοποίησης της αόριστης έννοιας της πόλωσης που έχει δοθεί μέχρι τώρα. Ουσιαστικά μετρά το συστηματικό σφάλμα του αλγόριθμου που δεν μπορεί να εξαλειφθεί ούτε εάν λάβουμε υπ'όψιν απεριόριστο αριθμό δειγμάτων.Φυσικά δε μπορεί να υπολογιστεί ακριβώς σε πραγματικές συνθήκες,απλά μόνο να προσεγγιστεί.

Μια δεύτερη αιτία σφαλμάτων σε μοντέλα εκμάθησης πηγάζει από το δεδομένο εκπαιδευόμενο σετ που χρησιμοποιείται το οποίο είναι αναπόφευκτα πεπερασμένο,αλλά όχι εντελώς αντιπροσωπευτικό του ακριβούς πληθυσμού στιγμιοτύπων. Η εκτιμώμενη τιμή αυτού του τμήματος του σφάλματος πάνω σε όλα τα δυνατά εκπαιδευόμενα σετ του δεδομένου μεγέθους και σε όλα τα δυνατά δοκιμαστικά σετ ονομάζεται μεταβλητότητα την μεθόδου εκμάθησης για αυτό το πρόβλημα. Το ολικό αναμενόμενο σφάλμα ενός κατηγοριοποιητή βρίσκεται από το άθροισμα της πόλωσης και της μεταβλητότητας κι αυτή ακριβώς είναι η αποσύνθεση πόλωσης-μεταβλητότητας.

Η αποσύνθεση πόλωσης-μεταβλητότητας εισήχθη στο πλαίσιο της αριθμητικής πρόβλεψης βασισμένη στο τετραγωνικό σφάλμα που είναι ένας ευρέως αποδεκτός τρόπος εφαρμογής της. Παρόλα αυτά η κατάσταση δεν είναι και τόσο καθαρή για την κατηγοριοποίηση και πολλές προτάσεις έχουν γίνει. Ανεξάρτητα από τη συγκεκριμένη αποσύνθεση που χρησιμοποιείται για την ανάλυση του του σφάλματος ο συνδυασμός πολλαπλών κατηγοριοποιητών γενικά μειώνει το αναμενόμενο σφάλμα με το να ελαττώνει τη συνιστώσα της μεταβλητότητας. Όσο περισσότεροι κατηγοριοποιητές περιλαμβάνονται

τόσο μεγαλύτερη η ελάττωση της μεταβλητότητας. Φυσικά προκύπτει μια δυσκολία όταν μπαίνει στην πράξη αυτό το σχήμα ψηφοφορίας που είναι ότι υπάρχει μόνο ένα σετ δεδομένων προς εκπαίδευση και η απόκτηση περισσότερων δεδομένων είναι είτε αδύνατη είτε ακριβή.

Η τεχνική του bagging προσπαθεί να ουδετεροποιήσει την αστάθεια των μεθόδων μάθησης με το να προσομοιώνει τη διαδικασία που περιγράφηκε χρησιμοποιώντας ένα δεδομένο εκπαιδευόμενο σύνολο. Αντί να δειγματοληπτείται ένα καινούργιο ανεξάρτητο εκπαιδευόμενο σετ κάθε φορά,μεταβάλλεταιτο αυθεντικό με τη διαγραφή κάποιων στιγμιοτύπων και την αντιγραφή κάποιων άλλων. Τα στιγμιότυπα δειγματοληπτούνται τυχαία με αντικατάσταση από το πρωτότυπο σετ για τη δημιουργία ενός καινούργιου ίδιου μεγέθους.Έτσι αυτή η διαδικασία αναγκαστικά αντιγράφει κάποια στιγμιότυπα και διαγράφει άλλα. Λόγω της μεθόδου που προσπαθεί να εκτιμήσει και να γενικεύσει το σφάλμα ο όρος bagging είναι συντομογραφία του bootstrap aggregating.

Η διαφορά μεταξύ του bagging και της εξιδανικευμένης διαδικασίας που περιγράφηκε νωρίτερα είναι ο τρόπος με τον οποίο τα εκπαιδευόμενα σύνολα προκύπτουν. Αντί να αποκτηθούν ανεξάρτητα σετ από το πεδίο ορισμού η τεχνική απλά δειγματοληπτεί τα αυθεντικά εκπαιδευόμενα σύνολα.Τα σύνολα που προκύπτουν από δειγματοληψία είναι διαφορετικά μεταξύ τους αλλά δεν είναι ανεξάρτητα γιατί όλα βασίζονται σε ένα σύνολο.Αποδεικνύεται ότι με το bagging παράγεται ένα συνδυαστικό μοντέλο που αποδίδει καλύτερα από το ένα μοντέλο που κατασκευάστηκε από τα πρωτότυπα δεδομένα και ποτέ χειρότερα.

Η τεχνική μπορεί να εφαρμοστεί σε σχήματα μάθησης για αριθμητική πρόβλεψη όπως για παράδειγμα δενδρικά μοντέλα.Η μόνη διαφορά είναι ότι αντί να γίνεται ψηφοφορία κατά το τέλος, στις μεμονομένες προβλέψεις, που είναι πραγματικοί αριθμοί, βρίσκουμε το μέσο όρο. Η αποσύνθεση πόλωσης-μεταβλητότητας εφαρμόζεται σε αριθμητικές προβλέψεις με την αποδόμηση της αναμενόμενης τιμής του μέσου τετραγωνικού σφάλματος των προβλέψεων των νέων δεδομένων.Η πόλωση ορίζεται ως το μέσο τετραγωνικό αναμενόμενο σφάλμα βρίσκοντας το μέσο όρο όλων των μοντέλων που κατασκευάζονται από όλα τα δυνατά εκπαιδευόμενα σετ του ίδιου μεγέθους ενώ η μεταβλητότητα είναι μέρος του αναμενόμενου σφάλματος του ενός μοντέλου από το συγκεκριμένο εκπαιδευόμενο σύνολο από το οποίο κατασκευάστηκε. Μπορεί να αποδειχθεί θεωρητικά ότι ο μέσος όρος απεριόριστα πολλών μοντέλων που κατασκευάστηκαν αναξάρτητα δείγματα πάντα μειώνει την αναμενόμενη τιμή του μέσου τετραγωνικού σφάλματος.Παρακάτω φαίνεται συνοπτικά ο αλγόριθμος bagging.

Model Generation

```
Let n be the number of instances in the training data.
For each of t iterations:
  Sample n instances with replacement from training data.
  Apply the learning algorithm to the sample.
  Store the resulting model.
```

Classification

```
For each of the t models:
  Predict class of instance using model.
Return class that has been predicted most often.
```

Όπως αναφέρθηκε πριν η τεχνική του bagging βοηθά περισσότερο στην περίπτωση που το σχήμα μάθησης είναι ασταθές και όταν μικρές αλλαγές στα δεδομένα εισόδου

οδηγούν τους κατηγοριοποιητές σε διαφορετικά αποτελέσματα.Τα αποτελέσματα αυτά μπορούν να βελτιωθούν αν αυξηθεί η ποικιλλία στους κατηγοριοποιητές κάνοντας το σχήμα όσο περισσότερο γίνεται ασταθές. Για παράδειγμα όταν χρησιμοποιείται το bagging σε δένδρα αποφάσεων τα οποία ήδη είναι αρκετά ασταθή επιτυγχάνεται καλύτερη απόδοση με το να μην χρησιμοποιείται κλάδεμα οπότε και υπάρχει ακόμα μεγαλύτερη αστάθεια.Μια άλλη βελτίωση μπορεί να γίνει με αλλαγή του τρόπου που συνδιάζονται οι προβλέψεις για την κατηγοριοποίηση.Όπως γνωρίζουμε στο bagging χρησιμοποιείται ψηφοφορία.Όταν όμως τα μοντέλα μπορεί να δώσουν ως έξοδο μια εκτίμηση πιθανότητας και όχι απλά μια κατηγοριοποίηση βγάζει νόημα να βρούμε το μέσο όρο αυτών των πιθανοτήτων. Το τελευταίο όχι μόνο βελτιώνει αισθητά την κατηγοριοποίηση αλλά έχουμε και εκτιμήσεις πιθανότητας που συχνά είναι πιο ακριβείς από αυτές που παρήγαγαν τα μοντέλα από μόνα τους.Οι επεκτάσεις του bagging χρησιμοποιούν αρκετά τη μέθοδο των συνδιαστικών προβλέψεων.

Ένα άλλο που μπορούμε να πετύχουμε είναι η κατασκευή κατηγοριοποιητών που είναι "ευαίσθητοι" στο κόστος,ελαχιστοποιώντας το αναμενόμενο κόστος των προβλέψεων.Οι ακριβείς εκτιμήσεις πιθανότητας είναι απαραίτητες διότι χρησιμοποιούνται για την απόκτηση του αναμενόμενου κόστους κάθε πρόβλεψης.Το bagging είναι ιδανικός υποψήφιος για ευαισθησία στο κόστος αφού παράγει πολύ ακριβείς εκτιμήσεις πιθανότητας από τα δέντρα αποφάσεων και άλλες ισχυρούς,όμως ασταθείς,κατηγοριοποιητές.Μια μέθοδος που ονομάζεται Μετα-κόστος(MetaCost) συνδυάζει τα οφέλη της πρόβλεψης του bagging με ένα κατανοητό μοντέλο για πρόβλεψη με ευαισθησία κόστους.Κατασκευάζει έναν κατηγοριοποιητή με bagging και τον αναπτύσσει ώστε να ξαναδωθεί ετικέτα στα εκπαιδευόμενα στιγμιότυπα την πρόβλεψη που που ελαχιστοποιεί το αναμενόμενο κόστος βασισμένο στις εκτιμήσεις πιθανότητας που αποκτήθηκαν από το bagging.

Στη συνέχεια η Metacost απορρίπτει τις ετικέτες της αυθεντικής κλάσης και μαθαίνει μόνο ένα νέο κατηγοριοποιητή από τα δεδομένα στα οποία μπήκε ξανά ετικέτα. Αυτό το νέο μοντέλο λαμβάνει αυτόματα υπ'όψιν τα κόστη διότι αυτά κατασκευάστηκαν μέσα στις ετικέτες των κλάσεων.Το αποτέλεσμα είναι ένας μοναδικός ευαίσθητος στο κόστος κατηγοριοποιητής που μπορεί να αναλυθεί και να δείξει πως έγιναν οι προβλέψεις.Επιπρόσθετα με την ευαίσθητη στο κόστος τεχνική που περιγράφηκε υπάρχει και μια ευαίσθητη στο κόστος μέθοδος μάθησης κατά την οποία γίνεται μάθηση ενός τέτοιου κατηγοριοποιητή αλλάζοντας τις αναλογίες της κάθε κλάσης στα δεδομένα εκπαίδευσης ώστε να αντικατοπτρίζουν τον πίνακα κόστους.Σε σύγκριση με αυτή την τεχνική το Metacost φαίνεται να παράγει πιο ακριβή αποτελέσματα αλλά χρειάζεται περισσότερους υπολογισμούς. Αν δεν υπάρχει κανένας λόγος για κατανοητό μοντέλο το βήμα μετεπεξεργασίας που κάνει το Metacost είναι περιττό. Είναι καλύτερα λοιπόν να χρησιμοποιούμε τον bagged κατηγοριοποιητή σε συνδυασμό με τη μέθοδο του ελάχιστου αναμενόμενου κόστους.

2.4.2 Boosting

Το bagging σαν τεχνική εκμεταλλεύεται την αστάθεια που χαρακτηρίζει τους αλγόριθμους εκμάθησης. Διαισθητικά συνδυάζοντας πολλαπλά μοντέλα βοηθά όταν αυτά τα μοντέλα είναι εμφανώς διαφορετικά μεταξύ τους και καθένα χειρίζεται ένα λογικό ποσοστό των δεδομένων με σωστό τρόπο. Ιδανικά τα μοντέλα αλληλοσυμπληρώνονται και το καθένα είναι ειδικό για το κομάτι του πεδίου ορισμού όπου τα άλλα μοντέλα δεν αποδίδουν πολύ καλά.

Το boosting είναι μια ακόμη τεχνική ensemble που εκπαιδεύει τους βασικούς κατηγοριοποιητές σε διαφορετικά δείγματα. Η κύρια ιδέα είναι η προσεκτική επιλογή των δειγμάτων για ενίσχυση της επίδοσης των δύσκολα κατηγοριοποιήσιμων στιγμιοτύπων. Ξεκινώντας από ένα αρχικό δείγμα προς εκπαίδευση D_1 εκπαιδεύουμε τον βασικό

κατηγοριοποιητή M_1 και παίρνουμε το ποσοστό σφάλματος της εκπαίδευσής του.Για να κατασκευάσουμε το επόμενο δείγμα D_2 επιλέγουμε τα λάθως κατηγοριοποιημένα στιγμιότυπα με τη μεγαλύτερ πιθανότητα και μετά από εκπαίδευση του M_2 παίρνουμε το δικό του ποσοστό σφάλματος από την εκπαίδευση.Για να κατασκευάσουμε το D_3 τα στιγμιότυπα που είναι δύσκολο να κατηγοριοποιηθούν από τον M_1 και τον M_2 έχουν μεγαλύτερη πιθανότητα να επιλεγούν.Η διαδικασία επαναλαμβάνεται για Κ φορές και αντίθετα με το bagging που χρησιμοποιεί ανεξάρτητα και τυχαία δείγματα από το αρχικό σετ δεδομένων το boosting επιλέγει σταθμισμένα-πολωμένα δείγματα για να κατασκευάσει τα διαφορετικά εκπαιδευόμενα σετ με το τρέχον δείγμα να εξαρτάται από τα προηγούμενα.Τελικά ο συνδυαστικός κατηγοριοποιητής προκύπτει μέσω σταθμισμένης ψηφοφορίας στην έξοδο των Κ βασικών κατηγοριοποιητών $M_1, M_2, \dots, M_K$.

Το boosting είναιπιο ωφέλιμο όταν οι βασικοί κατηγοριοποιητές είναι αδύναμοι,για παράδειγμα δηλαδή έχουν ένα ποσοσστό σφάλματος ελάιστα μικρότερο από τον τυχαίο κατηγοριοποιητή.Η ιδέα έγγυται στο ότι μπορεί ο M_1 να μην είναι ιδιαίτερα καλός σε όλα τα δοκιμαστικά στιγμιότυπα, κατασκευάζοντας τον M_2 μπορεί να βοηθά στην κατηγοριοποίηση κάποιων περιπτώσεων όπου ο M_1 αποτυγχάνει, ο M_3 μπορεί να βοηθά στην κατηγοριοποίηση όπου οι δύο προηγούμενοι απέτυχαν κ.ο.κ . Όμως το boosting δεν έχει μόνο επιρροή στη μείωση της πόλωσης.Καθένας από τους αδύναμους μαθητευόμενους είναι πιθανόν να έχει υψηλή αμεροληψία αλλά ο τελικός συνδυαστικός κατηγοριοποιητής μπορεί να έχει πολύ χαμηλότερη αμεροληψία αφού διαφορετικοί αδύναμοι μαθητευόμενοι μαθαίνουν να να κατηγοριοποιούν στιγμιότυπα σε διαφορετικές περιοχές του χώρου εισόδου.Μερικές παραλλαγές του boosting μπορούν να προκύψουν με βάση το πώς τα βάρη των στιγμιοτύπων υπολογίζονται για τη δειγματοληψία , τον τρόπο που οι κατηγοριοποιητές συνδυάζονται κλπ. Η πιο δημοφιλής παραλλαγή είναι το Adaboost,ή αλλιώς Adaptive Boosting δηλαδή προσαρμοσμένη ενίσχυση.

Κάποιες ομοιότητες με το bagging περιλαμβάνουν τη χρήση ψηφοφορίας για κατηγοριοποίηση ή μέσου όρου για αριθμητική πρόβλεψη στο συνδυασμό των ατομικών μοντέλων. Ακόμη συνδυάζονται μοντέλα του ίδιου τύπου,όπως και στο bagging, όπως για παράδειγμα δέντρα αποφάσεων.Οι διαφορές είναι ότι καταρχήν το boosting είναι επαναληπτικό. Ενώ στο bagging ατομικά μοντέλα κατασκευάζονται ξεχωριστά,στο boosting κάθε καινούργιο μοντέλο επηρρεάζεται από την επίδοση αυτών που κατασκευάστηκαν πιο πριν.Στο boosting ενθαρρύνεται τα καινούργια μοντέλα να γίνουν ειδικά για στιγμιότυπα που δεν χειρίστηκαν σωστά τα προηγούμενα οπότε και ανατίθονται μεγαλύτερα βάρη σε αυτά τα στιγμιότυπα.Μια άλλη διαφορά είναι ότι στο boosting σταθμίζεται η συμβολή ενός μοντέλου από τη σιγουριά του, αντί να δίνεται ίση τιμή σε όλα τα μοντέλα.

Ο αλγόριθμος AdaBoost που όπως είπαμε είναι ο πιο δημοφιλής και σχεδιάστηκε ειδικά για προβλήματα κατηγοριοποίησης χρησιμοποιείται πάρα πολύ στην πράξη.Όπως και αυτός του bagging μπορεί να εφαρμοστεί σε οποιοδήποτε αλγόριθμο εκμάθησης για κατηγοριοποίηση. Για να απλουστεύσουμε τα πράγματα υποθέτουμε ότι ο αλγόριθμος που χρησιμοποιούμε μπορεί να διαχειριστεί σταθμισμένα στιγμιότυπα,όπου το βάρος ενός στιγμιοτύπου είναι θετικός αριθμός.Η παρουσία των βαρών των στιγμιοτύπων αλλάζει τον τρόπο με τον οποίο υπολογίζεται το σφάλμα ενός κατηγοριοποιητή, που ισούται με το άθροισμα όλων των βαρών των λάθως κατηγοριοποιημένων στιγμιοτύπων διαιρεμένο με το άθροισμα των συνολικών βαρών όλων των στιγμιοτύπων,αντί γα το κλάσμα των των στιγμιοτύπων που κατηγοριοποιήθκαν λάθως.Με το να σταθμίζονται τα στιγμιότυπα ο αλγόριθμος εκμάθησης αναγκάζεται να εστιάσει σε συγκεκριμένο σύνολο από στιγμιότυπα και ειδικά σε αυτά με μεγάλα βάρη.Τέτοια στιγμιότυπα γίνονται εξαιρετικά γιατί υπάρχει μεγαλύτερο κίνητρο να κατηγοριοποιηθούν σωστά. Παρακάτω περιγράφεται συνοπτικά ο αλγόριθμος του boosting.

Model Generation

```
Assign equal weight to each training instance.
For each of t iterations:
  Apply learning algorithm to weighted dataset and store resulting
    model.
  Compute error e of model on weighted dataset and store error.
  If e equal to zero, or e greater or equal to 0.5:
    Terminate model generation.
  For each instance in dataset:
    If instance classified correctly by model:
       Multiply weight of instance by e / (1 - e).
  Normalize weight of all instances.
```

Classification

```
Assign weight of zero to all classes.
For each of the t (or less) models:
  Add -log(e / (1 - e)) to weight of class predicted by model.
Return class with highest weight.
```

Όπως περιγράφεται παραπάνω ο αλγόριθμος ξεκινά με την ανάθεση ίσων βαρών σε όλα τα στιγμιότυπα των επαιδευόμενων δεδομένων.Μετά καλέιται ο αλγόριθμος εκμάθησης να σχηματίσει τους κατηγοριοποιητές για τα συγκεκριμένα δεδομένα και ξανασταθμίζει το κάθε στιγμιότυπο ανάλογα με την έξοδο του κάθε κατηγοριοποιητή.Τα βάρη των σωστά κατηγοριοποιημένων στιγμιοτύπων μειώνονται και των λάθως κατηγοριοποιημένων αυξάνονται.Αυτό παράγει ένα σετ από εύκολα στιγμιότυπα με χαμηλά βάρη και ένα σετ από δύσκολα με υψηλά βάρη. Στην επόμενη επανάληψη και σε αυτές που θα ακολουθήσουν μετά κατασκευάζεται ένας κατηγοριοποιητής για τα ξανασταθμισμένα δεδομένα ο οποίος εστιάζει στο να κατηγοριοποιήσει τα δύσκολα στιγμιότυπα σωστά και τα βάρη αυξάνονται ή μειώνονται ανάλογα την έξοδο αυτού του κατηγοριοποιητή.Αυτό έχει ως αποτέλεσμα κάποια δύσκολα στιγμιότυπα να γίνουν ακόμα δυσκολότερα και κάποια εύκολα ακόμα ευκολότερα,ενώ από την άλλη πλευρά κάποια δύσκολα μπορεί να γίνουν ευκολότερα και κάποια εύκολα δυσκολότερα,όλες οι πιθανότητες αυτές εμφανίζονται στην πράξη.

Ύστερα από κάθε επανάληψη τα βάρη αντικατοπτρίζουν πόσο συχνά τα στιγμιότυπα κατηγοριοποιήθηκαν λάθως από τους κατηγοριοποιητές που παράχθηκαν μέχρι εκείνη τη στιγμή.Διατηρώντας ένα μέτρο δυσκολίας για κάθε στιγμιότυπο η διαδικασία αυτή προσφέρει ένα κομψό τρόπο παραγωγής μιας σειράς από ειδικούς κατηγοριοποιητές που αλληλοσυμπληρώνονται μεταξύ τους. Για το πόσο θα πρέπει να διαφοροποιούνται τα βάρη μετά από κάθε επανάληψη εξαρτάται από το συνολικό σφάλμα του τρέχοντος κατηγοριοποιητή. Συγκεκριμένα αν συμβολίσουμε με e το σφάλμα του κατηγοριοποιητή σε σταθμισμένα δεδομένα,που είναι ένα κλάσμα μεταξύ του 0 και 1, τότε τα βάρη ενημερώνονται με βάση τη σχέση: καινούργιο_βάρος $= \frac{\text{παλιό\_βάρος} * \text{e}}{1-e}$ για τα σωστά κατηγοριοποιημένα στιγμιότυπα ενώ τα βάρη των λάθως κατηγοριοποιημένων παραμένουν αμετάβλητα. Φυσικά αυτό δεν αυξάνει το βάρος των λάθως κατηγοριοποιημένων όπως αναφέρθηκε προηγουμένως. Όμως τελικά όλα τα βάρη αφού ενημερωθούν κανονικοποιούνται έτσι ώστε το άθροισμά τους να παραμένει το ίδιο όπως πριν.

Το βάρος καθενός στιγμιοτύπου διαιρείται με το άθροισμα των νέων βαρών και πολλαπλασιάζεται με το άθροισμα των παλιών.Έτσι αυτόματα αυξάνεται το βάρος των

λάθως κατηγοριοποιημένων στιγμιοτύπων και μειώνει αυτό των σωστά κατηγοριοποιημένων. Όταν το σφάλμα των σταθμισμένων εκπαιδευόμενων δεδομένων υπερβεί ή είναι ίσο με 0.5 η διαδικασία του boosting διαγράφει τον τρέχοντα κατηγοριοποιητή και δεν εφαρμόζει παρισσότερες επαναλήψεις.Το ίδιο συμβαίνει και όταν το σφάλμα είναι 0 διότι τότε όλα τα βάρη των στιγμιοτύπων θα είναι επίσης 0.Για να διαμορφωθεί μια πρόβλεψη οι έξοδοι των κατηγοριοποιητών συνδιάζονται με χρήση σταθμισμένης ψηφοφορίας.Για τον προσδιορισμό των βαρών ο κατηγοριοποιητής που απέδοσε καλά με τα σταθμισμένα εκπαιδευόμενα δεδομένα από τα οποία κατασκευάστηκε(το e πλησιάζει το 0)θα πρέπει να λάβει υψηλό βάρος ενώ σε αυτόν που δεν απέδοσε καλά(το e πλησιάζει το 0.5)θα πρέπει να λάβει χαμηλό βάρος.

Ο αλγόριθμος AdaBoost χρησιμοποιεί τη σχέση: βάρος=$-log\frac{e}{1-e}$ που είναι θετικός αριθμός μεταξύ του 0 και του απείρου. Έτσι εξηγείται λοιπόν ο λόγος που οι κατηγοριοποιητές που αποδίδουν τέλεια θα πρέπει να διαγραφούν ,γιατί όταν το e είναι 0 το βάρος είναι απροσδιόριστο.Για να γίνει λοιπόν πρόβλεψη τα βάρη όλων των κατηγοριοποιητών που συμμετέχουν στην ψηφοφορία για μια σηγκεκριμένη κλάση αθροίζονται και η κλάση με το μεγαλύτερο σύνολο επιλέγεται.Στην αρχή υποθέσαμε ότι ο αλγόριθμος εκμάθησης μπορεί να διαχειριστεί σταθμισμένα στιγμιότυπα όμως κάθε αλγόριθμος μπορεί να προσαρμοστεί ώστε να αντιμετωπίσει τέτοια στιγμιότυπα.Αντί να αλλάξουμε τον αλγόριθμο εκμάθησης όμως είναι δυνατό να παράγουμε ένα σετ δεδομένων χωρίς βάρη από τα σταθμισμένα δεδομένα με ξαναδειγματοληψία,δηλαδή την τεχνική που χρησιμοποιεί και το bagging.

Αντίθετα όμως με το bagging όπου κάθε στιγμιότυπο επιλέγεται με ίση πιθανότητα, στο boosting επιλέγεται αναλογικά με το βάρος του.Ως αποτέλεσμα στιγμιότυπα με μεγαλύτερο βάρος επαναλαμβάνοντια συχνά ενώ άλλα με χαμηλό βάρος μπορεί να μην επιλεγούν και ποτέ.Όταν το νέο σετ δεδομένων γίνει όσο μεγάλο είναι και το αρχικό τότε τροφοδοτείται στο σχήμα εκμάθησης αντί των σταθμισμένων βαρών.Ένα μειονέκτημα αυτής της διαδικασίας είναι ότι κάποια στιγμιότυπα με χαμηλό βάρος δεν ξαναπαρουσιάζονται στο σετ δεδομένων που ξαναδειγματοληπτείται οπότε χάνεται πληροφορία πριν καν εφαρμοστεί το σχήμα εκμάθησης.Παρόλα αυτά το γεγονός αυτό ίσως είναι πλεονέκτημα.Εάν το σχήμα εκμάθησης παράγει έναν κατηγοριοποιητή με σφάλμα που υπερβαίνει το 0.5,το boosting θα πρέπει να τερματιστεί αν τα σταθμισμένα δεδομένα χρησιμοποιηθούν απ ευθείας,ενώ με ξανά δειγματοληψία είναι πιθανό να παραχθεί κατηγοριοποιητής με σφάλμα χαμηλότερο του 0.5 απορρίπτοντας τον ξαναδειγματοληπτόμενο σετ και παράγοντας καινούργιο από διαφορετική και τυχαία σοδειά.Κάποιες φορές οι περισσότερες επαναλήψεις με boosting μπορούν να εφαρμοστούν με δειγματοληψία ξανά,παρά όταν χρησιμοποιείται η αυθεντική σταθμισμένη εκδοχή του αλγορίθμου.

Για να εκφράσουμε τον αλγόριθμο AdaBoost μέσω μαθηματικών σχέσεων ας υποθέσουμε ότι έχουμε ένα σετ δεδομένων εισόδου D που περιλαμβάνει n σημεία $x_i \in \mathcal{R}^d$. Η διαδικασία του boosting θα επαναληφθεί K φορές.Επίσης t είναι η επανάληψη ,α_t το βάρος για τον t-κατηγοριοποιητή M_t και w_i^tτο βάρος του x_i με $w^t = (w_1^t, w_2^t, \dots, w_n^t)^T$ το διάνυσμα βάρους πάνω από όλα τα σημεία για την t-ιοστή επανάληψη. Στην ουσία το w είναι το διάνυσμα πιθανότητας του οποίου το άθροισμα των σημείων είναι ίσο με 1.Αρχικά όλα τα σημεία έχουν ίσα βάρη,δηλαδή $w^0 = (\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n})^T = \frac{1}{n}1$, όπου $1 \in \mathcal{R}^n$ είναι το n-διάστατο διάνυσμα όλων των άσσων.Παρακάτω φαίνεται ο ψευδοκώδικα του αλγορίθμου AdaBoost.

AdaBoost(K,**D**):

$\mathbf{w}^0 \leftarrow \left(\frac{1}{n}\right) \cdot \mathbf{1} \in \mathbb{R}^n$

$t \leftarrow 1$

while $t \leq K$ **do**

  $\mathbf{D}_t \leftarrow$ weighted resampling with replacement from **D** using $\mathbf{w}^{t-1}$

  $M_t \leftarrow$ train classifier on $\mathbf{D}_t$

  $\epsilon_t \leftarrow \sum_{i=1}^{n} w_i^{t-1} \cdot I\big(M_t(\mathbf{x}_i) \neq y_i\big)$ // weighted error rate on **D**

  if $\epsilon_t = 0$ **then break**

  else if $\epsilon_t < 0.5$ **then**

    $\alpha_t = \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$ // classifier weight

    foreach $i \in [1, n]$ **do**

      // update point weights

      $$w_i^t = \begin{cases} w_i^{t-1} & \text{if } M_t(\mathbf{x}_i) = y_i \\ w_i^{t-1}\left(\frac{1-\epsilon_t}{\epsilon_t}\right) & \text{if } M_t(\mathbf{x}_i) \neq y_i \end{cases}$$

    $\mathbf{w}^t = \frac{\mathbf{w}^t}{\mathbf{1}^T\mathbf{w}^t}$// normalize weights

    $t \leftarrow t + 1$

return $\{M_1, M_2, \cdots, M_K\}$

Κατά τη διάρκεια της επανάληψης t το δείγμα προς εκπαίδευση D_t λαμβάνεται μέσω σταθμισμένης επαναδειγματοληψίας χρησιμόποιώντας την διανομή w^{t-1},δηλαδή έχουμε ένα δείγμα μεγέθους n με αντικατάσταση τέτοιο ώστε το i-ιοστό σημείο να επιλέγεται ανάλογα με την πιθανότητα w_i^{t-1} . Στη συνέχεια εκπαιδεύεται ο κατηγοριοποιητής M_t χρησιμοποιώντας το D_t και υπολογίζουμε το σταθμισμένο ποσοστό σφάλματος ε_t σε ολόκληρο το σετ δεδομένων εισόδου D που είναι: $\varepsilon_t = \sum_{i=1}^{n} w_i^{t-1} I(M_t(x_i) \neq y_i)$, όπου I είναι μια συνάρτηση ένδειξης η οποία είναι 1 όταν το όρισμά της είναι true,δηλαδή όταν ο M_t κατηγοριοποιεί λάθως το x_i.Το βάρος για τον t-ιοστό κατηγοριοποιητή τίθεται ως $\alpha_t = \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$ και το βάρος για κάθε σημείο $x_i \in D$ ενημερώνεται με βάση ποιο σημείο έχει κατηγοριοποιηθεί λάθως ή όχι ως εξής: $w_i^t = w_i^{t-1} \exp\{\alpha_t\} = w_i^{t-1} \exp\left\{\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)\right\} = w_i^{t-1}\left(\frac{1}{\varepsilon_t} - 1\right)$.Παρατηρούμε ότι αν το ποσοστό σφάλματος ε_t είναι μικρό τότε υπάρχει μεγαλύτερη αύξηση για το βάρος του x_i.

Διαισθητικά όποια σημεία έχουν κατηγοριοποιηθεί λάθως από επιτυχημένο κατηγοριοποιητή,που έχει δηλαδή μικρό ποσοστό σφάλματος είναι πιο πιθανό να επιλεγούν για το επόμενο εκπαιδευόμενο σετ δεδομένων.Από την άλλη εάν το ποσοστό σφάλματος του βασικού κατηγοριοποιητή είναι κοντά στο 0.5 τότε υπάρχει μικρή αλλαγή στο βάρος,αφου ένας ανεπιτυχής κατηγοριοποιητής αναμένεται να κατηγοριοποιήσει λάθως οποιοδήποτε στιγμιότυπο.Για ένα δυαδικό πρόβλημα κατηγοριοποίησης ένα ποσοστό σφάλματος 0.5 αντιστοιχεί σε τυχαίο κατηγοριοποιητή,που κάνει δηλαδή τυχαία εικασία.Έτσι απαιτείται ο βασικός κατηγοριοποιητής να έχει ποσοστό σφάλματος τουλάχιστον λίγο καλύτερο από την τυχαία εικασία δηλαδή $\varepsilon_t < 0.5$.Αν το ποσοστό σφάλματος είναι $\varepsilon_t \geq 0.5$ τότε η μέθοδος του boosting παραλείπει το σημείο και η ενημέρωση του βάρους του κατηγοριοποιητή προχωρά και επιστρέφει για να δοκιμάσει άλλο δείγμα.Αξίζει να δώσουμε έμφαση στο ότι για ένα πρόβλημα με πολλές κλάσεις,δηλαδή k>2,η απαίτηση ότι $\varepsilon_t < 0.5$ είναι πολύ πιο ισχυρή από ότι αν έχουμε δύο κλάσεις(k=2) αφού στην περίπτωση πολλών κλάσεων ένας τυχαίος κατηγοριοποιητής αναμένεται να έχει ποσοστό σφάλματος $\frac{1}{k}$.Αν το ποσοστό σφάλματος του βασικού κατηγοριοποιητή ε_t=0 τότε σταματάμε τις επαναλήψεις.Όταν τα βάρη των σημείων

ενημερωθούν ξανακανονικοποιούμε τα βάρη έτσι ώστε το w^t το διάνυσμα πιθανότητας να είναι $w^t = \frac{w^t}{1^T w^t} = \frac{1}{\sum_{j=1}^{n} w_j^t}(w_1^t, w_2^t, \dots, w_n^t)^T$.

Για τον συνδυαστικό κατηγοριοποιητή με δεδομένο το σετ των ενισχυμένων(boosted) κατηγοριοποιητών M_1, M_2, ... , M_K με τα βάρη τους α_1, α_2, ... , α_K η κλάση ενός σημείου x λαμβάνεται από την πλειοψηφία της σταθμισμένης ψηφοφορίας. Αν $\upsilon_j(x)$ συμβολίζεται η σταθμισμένη ψήφος για την κλάση c_j πάνω από Κ κατηγοριοποιητές τότε έχουμε τον τύπο: $\upsilon_j(x)= \sum_{t=1}^{K} \alpha_t I(M_t(x) = c_j)$. Αφού το $I(M_t(x) = c_j)$ είναι 1 μόνο όταν $M_t(x) = c_j$ η μεταβλητή $\upsilon_j(x)$ απλά λαμβάνει την αντιστοιχία για την κλάση c_j ανάμεσα στους Κ βασικούς κατηγοριοποιητές λαμβάνοντας υπ'όψιν τα βάρη τους.Ο συνδυαστικός κατηγοριοποιητής που συμβολίζεται με M^K προβλέπει την κλάση για το x ως $M^K(x)=argmax_{c_j}\{\upsilon_j(x)|j = 1, \dots, k\}$.Στην περίπτωση του δυαδικού κατηγοριοποιητή με κλάσεις {+1, -1} ο συνδυαστικός κατηγοριοποιητής M^K εκφράζεται πιο απλά ως $M^K(x)=\text{sign}(\sum_{t=1}^{K} \alpha_t M_t(x))$.

Το bagging μπορεί να θεωρηθεί ως μια ειδική περίπτωση του AdaBoost όπου $w^t=\frac{1}{n}1$ και $\alpha_t=1$ για όλες τις Κ επαναλήψεις.Σε αυτή την περίπτωση η σταθμισμένη επαναδειγματοληψία συμπίπτει με την κανονική επαναδειγματοληψία με αντικατάσταση και οι προβλεπόμενες κλάσεις για μια δοκιμαστική περίπτωση συμπίπτει επίσης με την απλή ψηφοφορία πλειοψηφίας.

Η δύναμη που παρέχει το boosting σήμερα είναι αδιαμφισβήτητη.Η ιδέα προέκυψε από έναν κλάδο της μηχανικής μάθησης γνωστό και ως θεωρία της μάθησης.Οι θεωριτικοί ενδιαφέρθηκαν πολύ για το boosting διότι είναι δυνατόν να αντληθούν εγγυήσεις στην καλή απόδοση.Για παράδειγμα αποδεικνύεται ότι το σφάλμα του συνδιαστικού κατηγοριοποιητή στα εκπαιδευόμενα δεδομένα πλησιάζει το 0 πολύ γρήγορα όσο εκτελούνται περισσότερες επαναλήψεις(εκθετικά με τον αριθμό των επαναλήψεων).Δυστυχώς οι εγγυήσεις για το σφάλμα εκπαίδευσης δεν είναι τόσο ενδιαφέρουσες γιατί δεν υποδεικνύουν απαραίτητακαλή επίδοση σε νέα δεδομένα.Παρόλα αυτά μπορεί να δειχθεί θεωρητικά ότι το boosting αποτυγχάνει μόνο σε νέα δεδομένα εάν οι μεμονωμένοι κατηγοριοποιητές είναι τόσο πολύπλοκοι για τον όγκο των δεδομένων λη εάν τα σφάλματα γίνουν πολύ μεγάλα πολύ γρήγορα.Ως συνήθως το πρόβλημα βρίσκεται στην εύρεση της σωστής ισορροπίας μεταξύ της πολυπλοκότητας των μεμονωμένων μοντέλων και την προσαρμογή τους στα δεδομένα.

Αν το boosting επιτύχει τη μείωση του σφάλματος σε νέα δοκιμαστικά δεδομένα αυτό γίνεται με θεαματικό τρόπο.Μια εκπληκτική ανακάλυψη είναι ότι εφαρμόζοντας περισσότερες επαναλήψεις στο boosting μπορεί να μειωθεί το σφάλμα σε νέα δεδομένα πολύ αργότερα αφότου το σφάλμα του συνδυαστικού κατηγοριοποιητή πέσει στο 0.Οι ερευνητές αναρωτήθηκαν πολύ για αυτό το αποτέλεσμα γιατί φαίνεται να αντιφάσκει με την αρχή του ξυραφιού του Όκαμ (Occam's razor) ,η οποία δηλώνει ότι μεταξύ δύο υποθέσεων που εξηγούν κάποια εμπειρικά στοιχεία το ίδιο καλά να προτιμάται η πιο απλή από τις δύο.Το να εφαρμόζονται περισσότερες επαναλήψεις ενίσχυσης χωρίς να μειώνεται το σφάλμα εκπαίδευσης δεν εξηγεί τα δεδομένα καλύτερα και φυσικά προσθέτει πολυπλοκότητα στον συνδυαστικό κατηγοριοποιητή.Η αντίφαση μπορεί να επιλυθεί με το να ληφθεί υπ'όψιν η σιγουριά του κατηγοριοποιητή στις προβλέψεις.Πιο συγκεκριμένα μετράμε τη σιγουριά με τη διαφορά μεταξύ της αναμενόμενης σιγουριάς για τη σωστή κλάση και αυτή της πιο πιθανής να προβλεφθεί κλάσης εκτός από την πραγματική κλάση,μια ποσότητα γνωστή και ως περιθώριο.Όσο πιο ευρύ περιθώριο τόσο πιο σίγουρος είναι ο κατηγοριοποιητής στην πρόβλεψη της σωστής κλάσης.

Προκύπτει επίσης ότι το boosting μπορεί να αυξήσει το περιθώριο πολύ αργότερα αφότου το σφάλμα εκπαίδευσης πέσει στο 0.Το αποτέλεσμα αυτού μπορεί να γίνει ορατό παρουσιάζοντας γραφικά την συσσωρευμένη κατανομή των τιμών του περιθωρίου όλων των εκπαιδευόμενων στιγμιοτύπων για διάφορες τιμές των επαναλήψεων παίρνοντας ένα γράφημα γνωστό και ως την καμπύλη περιθωρίου.Έτσι εάν η εξήγηση των εμπειρικών στοιχείων λάβει υπ'όψιν και τα περιθώρια τότε ισχύει η αρχή του ξυραφιού του Όκαμ.Ένα

άλλο στοιχείο για το boosting είναι ότι ένας ισχυρός συνδυαστικός κατηγοριοποιητής μπορεί να κατασκευαστεί από πολύ απλούς κατηγοριοποιητές εφόσον αυτοί πετυχαίνουν λιγότερο από 50% σφάλμα στα ξανασταθμισμένα δεδομένα.Αυτά τα απλά σχήματα εκμάθησης ονομάζονται ασθενείς μαθητευόμενοι και με το boosting μπορούν να μετατραπούν σε ισχυρούς μαθητευόμενους.Για παράδειγμα καλά δεδομένα για προβλήματα δύο κλάσεων μπορούν να ληφθούν με το να ενισχύσουμε εξαιρετικά απλά δέντρα αποφάσεων που έχουν μόνο ένα επίπεδο και αποκαλούνται κορμοί απόφασης.

Μια ακόμη καλή λύση είναι να εφαρμόσουμε το boosting σε έναν αλγόριθμο που μαθαίνει απλά ένα μοναδικό συνδετικό κανόνα όπως ένα μοναδικό μονοπάτι σε ένα δέντρο αποφάσεων οπότε τα στιγμιότυπα κατηγοριοποιούνται με βάση αν ο κανόνας τα καλύπτει ή όχι.Φυσικά τα σετ δεδομένων με πολλές κλάσεις είναι πιο δύσκολο να επιτύχουν ποσοστά σφάλματος κάτω από 0.5.Τα δέντρα αποφάσεων μπορούν ακόμα να ενισχυθούν αλλά συνήθως χρειάζεται να γίνουν πιο πολύπλοκα από κορμούς αποφάσεων.Πιο εξελιγμένοι αλγόριθμοι έχουν αναπτυχθεί που επιτρέπουν πολύ απλά μοντέλα να ενισχύονται επιτυχώς σε καταστάσεις με πολλές κλάσεις.Το boosting συχνά παράγει κατηγοριοποιητές που είναι εμφανώς πιο ακριβείς σε νέα δεδομένα από ότι αυτά που παρήχθησαν με το bagging.Όμως,αντίθετα με το bagging,το boosting κάποιες φορέςαποτυγχάνει σε πραγματικές καταστάσεις ,γιατί μπορεί να παράγει έναν κατηγοριοποιητή που είναι εμφανώς πιο ανακριβής από ότι ένας μοναδικός κατηγοριοποιητής που κατασκευάστηκε από τα ίδια δεδομένα.Αυτό το γεγονός υποδεικνύει ότι ο συνδυαστικός κατηγοριοποιητής υπερπροσαρμόζεται στα δεδομένα.

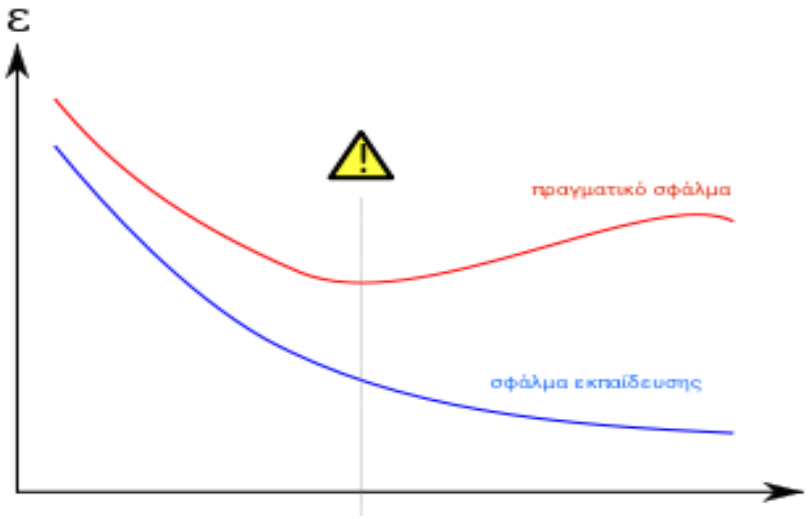
2.5 Το πρόβλημα της υπεπροσαρμογής στα δεδομένα(Overfitting)

Στη μηχανική μάθηση η γενίκευση είναι απολύτως μία από τις πιο σημαντικές λέξεις. Το πρόβλημα της υπερπροσαρμογής στα δεδομένα προκύπτει όταν ένα μοντέλο αντί για την υποκείμενη σχέση περιγράφει τυχαίο σφάλμα ή και θόρυβο. Ένας πλήρης ορισμός για την υπερπροσαρμογή είναι ο εξής: Με ένα χώρο υποθέσεων H, μια υπόθεση $h \in H$ λέμε ότι υπερπροσαρμόζεται στα εκπαιδευόμενα δεδομένα εάν υπάρχει μια εναλλακτική υπόθεση $h' \in H$ έτσι ώστε η h να έχει μικρότερο σφάλμα από την h' στα παραδείγματα εκπαίδευσης αλλά η h' έχει μικρότερο σφάλμα από την h σε ολόκληρη τη διανομή των στιγμιοτύπων.

Η υπερπροσαρμογή γενικά οφείλεται σε διάφορους παράγοντες.Ένας από αυτούς είναι ότι το μοντέλο μπορεί να είναι υπερβολικά πολύπλοκο,δηλαδή να έχει πάρα πολλές παραμέτρους.Επίσης τα δεδομένα για την εκπαίδευση μπορεί να εμπεριέχουν θόρυβο ή το μοντέλο μπορεί να βελτιώνεται με την πάροδο του χρόνου αλλά οι εισροή δεδομένων ολοένα να αυξάνεται. Άλλοι λόγοι μπορεί να είναι ότι το σετ δεδομένων για την εκπαίδευση ίσως είναι πολύ μικρό ή ο χώρος υποθέσεων είναι πολύ “πλούσιος” και με πολλούς βαθμούς ελευθερίας.Φαινόμενα της υπερπροσαρμογής είναι όταν για μικρές διακυμάνσεις στα δεδομένα παρουσιάζεται το μοντέλο να υπερβάλλει ή σε ένα πολύπλοκο μοντέλο να γίνεται ταίριασμα όχι μόνο με το πραγματικό σήμα αλλά και με το θόρυβο που περιέχεται.Ακόμη μπορεί τα δεδομένα να περιγράφονται καλά αλλά οι προβλέψεις να μην μπορούν να γενικευτούν σε καινούργια δεδομένα εκτός του δείγματος μελέτης.Τέλος μπορεί η απόδοση του εκπαιδευόμενου σετ να συνεχίζει να βελτιώνεται ενώ αντίθετα η απόδοση του δοκιμαστικού σετ να μην βελτιώνεται περισσότερο.

Αποτέλεσμα της υπερπροσαρμογής είναι ότι ένα μοντέλο θα έχει γενικά μικρή απόδοση στις προβλέψεις του καθώς γίνονται μικρές διακυμάνσεις στα δεδομένα.Η πιθανότητα υπερπροσαρμογής υπάρχει διότι το κριτήριο που χρησιμοποιήθηκε για την εκπαίδευση του μοντέλου δεν είναι το ίδιο με το κριτήριο που χρησιμοποιήθηκε για να κριθεί η αποτελεσματικότητα του μοντέλου.Συγκεκριμένα ένα μοντέλο τυπικά εκπαιδεύεται με το να μεγιστοποιείται η απόδοσή του σε ένα σύνολο δεδομένων.Όμως η αποτελεσματικότητά του φαίνεται όχι από την απόδοσή του στα δεδομένα που έχει δει

αλλά από την ικανότητά του να αποδίδει καλά σε άγνωστα δεδομένα.Έτσι εάν το μοντέλο αρχίσει να απομνημονεύει τα δεδομένα εκπαίδευσης αντί να μαθαίνει και να αποκτά την τάση να γενικεύει προκύπτει η υπερπροσαρμογή.Παρακάτω φαίνεται και σε γράφημα ένα παράδειγμα υπερπροσαρμογής.



Εικόνα 2.9. Υπερπροσαρμογή-υπερεκπαίδευση όπου το πραγματικό σφάλμα αυξάνεται και το σφάλμα εκπαίδευσης μειώνεται σταθερά

Ένα ακραίο παράδειγμα είναι η περίπτωση που ο αριθμός των παραμέτρων είναι ίσος ή μεγαλύτερος από τον αριθμό των παρατηρήσεων οπότε ένα απλό μοντέλο θα μπορούσε να μαντέψει τέλεια τα δεδομένα εκπαίδευσης με απομνημόνευση αλλά το μοντέλο αυτό θα αποτύγχανε παταγωδώς στο να κάνει προβλέψεις για καινούργια δεδομένα,διότι όπως πριν αναφέρθηκε δεν έχει μάθει να γενικεύει.Η πιθανότητα υπερπροσαρμογής δεν εξαρτάται μόνο από τον αριθμό των παραμέτρων και των δεδομένων αλλά επίσης και από την προσαρμοστικότητα του μοντέλου,τη δομή και το σχήμα των δεδομένων καθώς και μέγεθος του σφάλματος σε σύγκριση με το αναμενόμενο επίπεδο του θορύβου ή του σφάλματος στα δεδομένα.

Για να αποφευχθεί η υπερπροσαρμογή είναι απραίτητο να εφαρμοστούν επιπλέον τεχνικές που μπορούν να υποδείξουν ότι η περιταίρω εκπαίδευση δεν καταλήγει σε καλύτερη γενίξευση.Κάποιες από αυτές είναι η διασταύρωση-επαλήθευση,η κανονικοποίηση,το πρόωρο σταμάτημα,το κλάδεμα,η τεχνική Bayesian priors σε παραμέτρους ή η συγκριση μοντέλων.Κάποιες από αυτές τις τεχνικές βασίζονται είτε στην επιβολή ποινής σε υπερβολικά πολύπλοκα μοντέλα είτε στη δοκιμή της ικανότητας του μοντέλου να γενικεύει με αξιολόγηση της επίδοσής του σε σετ δεδομένων που δε χρησιμοποιούνται στην εκπαίδευση,τα οποία υποτίθεται ότι προσεγγίζουν τα άγνωστα δεδομένα που προκύπτουν.

Το 1987 ο George Box είπε ότι “όλα τα μοντέλα είναι λανθασμένα,αλλά κάποια μοντέλα είναι χρήσιμα” αναφερόμενος στην επιλογή μοντέλου που θα ταίριαζε καλύτερα στο εκάστοτε πρόβλημα.Μεγάλο μέρος της μηχανικής μάθησης ασχολείται με την επινόηση διαφορετικών μοντέλων και αλγορίθμων που ταιριάζουν με αυτά.Μπορούμε να χρησιμοποιήσουμε μεθόδους όπως η η διασταύρωση-επαλήθευση ώστε εμπειρικά να επιλέξουμε το καλύτερο μοντέλο για το πρόβλημά μας.Παρόλο αυτά όμως δεν υπάρχει καθολικά καλύτερο μοντέλο κι αυτή είναι η ιδέα του No free lunch(“δεν-παρέχεται-δωρεάν-γεύμα”) θεωρήματος.ουσιαστικά το θεώρημα λέει ότι ένα σύνολο από υποθέσεις που δουλεύει καλά σε ένα πεδίο ορισμού μπορεί να μη δουλεύει σχεδόν καθόλου σε ένα άλλο.Ως συνέπεια αυτού θα πρέπει να αναπτύξουμε πολλούς και διαφορετικούς τύπους μοντέλων για να καλύψουμε μεγάλο εύρος δεδομένων που προκύπτουν στον πραγματικό κόσμο.Επιπλεόν για το κάθε μοντέλο είναι πιθανόν να υπάρχουν πολλοί διαφορετικοί

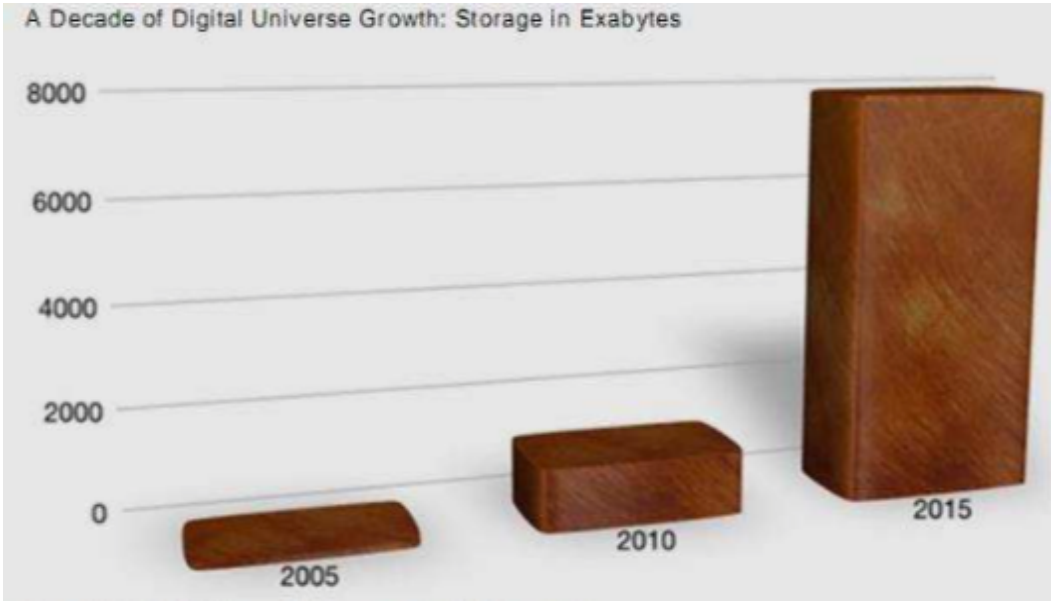
αλγόριθμοι που μπορούμε να χρησιμοποιήσουμε για να εκπαιδεύσουμε το μοντέλο με διαφορετικές αναλογίες ταχύτητας,ακρίβειας,πολυπλοκότητας.

ΚΕΦΑΛΑΙΟ 3

Big Data – "Πολλά" Δεδομένα

3.1 Κατακλυσμός δεδομένων

Πρόσφατα άρθρα αναφέρουν ότι μέχρι στιγμής έχουν παραχθεί 2,5 πεντάκις εκατομμύρια δεδομένων,εκ των οποίων το 90% δημιουργήθηκε μόνο τα τελευταία 2 χρόνια.Η έκρηξη που έχει γίνει στην πληροφορία μας αναγκάζει να χρησιμοποιήσουμε καινούργιες μονάδες μέτρησης με τις οποίες θα πρέπει να εξοικειωθούμε για να μπορέσουμε να μετρήσουμε τα "Πολλά" Δεδομένα.Όλα αυτά τα δεδομένα θα ήταν άχρηστα αν δε μπορούσαμε να τα αποθηκεύσουμε κάπου.Σε αυτό όμως το σημείο επεμβαίνει ο νόμος του Moore που αναφέρεις ότι ο αριθμός των τρανζίστορ στα ολοκληρωμένα κυκλώματα διπλασιάζεται κάθε 2 χρόνια σε αντίθεση με την ταχύτητα των επεξεργαστών που από το 1980 ως σήμερα αυξήθηκε από 10MHz σε 3.6GHz, δηλαδή μόνο 360 φορές περισσότερο.



Εικόνα 3.1. Αύξηση της αποθήκευσης παγκοσμίως 2005-2015

Επιπλέον υπήρξε μεγάλη αύξηση σε αποθηκευτική ικανότητα σε όλα τα επίπεδα μνήμης.Η RAM από 1000$/MB κοστίζει λιγότερο από 25$/GB μια μείωση τιμής ,δηλαδή, 40.000 φορές πιο κάτω ενώ ακολούθησε και μείωση στο χώρο που καταλαμβάνει ως υλικό και αύξηση στην ταχύτητα προσπέλασης.Σε όποιο λοιπόν μέτρο και να στραφούμε είτε bits/γραμμάριο, είτε bits/δολλάριο η δυνατότητα αποθήκευσης έχει αυξηθεί κατά πολύ περισσότερο της ταχύτητας του επεξεργαστή. Από την άλλη όχι μόνο οι διαστάσεις των

δεδομένων μεγαλώνουν αλλά η φύση τους αλλάζει,κυρίως με την ευρεία χρήση των μέσων κοινωνικής δικτύωσης και των υπερεσιών που παρέχονται σε κινητά τηλέφωνα.

3.2 Ορισμοί για τα "Πολλά" Δεδομένα

Γνωρίζουμε ότι το να δώσουμε έναν ορισμό για τα Big Data δεν είναι εύκολο,αφού για μια υπηρεσία που καλείται (Big)Data-as-a-Service μπορούν να προκύψουν πάνω από 30 ορισμοί.Έτσι θα γίνει μια προσπάθεια καταγραφής των πιο συνηθισμένων ορισμών που έχουν δωθεί κατά καιρούς.Ένας από αυτούς λέει ότι τα Big Data είναι μια συλλογή από σετ δεδομένων τόσο μεγάλα σε μέγεθος και τόσο πολύπλοκα που είναι αδύνατο να τα επεξεργαστούμε χρησιμοποιώντας εργαλεία διαχείρισης βάσεων δεδομένων ή παραδοσιακές εφαρμογές επεξεργασίας δεδομένων.Ένας άλλος ορισμός στον οποίο εμμένουν πολλοί συγγραφείς είναι αυτός των "4-V" που δείχνει τα 4 χαρακτηριστικά που αφορούν τα Big Data και είναι ο όγκος(volume), η ποικιλία(variety), η ταχύτητα(velocity) και η ακρίβεια(verasity).

- Όγκος (η ποσότητα των δεδομένων): αναφέρεται στη μαζική ποσότητα των δεδομένων όπου οι εταιρείες και οι οργανισμοί χρησιμοποιούν για να βελτιώσουν τη διαδιακασία λήψης αποφάσεων.Η ποσότητα των δεδομένων συνεχίζει να αυξάνεται συνεχώς και με πρωτοφανείς ρυθμούς.Ωστόσο αυτό που συνιστά πραγματικά υψηλή ποσότητα δεδομένων διαφέρει σε κάθε βιομηχανία και σε κάθε γεωγραφική περιοχή αλλά σχεδόν πάντα είναι μικρότερη από petabytes και zettabytes.Πολλές εταιρείες μάλιστα θεωρούν σύνολα μεταξύ ενός terabyte και ενός petabyte να ανήκουν στην κατηγορία των Big Data.Μπορούμε να συμφωνήσουμε στο γεγονός ότι αυτό που θεωρείται μεγάλη ποσότητα δεδομένων σήμερα μπορεί να είναι ακόμη μεγαλύτερο στο άμεσο μέλλον.
- Ποικιλία (διαφορετικοί τύποι δεδομένων και πηγών που προέρχονται): η ποικιλία αναφέρεται περισσότερο στη διαχείριση της πολυπλοκότητας από τους πολλαπλούς τύπους δεδομένων συμπεριλαμβανομένων των δομημένων, ημι-δομημένων και μη δομημένων δεδομένων.Οι οργανισμοί θα πρέπει να ενσωματώσουν και να αναλύσουν δεδομένα από μια σύνθετη σειρά από παραδοσιακές και μη πηγές πληροφόρησης εντός και εκτός της επιχείρησης.Με την έκρηξη της τεχνολογίας των αισθητήρων,των έξυπνων συσκευών και των μέσων κοινωνικής δικτύωσης τα δεδομένα παράγονται σε αμέτρητες μορφές όπως κείμενο,διαδικτυακό υλικό,ήχος,εικόνα,tweets,αρχεία καταγραφής κ.ά.
- Ταχύτητα (δεδομένα σε κίνηση): η ταχύτητα με την οποία τα δεδομένα δημιουργούνται,επεξεργάζονται και αναλύονται αυξάνεται συνεχώς. Η υψηλότερη αυτή ταχύτητα οφείλεται τόσο στην πραματικού χρόνου φύση της δημιουργίας δεδομένων όσο και στην ανάγκη ενσωμάτωσης των δεδομένων ροής στις επιχειρηματικές διαδικασίες.Σήμερα τα δεδομένα παράγονται με τέτοιο ρυθμό που είναι αδύνατο για τα παραδοσιακά συστήματα να συλλέξουν να αποθηκεύσουν και να αναλύσουν.Για τις διαδικασίες που είναι ευαίσθητες ως προς το χρόνο όπως το άμεσο,στιγμιαίο και πολυκαναλικό μάρκετινγκ τα δεδομένα θα πρέπει να αναλυθούν σε πραγματικό χρόνο ώστε να έχουν αξία για τις επιχειρήσεις.
- Ακρίβεια (αβεβαιότητα των δεδομένων): αναφέρεται στο επίπεδο αξιοπιστίας που σχετίζεται με ορισμένους τύπους δεδομένων.Η αναζήτηση για υψηλή ποιότητα δεδομένων είναι σημαντική απαίτηση και πρόκληση για τον τομέα των Big Data,αλλά ακόμη και οι καλύτερες μέθοδοι καθαρισμού των δεδομένων δε μπορούν να αφαιρέσουν την εγγενή μη προβλεψιμότητα μερικών δεδομένων που σχετίζονται για παράδειγμα με τον καιρό,την οικονομία ή τις αποφάσεις αγοράς ενός πελάτη.Η ανάγκη για αναγνώριση κα σχεδιασμό της αβεβαιότητας είναι μια διάσταση που εισήχθη ως

μια προσπάθεια να προσπαθήσουμε να κατανοήσουμε καλύτερα τον αβέβαιο κόσμο γύρω από τα Big Data.

Ένας τρίτος ορισμός αναφέρει ότι “Big Data είναι ένας όρος που περιγράφει μεγάλους όγκους δεδομένων υψηλής ταχύτητας,με πολύπλοκα και μεταβλητά δεδομένα που απαιτούν προηγμένες τεχνικές και τεχνολογίες για τη συγκέντρωση,την αποθήκευση,τη διανομή,τη διαχείριση και την ανάλυση των πληροφοριών”.Ένας ορισμός που δόθηκε από τον Tim O’Reilly αναφέρει ότι “Big Data ονομάστηκε το φαινόμενο που συνέβη όταν το κόστος αποθήκευσης των δεδομένων έγινε μικρότερο από το κόστος της λήψης απόφασης στην περίπτωση που τα δεδομένα πεταχτούν ή καταστραφούν”. Άλλοι ορισμοί εστιάζουν περισσότερο στην προέλευση των Big Data, η οποία ταξινομείται ως εξής:

1. Πληροφορία ανθρώπινης προέλευσης: ουσιαστικά αποτελείται από τα δεδομένα που παράγονται αποκλειστικά από ανθρώπους.Περιλαμβάνει την υποκειμενική καταγραφή των ανθρώπινων εμπειριών που προηγουμένως έχουν καταγραφεί σε βιβλία και αργότερα σε φωτογραφίες, ήχο και βίντεο. Σήμερα όλες αυτές οι πληρφορίες σχεδόν εξ’ολοκλήρου ψηφιοποιούνται και αποθηκεύονται με ηλεκτρονικά μέσα. Αυτή η τυποποιημένη και δομημένη μοντελοποίηση ορίζει μια κοινή εκδοχή που επιτρέπει στην επιχείρηση να μετατρέπει τα δεδομένα που προέρχονται από ανθρώπους σε πιο αξόπιστα δεδομένα.Η διαδικασία ξεκινά με την εισαγωγή και την επικύρωση των δεδομένων και συνεχίζει με τον καθαρισμό και την εξαγωγή συμπερασμάτων από τα δεδομένα (Business Intelligence). Οι πληροφορίες που αντλούνται από αυτή την πηγή συνήθως είναι χαλαρά δομημένες και προέρχονται συνήθως από κοινωνικά δίκτυα(Facebook, Twitter, Tumblr),ιστολόγια και σχόλια σε αυτά,προσωπικά έγγραφα,εικόνες και φωτογραφίες που τράβηξε και ανέβασε ο χρήστης(Instagram, Picassa, Flickr),βίντεο(Youtube), αναζητήσεις σε μηχανές,περιεχόμενο από κινητά τηλέφωνα(γραπτά μηνύματα),χάρτες που δημιουργούνται από χρήστες,ηλεκτρονικό ταχυδρομείο.
2. Παραδοσιακά Επιχειρηματικά Συστήματα: αυτά περιλαμβάνουν καταγραφή και παρακολούθηση επιχειρηματικών διαδικασιών,εκδηλώσεις ενδιαφέροντος όπως εγγραφή ενός πελάτη,κατασκευή ενός προϊόντος,λήψη μιας παραγγελίας.Τα δεδομένα που συλλέγονται είναι εξαιρετικά δομημένα και περιλαμβάνουν συναλλαγές,πίνακες, σχέσεις αναφοράς καθώς και μεταδεδομένα ενώ συνήθως αποθηκεύονται σε σχεσιακές βάσεις δεδομένων.Περιλαμβάνουν δεδομένα που παράγονται από δημόσιους οργανισμούς,όπως ιατρικά αρχεία,αλλά και από επιχειρήσεις,όπως εμπορικές συναλλαγές,τραπεζικά αρχεία,πιστωτικές κάρτες.
3. Πληροφορία παραγόμενη από μηχανές: αναφέρεται στην έξοδο μηχανών και αισθητήρων που χρησιμοποιούνται για τη μέτρηση και καταγραφή γεγονότων και καταστάσεων στον πραγματικό κόσμο, ενώ πολλές φορές αναφέρεται και ως διαδίκτυο των πραγμάτων(Internet of Things).Έτσι από απλά αρχεία καταγραφής παράγονται πιο σύνθετα τα οποία είναι καλά δομημένα και εξαιρετικά αξιόπιστα.Με την αύξηση των αισθητήρων και τον πολλαπλασιασμό του όγκου των δεδομένων γίνεται όλο και πιο σημαντική η αποθήκευση και επεξεργασία από πολλούς υπολογιστές και με μεγάλη ταχύτητα.Η καλά δομημλενη φύση της πληροφορίας βοηθά,όπως επίσης και η προσέγγιση των δεδομένων με χρήση μη σχεσιακών βάσεων δεδομένων(NoSQL) ώστε να επιτευχθεί η επιθυμητή απόδοση.Σε αυτή την κατηγορία περιλαμβάνονται εφαρμογές από αισθητήρες όπως οικιακοί αυτοματισμοί,ασφάλεια και παρακολούθηση, αισθητήρες καταγραφής καιρού, ρύπανσης, κίνησης,αναγνώρισης τοποθεσίας από στίγματα κινητών τηλεφώνων, δορυφορικές εικόνες,αισθητήρες αυτοκινήτων.

3.3 Τεχνολογίες Big Data

Μιλώντας για τεχνολογίες που επιτρέπουν τη χρήση των Big Data υπάρχουν 3 ουσιαστικές τεχνολογικές στρατηγικές για την αποθήκευση και παροχή γρήγορης πρόσβασης σε μεγάλα σύνολα δεδομένων:

1. Βελτιώνεται η απόδοση και η ικανότητα του υλικού με τη χρήση γρήγορων επεξεργαστών,χρήση περισσότερων πυρήνων,όπου απαιτείται παράλληλοποίηση διαδικασιών με νήματα ώστε να επωφεληθούμε από τους πολυπύρηνους επεξεργαστές.Επιπλέον αυξάνεται η χωρητικότητα του δίσκου και η απόδοση μεταφοράς δεδομένων και η απόδοση του δικτύου(throughput).
2. Μειώνεται το μέγεθος των δεδομένων στα οποία γίνεται προσπέλαση μέσω της συμπίεσης και της δόμησης των δεδομένων περιορίζοντας την ποσότητα των δεδομένων που χρειάζονται για ερωτήματα.
3. Γίνεται κατανομή των δεδομένων και παράλληλη επεξεργασία αυτών.Βάζοντας τα δεδομένα σε περισσότερους δίσκους επιτυγχάνεται παραλληλοποίηση εισόδων/εξόδων (I/O), καθώς τοποθετούνται κομμάτια δεδομένων σε ξεχωριστούς υπολογιστικούς κόμβους κι έτσι χρησιμοποιείται κατανεμημένη αρχιτεκτονική και δίνεται έμφαση στην ανοχή σφαλμάτων,την παρακολούθηση της απόδοσης για τη βελτίωση μεταφοράς δεδομένων μεταξύ των κόμβων.

3.3.1 Μαζική Παράλληλη Επεξεργασία (MPP)

Η αρχιτεκτονική μαζικής παράλληλης επεξεργασίας των σχεσιακών βάσεων δεδομένων χωρίζει τα δεδομένα σε ένα μεγάλο αριθμό ανεξάρτητων εξυπηρετητών κόμβων κατά τρόπο διαφανή για όσους χρησιμοποιούν τη βάση.Σε περιβάλλοντα Big Data χρησιμοποιούνται συστήματα παράλληλης επεξεργασίας που συνήθως αποκαλούνται “καθόλου διαμοιραζόμενα” καθώς οι κόμβοι που απαρτίζουν μια συστάδα λειτουργούν ανεξάρτητα και επικοινωνούν μέσω ενός δικτύου αλλά δεν μοιράζονται ούτε το δίσκο ούτε πόρους της μνήμης.Με τους σύγχρονους πολυπύρηνους επεξεργαστές οι βάσεις δεδομένων παράλληλης επεξεργασίας μπορούν να ρυθμιστούν έτσι ώστε να αντιμετωπίζεται κάθε πυρήνας σαν ένας κόμβος και να εκτελεί κάποιες εργασίες παράλληλα σε ένα μόνο επεξεργαστή.

Με τη διανομή των δεδομένων στους κόμβους και την εκτέλεση λειτουργιών της βάσης σε όλους τους κόμβους παράλληλα οι MPP βάσεις παρέχουν γρήγορη απόδοση ακόμα και όταν χειρίζονται πολλά δεδομένα.Η αρχιτεκτονική μαζικής παραλληλίας ή ο “μηδενικός διαμοιρασμός” επιτρέπει σε τέτοιου είδους βάσεις να κλιμακώσουν την απόδοσή τους καθώς καινούργιοι κόμβοι προστίθονται, οπότε για παράδειγμα μια συστάδα με 8 κόμβους μπορεί να εκτελείται δύο φορές πιο γρήγορα από ότι μια συστάδα με 4 κόμβους για ίδια δεδομένα.Η ομάδα των εξυπηρετητών που συνθέτουν ένα σύστημα MPP ονομάζεται συστάδα ή σύμπλεγμα.

Ένα βασικό σημείο για την απόδοση των MPP είναι η ομοιόμορφη διανομή των δεδομένων σε όλους τους κόμβους του συμπλέγματος.Αυτό απαιτεί τον προσδορισμό ενός κλειδιού του οποίου η τιμή είναι αρκετά τυχαία έτσι ώστε ακόμα και με την πάροδο του χρόνου τα δεδομένα δεν συγκεντρώνονται σε έναν ή ένα υποσύνολο κόμβων.Οι βάσεις δεδομένων MPP διαθέτουν αλγόριθμους που βοηθούν στο να κρτήσουν τα δεδομένα διανεμημενα,αφού σε αντίθετη περίπτωση που χρησιμοποιηθούν λανθασμένα πεδία για τη διανομή μπορούν να οδηγήσουν σε ασύμμετρη κατανομή και κατά συνέπεια πολύ κακή απόδοση.Οι MPP βάσεις έχουν το πλεονέκτημα ότι επεκτείνονται πολύ εύκολα με την προσθήκη επιπλέον υλικού και με τη χρησιμοποίηση της γλώσσας SQL έτσι ώστε να μπορούν εύκολα να ενσωματωθούν με εργαλεία προβολής χωρίς να απαιτούνται νέες δεξιότητες.

3.3.2 Μη σχεσιακές βάσεις δεδομένων προσανατολισμένες σε στήλες

Τα παραδοσιακά Συστήμα Διαχείρισης Σχεσιακών Βάσεων Δεδομένων(RDBMS) εισήγαγαν το πρότυπο πρόσβασης στα δεδομένα χρησιμοποιώντας τη γλώσσα SQL και αναπτύχθηκαν σε εποχές που στη δομημενη πληροφορία μπορούσε να δοθεί πρόσβαση,να κατηγοριοποιηθεί και να κανονικοποιηθεί με σχετική ευκολία.Τα RDBMS σχεδιάστηκαν για να καλύψουν ένα ευρύ φάσμα διαφορετικών τύπων ερωτημάτων κοιτάζοντας να ενσωματώσουν τα δεδομένα που υποβάλλονταν σε επεξεργασία από το λογισμικό με εξαιρετικά δομημένο τρόπο.

Η πρώτη προσπάθεια για να ξεπεραστεί η λογική του RDBMS ήταν οι βάσεις δεδομένων προσανατολισμένη σε στήλες με την οποία τα δεδομένα αποθηκεύονται κατά στήλες, η οποία μετατράπηκε όταν αυτό ήταν δυνατό σε bitmaps ή σε συμπίεση με άλλους τρόπους έτσι ώστε να μειωθεί ο όγκος των αποθηκευμένων δεδομένων.Ο συνδιασμός των συμπιεσμένων δεδομένων και η ανάκτηση μόνο όσων στηλών ζητώνται επιταχύνει την ταχύτητα με την οποία εκτελείται το ερώτημα με το να μειώνεται ο αριθμός των προσβάσεων I/O που χρειάζονται και με το να αυξάνεται ο αριθμός των ερωτηθέντων δεδομένων που μπορεί να παραμείνει στη μνήμη.Τα πρώτα παραδείγματα τέτοιων βάσεων διατήρησαν τη χρήση της SQL αλλά τελευταία καθιερώθηκε το μοντέλο NoSQL που προχωρά το σχεσιακό μοντέλο.Με τη χρηση αυτού χτίζονται προϊόντα που στόχο έχουν να διαχειριστούν πολύ μεγάλα σετ.Σε NoSQL βάσεις το σχήμα(schema)δεν είναι γνωστό από την αρχή που σημαίνει ότι υπάρχει δυνατότητα να προσαρμοστεί στην πραγματική μορφή των δεδομένων.

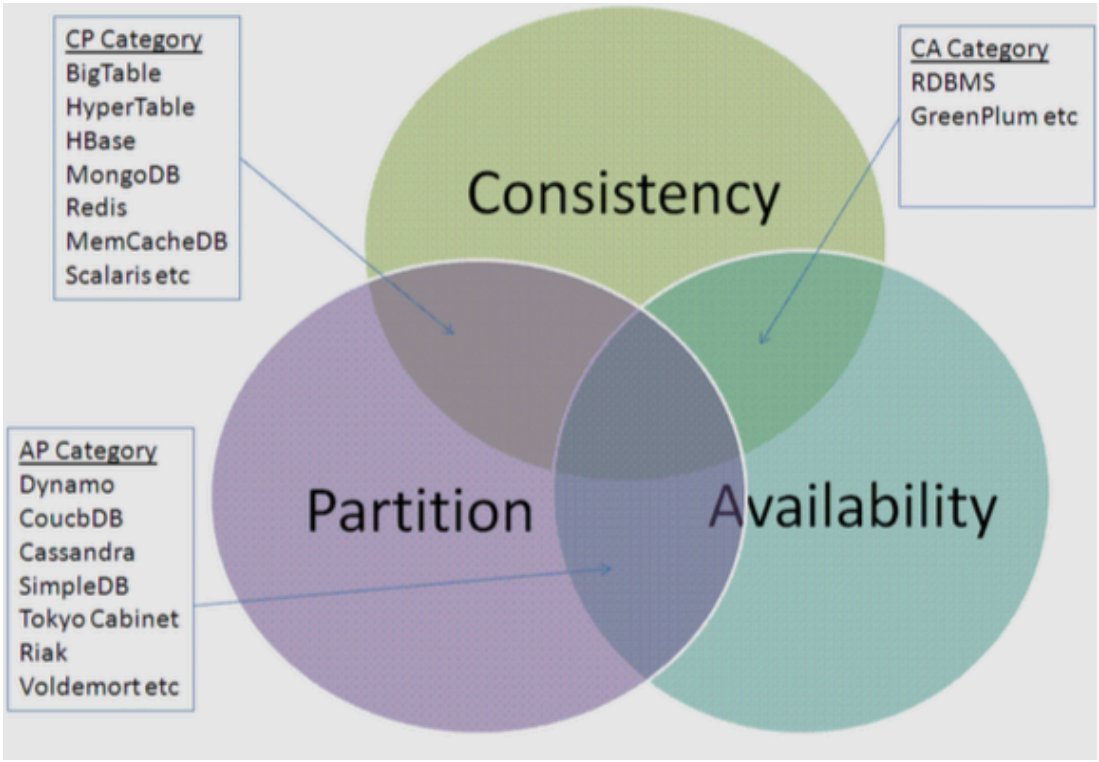
Τέτοια ευελιξία είναι το κλειδί για μεγαλύτερη επέκταση μιας και η φυσική αναπαράσταση των δεδομένων απλοποιείται,δεν υπάρχουν πολλοί περιορισμοί και τα δεδομένα μπορούν να διαχωριστούν και να αποθηκευτούν τμηματικά σε διάφορα μηχανήματα.Το προφανές μειονέκτημα βρίσκεται στην έλλειψη ισχυρών εγγυήσεων συνέπειας,όπως για παράδειγμα οι περιορισμοί του εξωτερικού κλειδιού δεν υπάρχουν πια,καθώς και η δυσκολία εκτέλεσης συνενώσεων μεταξύ διαφορετικών πινάκων και συνόλων.Υπάρχει μια ποικιλλία από διαφορετικούς τύπυς βάσεων που περιλαμβάνονται στην κατηγορία του μοντέλου NoSQL οι πιο σημαντικές από τις οποίες είναι:

- Ζεύγη κλειδί-τιμή(key-value),όπου χρησιμοποιείται ένας πίνακας κατακερματισμού με ένα μοναδικό κλειδί και ένα δείκτη στο αντίστοιχο στοιχείο δεδομένων.Οι βάσεις κλειδί-τιμή δεν απαιτούν σχήμα και προσφέρουν μεγάλη ευελιξία και επεκτασιμότητα,δεν προσφέρουν τη δυνατότητα ατομικότητας, συνεκτικότητας, απομόνωσης, διατηρησιμότητα (ACID-Atomicity, Consistency, Isolation, Durability) και απαιτούν κάποια εργαλεία για την τοποθέτηση των δεδομένων,την αποφυγή διπλών αντιγράφων και την ανοχή σε σφάλματα καθώς όλα αυτά δεν ελέγχονται ρητά από την ίδια την τεχνολογία.Οι πιο γνωστές είναι η Memcached,η Dynamo και η Voldemort.Η S3 της Amazon χρησιμοποιεί Dynamo ως ,μηχανισμό αποθήκευσης ενώ η Riak είναι αρκετά διαδεδομένη key-value NoSQL βάση ελεύθερου λογισμικού και ανεκτική στα σφάλματα.
- Αποθήκευση κατά στήλες(columnar systems),που χρησιμοποιούνται για την αποθήκευση και επεξεργασία πολύ μεγάλων ποσοστήτων δεδομένων κατανεμημένων σε πολλά μηχανήματα.Οι σχεσιακές βάσεις είναι προσανατολισμένες κατά γραμμή καθώς τα δεδομένα σε κάθε γραμμή του πίνακα αποθηκεύονται μαζί. Σε κατά στήλη προσανατολισμένες βάσεις τα δεδομένα αποθηκεύονται κατά μήκος των γραμμών. Είναι πολύ εύκολο να προσθέσουμε στήλες και μπορούν να προστεθούν σειρά-σειρά προσφέροντας μεγαλύτερη ευελιξία,απόδοση και επεκτασιμότητα.Όταν υπάρχει μεγάλος όγκος και ποικιλία δεδομένων είναι μια πολύ καλή λύση.Είναι ευπροσάρμοστη αφού το μόνο που κάνουμε ειναι η προσθήκη κι άλλων στηλών.Το πιο σημαντικό παράδειγμα αυτής της κατηγορίας είναι το Big Table της Google όπου οι γραμμές ταυτίζονται με ένα κλειδί με τα δεδομένα να ταξινομούνται και να αποθηκεύονται από αυτό.Το Big Table αποτέλεσε τη βάση για κάποιες άλλες NoSQL βάσεις δεδομένων που

βρήκαν απήχηση στη συνέχεια όπως το Cassandra Hadoop που χρησιμοποιείται από το Facebook και τα HBase και Hypertable.

- Βάσεις δεδομένων εγγράφων(Document Databases),οι οποίες είναι παρόμοιες με τις key-value αλλά βασίζονται σε έγγραφα που αποτελούν συλλογές από άλλες key-value συλλογές,υπάρχει δηλαδή εμφώλευση.Η δομή αυτών των εγγράφων και μερών αυτών ονομάζεται JavaScript Object Notation(JSON) ή Binary JSON(BSON).Τέτοιες βάσεις είναι συνήθως χρήσιμες όταν υπάρχουν αρκετές αναφορές κι αυτές παράγονται και συναρμολογούνται από στοιχεία που αλλάζουν συχνά.Οι πιο διαδεδομένες αυτής της κατηγορίας είναι η MongoDB και η CouchDB.
- Βάσεις δεδομένων-Γράφοι (Graph Database), όπου η βασική δομή αποκαλείται και "σχέση κόμβων". Η δομή αυτή είναι χρήσιμη όταν έχουμε να κάνουμε με διασυνδεδεμένα δεδομένα.Οι κόμβοι και οι σχέσεις υποστηρίζουν κάποιες ιδιότητες, δηλαδή ένα ζεύγος key-value όπου αποθηκεύονται τα δεδομένα.Η πλοήγηση στη βάση γίνεται ακολουθώντας τις σχέσεις.Αυτού του είδους η αποθήκευση και πλοήγηση στα συστήματα RDBMS δεν είναι δυνατή εξαιτίας της ακαμψίας της δομής των πινάκων και της αδυναμίας να ακολουθηθούν οι συνδέσεις μεταξύ των στοιχείων, όπου κι αν αυτές οδηγούν.Παραδείγματα τέτοιων βάσεων είναι το Neo4J, το Allegro και το Virtuoso.

Όσον αφορά την επιλογή του συστήματος διαχείρισης δεδομένων θα πρέπει να λάβουμε υπόψην μας τρεις σημαντικούς παράγοντες ισορροπίας που συνοψίζονται στο γνωστό θεώρημα CAP.Οι παράγοντες αυτοί είναι: συνοχή (Consistency), διαθεσιμότητα(Availability) και ανοχή σε διαχωρισμό(Partition tolerance).Το θεώρημα αναφέρει ότι ένα κατενεμημένο σύστημα μπορεί ταυτόχρονα να προσφέρει μόνο 2 από τους 3 παραπάνω παράγοντες.Η συνοχή σημαίνει ότι κάθε πελάτης όλες τις στιγμές έχει την ίδια όψη για τα δεδομένα,η διαθεσιμότητα δίνει τη δυνατότητα σε όλους τους πελάτες να μπορούν να διαβάσουν και να γράψουν οποιαδήποτε στιγμή και η ανοχή στο διαχωρισμό σημαίνει ότι το σύστημα λειτουργεί καλά σε όλα τα διαχωρισμένα τμήματα του δικτύου.



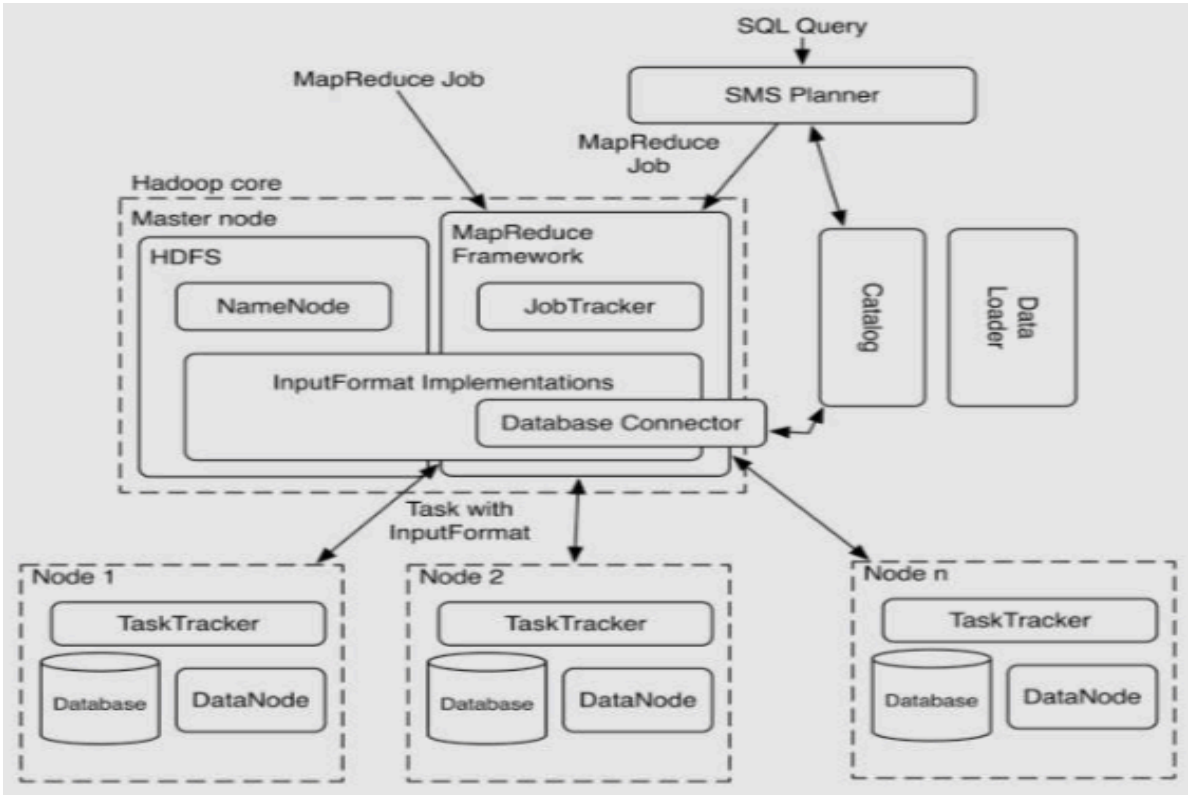
Εικόνα 3.2. Γραφική απεικόνιση του θεωρήματος CAP με NoSQL εργαλεία

Στην παραπάνω εικόνα φαίνονται τα πιο δημοφιλή προϊόντα λογισμικού NoSQL βάσεων τοποθετημένα στις τομές PA(Partition-Availability), CP(Consistency-Partition) και AC(Availability-Consistency).Γεγονός είναι ότι δεν υπάρχει μία και μοναδική επιλογή για την επίλυση προβλημάτων Big Data.Αντί αυτού υπάρχει πληθώρα διαφορετικών μοντέλων βάσεων δεδομένων καθένα από τα οποία είναι πιο ειδικό και πιο κατάλληλο για τη διαχείριση συγκεκριμένων ειδών προβλημάτων.Για παράδειγμα οι columnar databases που έχουν αρκετά μεγάλη απήχηση τελευταία ειδικεύονται σε προβλήματα που απαιτείται μεγάλη ταχύτητα πρόσβασης στα δεδομένα σε κατανεμημένα συστήματα και μπορούν να λειτουργήσουν αρμονικά με την τεχνική του MapReduce που θα εξεταστεί στη συνέχεια.Από την άλλη οι document databases λειτουργούν καλύτερα με έγγραφακαι ενσωματώνουν χαρακτηριστικά για μεγαλύτερη ταχύτητα κατά την επεξεργασία μεγάλου όγκου αντικειμένων.Τέλος οι graph databases ειδικεύονται σε δεδομένα που αναπαριστώνται με γράφους ενώ οι key-value databases είναι μια άλλη μορφή για ταχύτερη επεξεργασία μεγάλων σετ δεδομένων που έχουν σχετικά απλή μπορφή και χαρακτηριστικά.

3.3.3 Hadoop και MapReduce

Το Hadoop είναι ένα ανοιχτού λογισμικού framework γραμμένο σε Java για την εκτέλεση εφαρμογών που τρέχουν παράλληλα και σε μεγάλες συστάδες υλικού.Αναπτύχθηκε από τον Doug Cutting,ο οποίος ανέπτυξε ακόμα το Lucene,ένα εργαλείο αναζήτησης,και το Nutch,έναν κατανεμημένο web crawler.Ο Cuting επηρρεάστηκε από όσα έμαθε για το Google File System(GFS) και την τεχνολογία MapReduce μέχρι το 2006.Από τότε προσελήφθη από τη Yahoo! και με μια ομάδα μηχανικών ξεκίνησαν το Hadoop project ώστε να προσφέρει στην εταιρεία τέτοιου είδους υπηρεσίες κατανεμημένης επεξεργασίας.Σήμερα το Hadoop έχει γίνει η πρωταρχική MapReduce πλατφόρμα που χρησιμοποιείται παγκοσμίως εκτός Google.

Το Hadoop project διαχειρίζεται το Apache Software Foundation και περιλαμβάνει αρκετά άλλα πρότζεκτς όπως τα: Pig, Hive, Cassandra, HBase, Avro, Chukwa, Mahout και Zookeeper.Μάλιστα οι NoSQL βάσεις HBase και Cassandra χρησιμοποιούνται ως βάσεις δεδομένων για σημαντικό αριθμό διάφορων Hadoop projects.Όπως φαίνεται παρακάτω το Hadoop αποτελείται από τρία μέρη: το Hadoop Distributed File System (HDFS), το μοντέλο MapReduce και το Hadoop Common.



Εικόνα 3.3. HDFS Hadoop File System schema

Το HDFS είναι ένα κατανεμημένο,επεκτάσιμο και φορητό λειτουργικό σύστημα γραμμένο σε Java για το Hadoop framework.Κάθε κόμβος ενός στιγμιοτύπου έχει ουσιαστικά ένα μοναδικό Namenode,που είναι ο πυρήνας του HDFS που κρατά έναν κατάλογο,σε μορφή δενδρική,όλων των αρχείων του συστήματος και καταγράφει σε ποιες συστάδες είναι αποθηκευμένο το καθένα από τα αρχεία.Οι κόμβοι δεδομένων(DataNodes) αποθηκεύουν δεδομένα στο HDFS και το λειτουργικό σύστημα έχει περισσότερους από έναν DataNode με αντίγραφα των αρχείων.Το σύστημα χρησιμοποιεί το επίπεδο TCP/IP για την επικοινωνία,όπου οι πελάτες χρησιμοποιούν Remote Procedure Call (RPC) ώστε να επικοινωνούν μεταξύ τους.

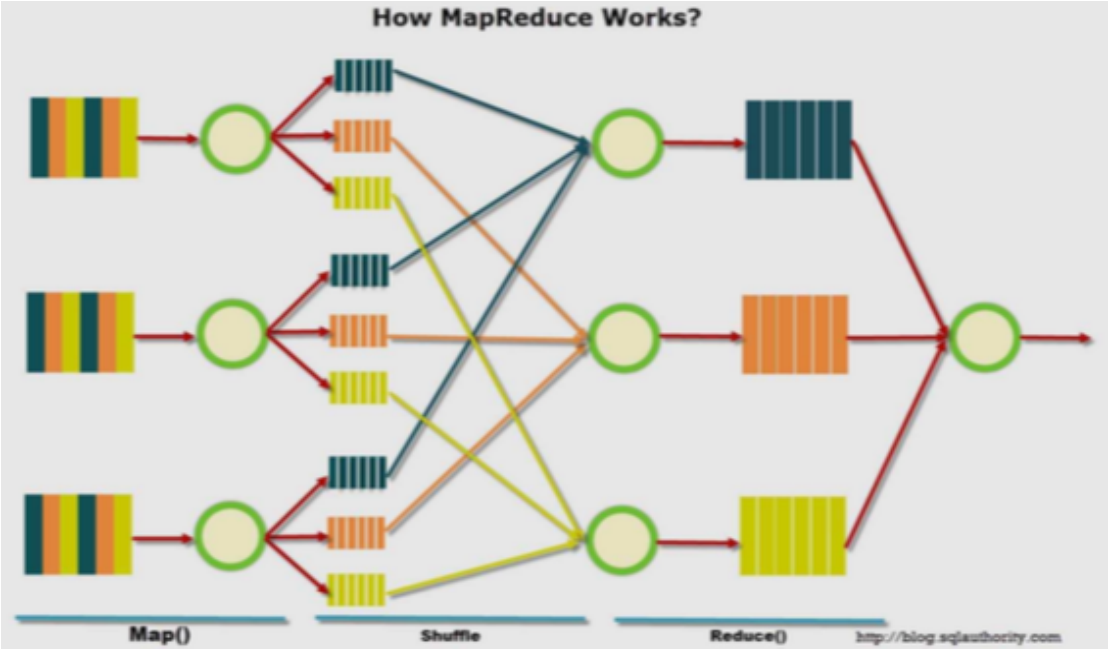
Το HDFS αποθηκεύει μεγάλα αρχεία,τυπικά από gigabytes μέχρι terabytes, κατά μήκος πολλαπλών μηχανημάτων.Επιτυγχάνεται η αξιοπιστία μέσω της αντιγραφής των δεδομένων στους διαφορετικούς hosts κι έτσι δεν απαιτείται η τεχνική RAID,μέσω αποθηκευτικών δίσκων,ώστε να φυλάσσονται αντίγραφα στους hosts.Έχοντας μια προεπιλεγμένη τιμή για τον αριθμό των αντιγράφων ,για παράδειγμα 3, αποθηκεύονται 3 αντίγραφα των δεδομένων σε 3 κόμβους ως εξής:2 από αυτά φυλάσσονται σε 2 κόμβους που ανήκουν στην ίδια συστάδα και 1 σε κάποιο κόμβο που ανήκει σε διαφορετική.Οι κόμβοι μπορούν να επικοινωνήσουν μεταξύ τους έτσι ώστε να ισορροπήσουν τον αριθμό των δεδομένων και να μετακινήσουν αντίγραφα μεταξύ τους.Το κύριο χαρακτηριστικό του HDFS είναι η δυνατότητα επέκτασής του σε εικονικά χωρίς όριο αποθήκευση απλά με την πρόσθεση καινούργιων μηχανημάτων στη συστάδα οποιαδήποτε στιγμή.

Το 2000 κάποιοι μηχανικοί της Google με βλέμμα προς το μέλλον διέκριναν ότι οι λύσεις που είχαν προταθεί μέχρι στιγμής όπως το web crawling και το query frequency ήταν επαρκείς για τις διάφορες υφιστάμενες εφαρμογές όμως από την άλλη αυξανόταν με μεγάλους ρυθμούς η πολυπλοκότητα καθώς οι χρήστες γίνονταν ολοένα και περισσότεροι.Αποτέλεσμα αυτών των παρατηρήσεων ήταν η δημιουργία ενός νέου μοντέλου όπου η δουλειά θα μοιραζόταν μεταξύ φθηνών υπολογιστών διασυνδεδεμένων στο δίκτυο με τη μορφή μιας συστάδας.Η κατανομή της εργασίας από μόνη της δεν ήταν αρκετή,αφού έπρεπε επιπλέον να γίνει παραλληλοποίηση για 3 σημαντικούς

λόγους.Πρώτον η επεξεργασία όφειλε να μπορεί να διαστέλλεται και να συστέλλεται με αυτόματο τρόπο,θα έπρεπε να μπορεί να συνεχιστεί παρόλες τις αποτυχίες του δικτύου ή κάποιου μεμονωμένου συστήματος και τέλος οι προγραμματιστές θα έπρεπε να μπορούν να αναπτύξουν υπηρεσίες που να μπρούν να χρησιμοποιηθούν κι από άλλους προγραμματιστές.Αυτή η προσέγγιση όμως θα έπρεπε να είναι ανεξάρτητη από το πού βρίσκονται τα δεδομένα και που γίνονται οι υπολογισμοί.

Έτσι το 2004 γεννήθηκε το MapReduce ένα προγραμματιστικό μοντέλο για την επεξεργασία και την παραγωγή μεγάλων σετ δεδομένων.Ο όρος κάνει λόγο για 2 ξεχωριστές και διακριτές μεταξύ τους εργασίες.Η πρώτη που είναι το Map,η χαρτογράφηση δηλαδή,που παίρνει ένα σετ δεδομένων και τα μετατρέπει σε ένα άλλο σετ δεδομένων όπου τα επιμέρους στοιχεία αναλύονται σε πλειάδες(ζευγάρια key-value).Η δεύτερη εργασία που είναι το Reduce παίρνει την έξοδο του Map ως είσοδο και συνδυάζει αυτές τις πλειάδες σε πιο μικρά σετ από πλειάδες.Όπως είναι προφανές το Reduce λαμβάνει χώρα ύστερα από το Map.Για αν λειτουργήσει αποδοτικά το μοντέλο απαιτεί έναν αλγόριθμο και περιλαμβάνει τα εξής στάνταρ βήματα:

1. Ξεκινά με ένα πολύ μεγάλο σετ δεδομένων ή αρχείων.
2. Κάνει επαναλήψεις στα δεδομένα.
3. Χρησιμοποιεί τη συνάρτηση Map για την εξαγωγή πλειάδων ενδιαφέροντος και για τη δημιουργία μια λίστας εξόδου.
4. Οργανώνει τη λίστα εξόδου για τη βελτιστοποίηση επιπλέον επεξεργασίας.
5. Χρησιμοποιεί τη συνάρτηση Reduce για τον υπολογισμό ενός σετ αποτελεσμάτων.
6. Παράγει την τελική έξοδο.



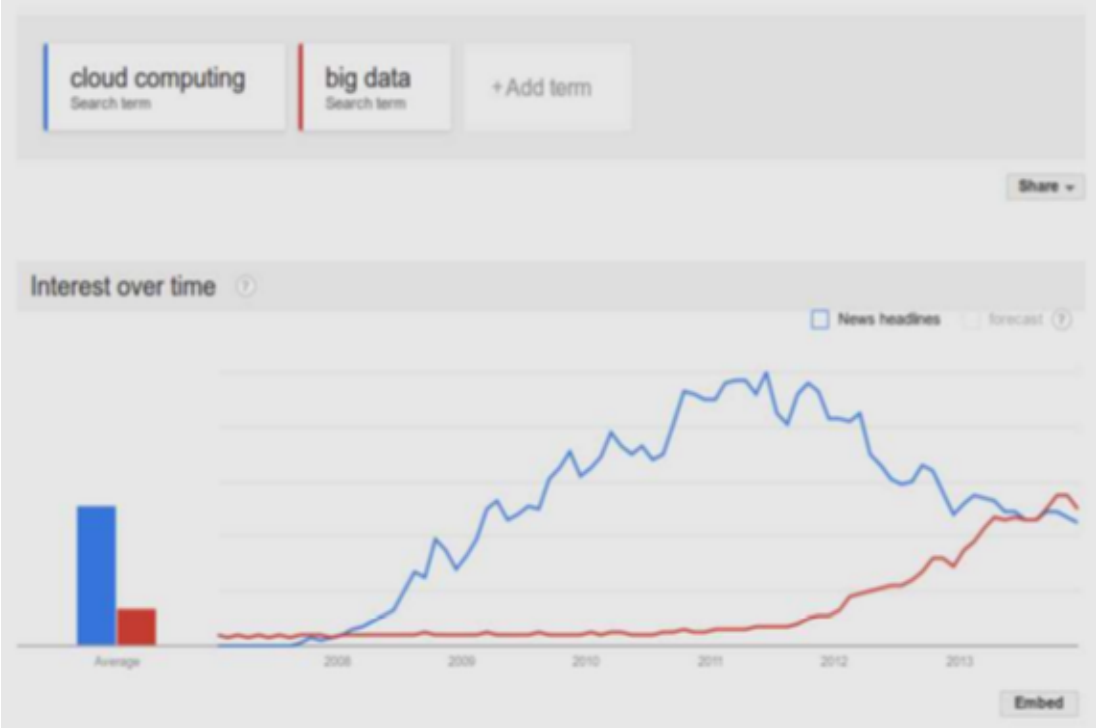
Εικόνα 3.4. Τρόπος λειτουργίας του MapReduce

Κατ'αναλογία με τη γλώσσα SQL μπορούμε να παρομοιάσουμε το Map με την εντολή "group by" και τη συνάρτηση Reduce με κάποια εντολή συνάθροισης όπως το sum ή το count,για ένα ερώτημα συνάθροισης.Παραδείγματα εργασιών του πραγματικού κόσμου που μπορούν να εκφραστούν με υπολογισμούς MapReduce υπάρχουν αρκετά,όπως η καταμέτρηση συχνότητας πρόσβασης σε διάφορα URL.Η συνάρτηση Map επεξεργάζεται το αρχείο καταγραφής των αιτήσεων στην ιστοσελίδα και τις εξόδους που παράγονται και η

συνάρτηση Reduce προσθέτει όλες τις τιμές για το ίδο URL και προκύπτει ένα (URL;total count) ζευγάρι.

3.3.4 Big Data και Cloud Computing

Το σύννεφο(cloud) έχει αναδειχτεί ως κύριος διαμεσολαβητής των Big Data τόσο στην υποδομή όσο και στα επίπεδα αναλυσης.Το Cloud προσφέρει ένα φάσμα επιλογών για την ανάλυση των Big Data.Από την πλευρά της υποδομής παρέχονται επιλογές για τη διαχείριση και πρόσβαση σε πολύ μεγάλα σύνολα δεδομένων καθώς και για την υποστήριξη ισχυρών στοιχείων υποδομών με σχετικά χαμηλό κόστος.Το περιβάλλον του Cloud λόγω της εικονικής του φύσης,της προσαρμοστικότητας και της ευελιξίας του προσφέρεται σίγουρα για το τεράστιο και μεταβαλλόμενο περιβάλλον των Big Data. Οι αρχιτεκτονικές συννέφου αποτελούνται από συστοιχίες εικονικών μηχανών και είναι ιδανικές για την επεξεργασία μεγάλων συνόλων στο βαθμό που αυτή μπορεί να υποδιαιρεθεί σε πολυάριθμες και παράλληλες διαδικασίες.Η συγγένεια αυτή ανακαλύφθηκε στο πρώιμο στάδιο ανάπτυξης του Cloud και οδήγησε απευθείας στην ανάπτυξη των Hadoop clusters που μπορούν να χρησιμοποιηθούν για υπολογιστικές αναλύσεις και στατιστικές.



Εικόνα 3.5. Google Trends: Αναζητήσεις στη μηχανή αναζήτησης για τους όρους “Cloud Computing” και “Big Data”

Κάποια σύγχρονα παραδείγματα που συνδυάζουν το Cloud Computing με τα Big Data είναι τα εξής:

- IaaS (Infrastructure as a Service) σε δημόσιο νέφος: σε αυτό το σενάριο οι εταιρείες χρησιμοποιούν το νέφος που παρέχεται από την υποδομή ενός δημόσιου φορέα για τις υπηρεσίες των Big Data που παρέχουν οι ίδιες χωρίς όμως να χρησιμοποιούν τις δικές τους φυσικές υποδομές.Με το IaaS είναι διατίθεται η δημιουργία εικονικών μηχανών με σχεδόν απεριόριστο αποθηκευτικό χώρο και υπολογιστική ισχύ.Οι χρήστες επιλέγουν το λειτουργικό σύστημα που επιθυμούν και έχουν την ευελιξία δυναμικά να

επεκτείνουν το περιβάλλον τους και να το προσαρμόσουν στις ανάγκες τους.Κλασσικό παράδειγμα IaaS είναι το Amazon EC2 που σαν υπηρεσία τρέχει σε πραγματικό χρόνο και τα δεδομένα επεξεργάζονται μαζικά και με παραλληλοποίηση.Μπορεί να χρησιμοποιηθεί για την επεξεργασία δεδομένων από λιανική πώληση όπου οι χρήστες επιθυμούν να επεγεργαστούν δισεκατομμύρια δεδομένων που απευθύνονται σε πελάτες σε πραγματικό χρόνο και θα πρέπει να γίνει σωστή διαφήμιση των προϊόντων.

- PaaS (Platform as a Service) σε ιδιωτικό νέφος: το PaaS αποτελεί ολόκληρη συσκευασμένη υποδομή έτσι ώστε να μπορεί να χρησιμοποιηθεί για το σχεδιασμό την υλοποίηση και την ανάπτυξη εφαρμογών και υπηρεσιών σε δημόσιο ή ιδιωτικό cloud περιβάλλον.Επιτρέπει σε έναν οργανισμό να κάνει τη μόχλευση βασικών υπηρεσιών ενδιάμεσου λογισμικού χωρίς να χρειάζεται να ασχοληθεί με τη διαχείριση των επιμέρους στοιχείων υλικού και λογισμικού.Οι πωλητές PaaS ήδη έχουν ενσωματώσειτεχνολογίες Big Data όπως το Hadoop και το MapReduce.Παράδειγμα εφαρμογής αποτελεί ένας οργανισμός που θέλει να αναπτύξει μια εξειδικευμένη εφαρμογή για την ανάλυση μεγάλης ποσότητας ιατρικών δεδομένων.Η χρήση της εφαρμογής περιλαμβάνει δεδομένα πραγματικού χρόνου και μη.Ένα τέτοιο σενάριο θα χρειαστεί το Hadoop MapReduce για αποθήκευση και επεξεργασία ενώ αξιοσημείωτο είναι ότι μια τέτοια εφαρμογή με χρήση PaaS μπορεί να αναπτυχθεί πολύ γρήγορα και με μεγαλύτερη ελευθερία σε πειραματισμούς αφού μόλις εντοπιστεί η κατάλληλη λύση μπορεί άνετα να υποστηριχθεί.
- SaaS (Software as a Service) σε υβριδικό νέφος: χρήσιμο για την ανάλυση δεδομένων που προέρχονται από καταναλωτές.Πολλές εταιρείες διαπίστωσαν ότι ένα από τις πιο σημαντικές πηγές πληροφοριών είναι οι γνώμες των πελατών τους για τα δικά τους προϊόντα, τις υπηρεσίες και την εταιρεία γενικότερα.Αποκτώντας πρόσβαση λοιπόν στη “φωνή του πελάτη” μπορούν να αντλήσουν πολύτιμες γνώσεις και να βελτιωθούν σημαντικά.Τέλος μπορεί να γίνει χρήση τόσο των πλατφορμών SaaS για την ανάλυση των δεδομένων από κοινωνικά δίκτυα όσο και των εταιρικών CRM ώστε να συμπεριληφθούν και τα ιδιωτικά δεδομένα στην ανάλυση.

ΚΕΦΑΛΑΙΟ 4

Ευρετηριοποίηση (Indexing)

4.1 Γιατί χρειαζόμαστε τα ευρετήρια

Όπως έχουμε ήδη αναφέρει στο προηγούμενο κεφάλαιο με τις ταχέως αναπτυσσόμενες τεχνολογίες όπως τα κοινωνικά δίκτυα, το cloud computing, τις εφαρμογές των "έξυπνων" τηλεφώνων και του μεγάλου όγκου δεδομένων, μία από τις μεγαλύτερες προκλήσεις που αντιμετωπίζουν οι αρχιτέκτονες λογισμικού είναι η διαχείριση και συνεπώς η εξαγωγή χρήσιμης πληροφορίας από τον τεράστιο όγκο δεδομένων που παράγεται παγκοσμίως.Επιπρόσθετα οι χρήστες απαιτούν οι εφαρμογές διαδικτύου που χρησιμοποιούν να ανταποκρίνονται στις ανάγκες τους και να είναι συνεχώς διαθέσιμες.Για να ανταποκρινεται σε αυτές τις απαιτήσεις ένα σύστημα θα πρέπει να διαθέτει τα εξής χαρακτηριστικά:

- Επεκτασιμότητα,η οποία μπορεί να επιτευχθεί με κατανομή των εργασιών στους πολλούς εξυπηρετητές μιας συστάδας.
- Ευκολία στην ανάπτυξη και στη ρύθμιση παραμέτρων,όπως για παράδειγμα εύκολο στην εγκατάσταση και δυνατότητα προβολής ή και αλλαγής του κώδικα(open source).
- Βέλτιστη αναζήτηση,δηλαδή ταχύτητα στην εκτέλεση πολύπλοκων ερωτημάτων εντός πολύ μικρού χρονικού διαστήματος,της τάξης των milliseconds.
- Διαχείριση τεράστιου αριθμού ευρετηριοποιημένων εγγράφων που τις περισσότερες φορές φτάνουν αρκετά εκατομμύρια.
- Κειμενο-κεντρικότητα,έτσι ώστε να γίνεται "κατανόηση" και επεξεργασία φυσικής γλώσσας σε κείμενο όπως emails, ιστοσελίδες, εγγράφων pdf, μηνύματα σε κοινωνικά δίκτυα όπως στο twitter αλλά και σε blogs.
- Κατάταξη αποτελεσμάτων με κριτήριο σχετικότητας,ύπαρξη δηλαδή ενός μέτρου σχετικότητας με βάση το οποίο φαίνεται που ανήκει το δεδομένο έγγραφο που ψάχνει ή θέλει να ταξινομήσει ο χρήστης.

Για την εύρεση λοιπόν και τη σωστή ταξινόμηση της πληροφορίας, έτσι ώστε να μπορεί να γίνει ανάκτησή της όταν χρειαστεί θα πρέπει να υπάρξει οργάνωση αλλά και ιεράρχηση.Για να επιτευχθούν λοιπόν όλα τα παραπάνω χρησιμοποιούμε την ευρετηριοποίηση ως μια αποτελεσματική τεχνική γρήγορης και ακριβούς ανάκτησης πληροφορίας.Ένα από τα εργαλεία που χρησιμοποιούνται για την ευρετηριοποίηση είναι και χρησιμοποιήσαμε και στην παρούσα εργασία είναι το Lucene.Για την ακρίβεια έγινε χρήση του Solr ενός λογισμικού που ενσωματώνει το Lucene και είναι μια "serverised" έκδοση αυτού.Στην επόμενη ενότητα θα εξηγηθούν πιο αναλυτικά κάποιες λεπτομέριες για το Lucene και το Solr.

4.2 Lucene

Το Lucene είναι μια βιβλιοθήκη ανάκτησης πληροφορίας υψηλών επιδόσεων και με πολλές δυνατότητες επεκτασιμότητας.Η ανάκτηση πληροφορίας(information retrieval- IR) αναφέρεται στη διαδικασία της αναζήτησης εγγράφων και σημαντικής πληροφορίας ,μέσα σε αυτά, ή μεταδεδομένων εγγράφων.Το Lucene ως εργαλείο επιτρέπει την ενσωμάτωση σε οποιαδήποτε εφαρμογή και παρέχει τη δυνατότητα αναζήτησης στα δεδομένα της εφαρμογής.Ως λογισμικό είναι ώριμο,δωρεάν,ανοιχτού κώδικα,γραμμένο σε Java και συντηρείται από το Apache Software Foundation.Σήμερα αποτελεί την πιο διαδεδομένη δωρεάν IR βιβλιοθήκη που κυκλοφορεί.Αρχικά γράφτηκε από τον Doug Cutting και ήταν διαθέσιμο για κατέβασμα από την ιστοσελίδα του και το SourceForge πριν γίνει μέλος του Apache Software Foundation το σεπτέμβριο του 2001.

Το Lucene προσφέρει μια απλή αλλά παντοδύναμη διεπαφή διασύνδεσης(core API) που απαιτεί την ελάχιστη δυνατή γνώση των τεχνικών ευρετηριοποίησης και αναζήτησης απο τη μεριά του προγραμματιστή.Για την ενσωμάτωσή του σε desktop εφαρμογές χρειάζεται μόνο να γνωρίζουμε σε επίπεδο κώδικα μόνο μερικές από τις πιο συνηθισμένες κλάσεις του.Ενδεικτικά να αναφερθούν μερικές πολύ γνωστά “μέρη” που συναντάμε το Lucene: NetFlix, Digg, MySpace, LinkedIn, FedEx, Apple, Ticketmaster, SalesForce.com, the Encyclopedia Britannica, Eclipse IDE, the Mayo Clinic, New Scientist magazine, Atlassian JIRA, Epiphany, MIT’s OpenCourseWare and DSpace και πολλά άλλα.

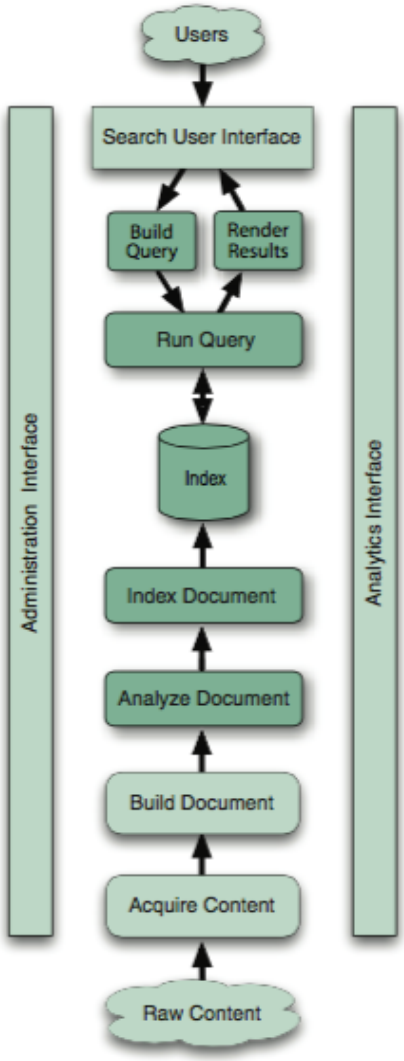
Είναι σημαντικό να γίνει κατανοητή η μεγάλη εικόνα όσον αφορά τη δομή και τα μέρη που αποτελούν το Lucene έτσι ώστε να μπορούμε να διαχειριστούμε ποιες ανάγκες της δικής μας εφαρμογής μπορούμε να καλύψουμε με τη χρήση του.Ας υποθέσουμε ότι χρειάζεται να αναζητήσουμε μέσα από μεγάλο όγκο αρχείων και να βρούμε αυτά που περιέχουν μια συγκεκριμένη λέξη ή φράση.Μια αφελής προσέγγιση θα ήταν να ψάξουμε σειριακά κάθε ένα από αυτά τα αρχεία για τη δεδομένη λέξη ή φράση.Παρόλο που αυτή η προσέγγιση θα δούλευε στην πράξη υπάρχουν πολλά μειονεκτήματα,με πιο προφανή την έλλειψη επεκτασιμότητας που προκύπτει στην περίπτωση που τα αρχεία είναι πολύ μεγάλα.

Εδώ επεμβαίνει η ευρετηριοποίηση: για να αναζητηθεί γρήγορα μεγάλου όγκου κειμενο θα πρέπει πρώτα να έχει οργανωθεί σε ευρετήριο και να έχει μετατραπεί σε μορφή που να επιτρέπει τη γρήγορη αναζήτηση εξαλείφοντας την αργή σειριακή τακτική σκαναρίσματος ένα-προς-ένα των εγγράφων.Ένα ευρετήριο είναι μια δομή δεδομένων που επιτρέπει τη γρήγορη πρόσβαση σε λέξεις που είναι αποθηκευμένες εντός του.Η φιλοσοφία είναι ανάλογη του ευρετηρίου ενός βιβλίου όπου είναι καταγεγραμμένες οι σελίδες που γράφουν για ένα συγκεκριμένο θέμα.Στην περίπτωση του Lucene το ευρετήριο είναι ειδικά σχεδιασμένη δομή που αποθηκεύεται στο σύστημα αρχείων ως ένα σετ αρχείων ευρετηρίου.

Με μια πιο προσεκτική ματιά ανακαλύπτουμε ότι η διαδικασία της ευρετηριοποίησης αποτελείται από μια σειρά από λογικά και διακριτά βήματα αρχίζοντας από την πρόσβαση στο περιεχόμενο που χρειάζεται να αναζητήσουμε. Στην εικόνα που ακολουθεί παρουσιάζονται και αναλύονται αυτά τα βήματα. Να σημειωθεί ότι θα αναλυθούν μόνο τα μέρη που αφορούν την ευρετηριοποίηση κι όχι αυτά της αναζήτησης. Το πρώτο βήμα είναι η απόκτηση του περιεχομένου που περιλαμβάνει τη χρήση ενός ανιχνευτή- “αράχνης” ο οποίος συγκεντρώνει και οριοθετεί το περιεχόμενο. Αυτή η διαδικασία μπορεί να είναι δύσκολη στην περίπτωση που για παράδειγμα θέλουμε να ευρετηριοποιήσουμε ένα σετ από XML αρχεία που βρίσκονται σε συγκεκριμένο κατάλογο στο σύστημα αρχείων ή από την άλλη εάν το περιεχόμενο βρίσκεται σε μια πολύ καλά οργανωμένη και δομημένη βάση δεδομένων.Εναλλακτικά μπορεί να είναι εξίσου δύσκολο εάν το περιεχόμενο είναι πολύπλοκο, διαφορετικών μορφών και διασκορπισμένο σε κάθε λογής κατάλογο στο σύστημα αρχείων.

Με χρήση αδειοδότησης,που σημαίνει ότι επιτρέπεται μόνο σε πιστοποιημένους χρήστες να βλέπουν συγκεκριμένα έγγραφα, μπορεί η διαδικασία πρόσβασης στα δεδομένα

να γίνει πιο πολύπλοκη.Το Lucene στον πυρήνα του,ως βιβλιοθήκη αναζήτησης, δεν προσφέρει καμία λειτουργικότητα για να υποστηρίξει το πως θα αποκτηθεί το περιεχόμενο, αφού αυτό επαφίεται αποκλειστικά στην εφαρμογή ή σε ξεχωριστό λογισμικό.Διαθέσιμοι ανιχνευτές(crawlers) ελεύθερου λογισμικού είναι μεταξύ άλλων τα Solr, Nutch, Grub, Droids κλπ.Στην περίπτωση που για το περιεχόμενό μας δεν υπάρχει διαθέσιμος διαμεσολαβητής(connector) που μπορεί να βοηθήσει τον ανιχνευτή δεν είναι δύσκολο απλά να αναπτύξουμε τον δικό μας.



Εικόνα 4.1. Μέρη που αποτελείται μια τυπική εφαρμογή.Τα σκιασμένα μέρη δείχνουν ποια από αυτά μπορεί να διχειριστεί το Lucene

Το επόμενο βήμα είναι η δημιουργία bite-sized κομματιών τα οποία ονομάσαμε προηγουμενως έγγραφα και προκύπτουν από το περιεχόμενο.Από τη στιγμή που υπάρχει το “ακατέργαστο” κείμενο που θα ευρετηριοποιηθεί,θα πρέπει να το οργανώσουμε σε μονάδες τις οποίες θα επιστρέφει η ζήτησης.Ένα έγγραφο τυπικά αποτελείται από αρκετά ξεχωριστά πεδία με όνομα και τιμή,χαρακτηριστικά παραδείγματα των οποίων είναι τίτλος,συγγραφέας,url κλπ.Ο σχεδιασμός της μορφής που θα έχουν οι μονάδες θα πρέπει να είναι προσεκτικός, περισσότερο ως προς το διαχωρισμό των πεδίων τους αλλά και τον τρόπο υπολογισμού των τιμών τους.Πολλές φορές ο διαχωρισμός είναι προφανής αφού για παράδειγμα ένα μήνυμα ηλεκτρονικού ταχυδρομείου ή ένα έγγραφο pdf αποτελεί ένα αυτοτελές έγγραφο-μονάδα.Άλλες φορές αυτό δεν είναι τόσο προφανές όπως για

παράδειγμα μια επισύναψη σε ένα μήνυμα ηλεκτρονικού ταχυδρομείου κι αν αυτή θα αποτελεί ξεχωριστό έγγραφο ή θα πρέπει κάπως να γίνει η διασύνδεση με το μήνυμα στο οποίο ανήκει.

Μόλις ξεκαθαρίσει και το σχεδιαστικό ζήτημα θα πρέπει να γίνει η εξαγωγή του κειμένου για κάθε έγγραφο από το αρχικό "ακατέργαστο" περιεχόμενο. Αν το περιεχόμενο αποτελείται από κείμενο με τη γνωστή στάνταρ κωδικοποίηση τότε δεν υπάρχει ιδιαίτερη δυσκολία. Πιο συχνά όμως έχουμε να κάνουμε με περιεχόμενο σε binary μορφή όπως για παράδειγμα PDF, αρχεία Office (Microsoft, Open), Adobe Flash, βίντεο από streaming υπηρεσίες, πολυμεσικά αρχεία είτε αρχεία που έχουν ουσιαστική σήμανση που θα πρέπει να απομακρυνθεί πριν τη δημιουργία ευρετηρίου (RDF, XML, HTML). Σε αυτές τις περιπτώσεις θα πρέπει να φιλτράρουμε τα έγγραφα πριν πάρουν την τελική τους μορφή. Ακόμη, κατά τη διάρκεια αυτού του βήματος μπορεί να εφαρμοστεί και ενδιαφέρουσα business logic στη δημιουργία επιπλέον πεδίων που να ενισχύουν τη σημασιολογία ή από την άλλη να διαχωριστούν κάποια πεδία αν κριθεί σκόπιμο.

Ένα άλλο κοινό κομμάτι στη δημιουργία των εγγράφων είναι η ενίσχυση κάποιων μεμονωμένων εγγράφων και πεδίων που είναι πιο σημαντικά, παρά να είναι όλα της ίδιας σημασίας. Για παράδειγμα τα προσφάτως τροποποιημένα έγγραφα να είναι πιο σημαντικά από τα παλαιότερα.Η ενίσχυση μπορεί να πραγματοποιηθεί τόσο στατικά (ανά έγγραφο ή ανά πεδίο) κατά την ευρετηριοποίηση ή δυναμικά κατά την αναζήτηση. Σχεδόν όλες οι μηχανές αναζήτησης, συμπεριλαμβανομένου και του Lucene, πραγματοποιούν στατική ενίσχυση των πιο μικρών πεδίων έναντι των μεγαλύτερων. Αυτό είναι λογικό γιατί είναι πιο σχετικό να υπάρχει αντιστοιχία μιας ή δύο λέξεων σε ένα μικρό έγγραφο τριών,τεσσάρων λέξεων παρά σε ένα πολύ μεγαλύτερο.

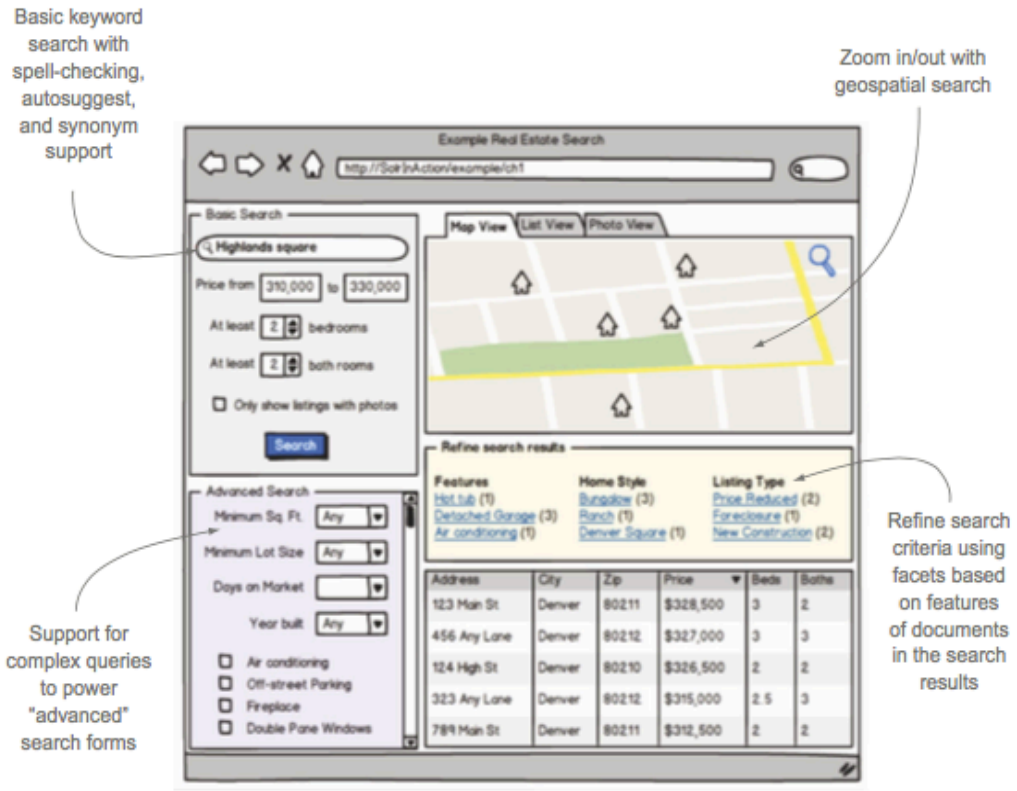
Το Lucene παρέχει μια διεπαφή (API) για τη δημιουργία των εγγράφων και των πεδίων αλλά δεν παρέχει καμία λογική στο χτίσιμο των εγγράφων,διότι όπως αναφέρθηκε, αυτό εξαρτάται από την προοριζόμενη εφαρμογή. Επίσης δεν παρέχεται κανένα φίλτρο για τα έγγραφα,παρόλο που υπάρχει το Apache Tika, ένα συμπληρωματικό λογισμικό που κάνει εξαιρετικά αυτή τη δουλειά. Τα πεδία ενός εγγράφου που περιλαμβάνουν κείμενο παρόλα αυτά δεν είναι ακόμα έτοιμα να ευρετηριοποιηθούν από τη μηχανή αναζήτησης.Για να γίνει αυτό θα πρέπει πρώτα να γίνει ανάλυση του κειμένου.

Σε αυτό το βήμα, της ανάλυσης των εγγράφων, θα πρέπει το κείμενο να τεμαχιστεί σε μια σειρά από μικρά μεμονωμένα ατομικά στοιχεία που ονομάζονται tokens. Καθένα από αυτά αποτελεί μια λέξη και σε αυτό το βήμα καθορίζεται πώς τα πεδία κειμένου του εγγράφου χωρίζονται σε σειρές από tokens. Ενδιαφέρουσες ερωτήσεις μπορούν να προκύψουν όπως πώς διαχειριζόμαστε τις σύνθετες λέξεις, αν θα πρέπει να υπάρχει ορθογραφικός έλεγχος, αν θα προτείνονται συνώνυμες λέξεις, εάν ο ενικός και ο πληθυντικός μιας λέξης αντιμετωπίζονται με τον ίδιο τρόπο ή τι ορίζεται ως λέξη σε γλώσσες που δεν είναι λατινογενείς. Το Lucene προσφέρει μια σειρά από ενσωματωμένους αναλυτές που μπορούν και ελέγχουν όλα αυτά τα θέματα. Είναι επίσης ξεκάθαρο ότι μπορούμε να αναπτύξουμε το δικό μας αναλυτή ή να δημιουργήσουμε αυθαίρετες αλυσίδες αναλυτών που συνδιάζονται με αυτούς του Lucene για να ρυθμίσουμε τον τρόπο που δημιουργούνται τα tokens. Το τελευταίο βήμα, αυτό της ευρετηριοποίησης το έγγραφο προστίθεται στο ευρετήριο. Το Lucene παρέχει ό,τι είναι απαραίτητο για το βήμα αυτό και κάτω από ένα εξαιρετικά απλό API δουλεύει κάπως...μαγικά.

4.3 Solr

Στην ενότητα αυτή θα περιγράψουμε τι είναι το Solr καθώς και κάποια χαρακτηριστικά που διαθέτει που το κάνουν χρήσιμο για το σκοπό που το χρησιμοποιήσαμε και έχουν να κάνουν με την ανάκτηση πληροφορίας. Αρχικά θα πρέπει να

(ξανα-)αναφερθεί ότι το Solr έχει χτιστεί πάνω στο Lucene που περιγράφηκε στην προηγούμενη ενότητα και δεν είναι μηχανή αναζήτησης όπως το Google ή το Bing, αν και διαθέτει κάποια κοινά με αυτά χαρακτηριστικά, ούτε έχει να κάνει με βελτιστοποίηση μηχανής αναζήτησης(search engine optimization-SEO).Ας υποθέσουμε μια εφαρμογή που διευκολύνει την αγορά σπιτιών σε υποψήφιους αγοραστές όπως φαίνεται παρακάτω:



Εικόνα 4.2. Υποθετική εφαρμογή στην οποία φαίνονται χαρακτηριστικά του Solr

Η ανάκτηση πληροφορίας είναι η εύρεση περιεχομένου, συνήθως εγγράφων, μη δομημένης μορφής, συνήθως κειμένου, που ικανοποιεί την ανάγκη παροχής αυτής της πληροφορίας που βρίσκεται σε μεγάλες συλλογές αποθηκευμένη σε υπολογιστές. Στο παράδειγμά μας η ανάγκη του χρήστη είναι να βρει ένα σπίτι για αγορά με βάση κάποια κριτήρια όπως η τοποθεσία, το είδος, τα χαρακτηριστικά και η τιμή. Το ευρετήριο αναζήτησης που διαθέτουμε περιλαμβάνει λίστες από τέτοια σπίτια που φυσικά μπορεί να χαρακτηριστεί ως "μεγάλη" συλλογή. Εν συντομία λοιπόν το Solr χρησιμοποιεί το Lucene για να παρέχει τις απαραίτητες δομές δεδομένων για την ευρετηριοποίηση των εγγράφων και την εκτέλεση αναζήτησης ώστε να ευρεθούν αυτά τα έγγραφα.

Ουσιαστικά όμως η δομή του Lucene που χρησιμοποιείται από το Solr είναι το ανεστραμμένο ευρετήριο(inverted index) λόγω των δυνατοτήτων γρήγορης εύρεσης που διαθέτει. Στην εικόνα που ακολουθεί φαίνεται η μορφή ενός ανεστραμμένου ευρετηρίου. Ενώ μια παραδοσιακή βάση δεδομένων με πολλά έγγραφα θα περιλάμβανε ένα ID του εγγράφου να αντιστοιχεί σε ένα ή περισσότερα πεδία που θα περιείχαν όλες τις λέξεις/όρους του εγγράφου αυτού, σε ένα ανεστραμμένο ευρετήριο, όπως λέει και η λέξη, αναστρέφει αυτό το μοντέλο και χαρτογραφεί την κάθε λέξη/όρο σε όλα τα έγγραφα στα οποία περιέχεται. Το αυθεντικό κείμενο απλά χωρίστηκε με κενό και όλοι οι όροι μετασχηματίστηκαν σε πεζά πρωτού εισαχθούν στο ευρετήριο και τελικά οι όροι κατατάσσονται σε αύξουσα λεξικογραφική διάταξη. Από τη δομή του το ανεστραμμένο

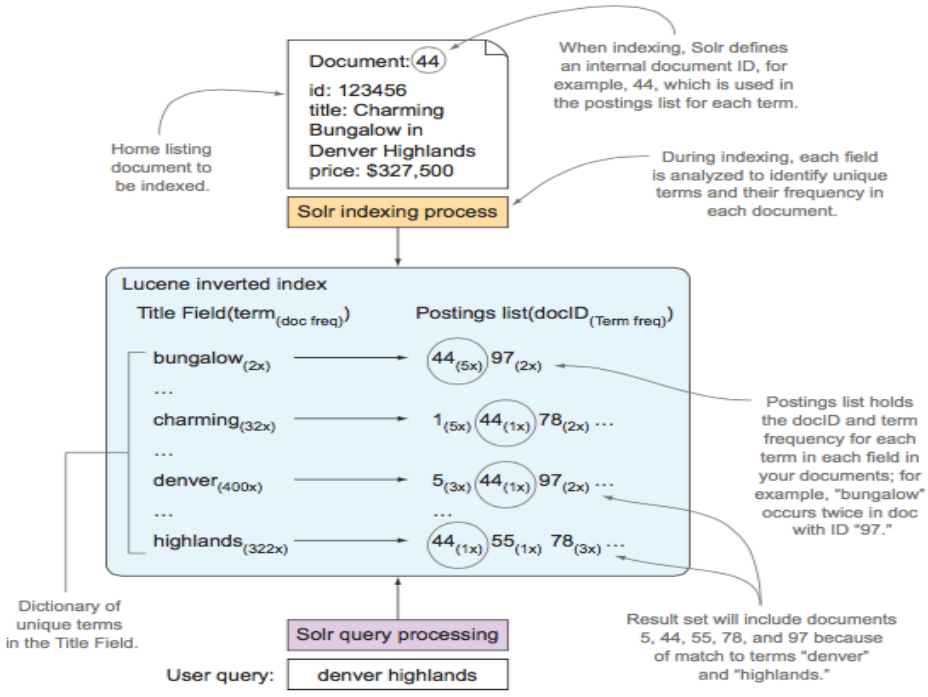
ευρετήριο δίνει πολλές δυνατότητες κατά τη διάρκεια που τίθεται το ερώτημα που μεγιστοποιούν την ταχύτητα και την ευελιξία της αναζήτησης με λέξεις-κλειδιά.

Original documents	
Doc #	**Content field**
1	A Fun Guide to Cooking
2	Decorating Your Home
3	How to Raise a Child
4	Buying a New Car
5	Buying a New Home
6	The Beginner's Guide to Buying a House
7	Purchasing a Home
8	Becoming a New Home Owner
9	How to Buy Your First House

Lucene's inverted index			
Term	**Doc #**	**(Continued)...**	
a	1,3,4,5,6,7,8	...	...
becoming	8	guide	1,6
beginner's	6	home	2,5,7,8
buy	9	house	6,9
buying	4,5,6	how	3,9
car	4	new	4,5,8
child	3	owner	8
cooking	1	purchasing	7
decorating	2	raise	3
first	9	the	6
fun	1	to	1,6,9
...	...	your	2,9

Εικόνα 4.3. Χαρτογράφηση κειμένου από πολλά έγγραφα σε ανεστραμμένο ευρετήριο.Δεξιά φαίνεται το ανεστραμμένο ευρετήριο με τον καθένα από τους όρους ενώ αριστερά φαίνονται τα πρωτότυπα έγγραφα.

Μια σχεσιακή βάση θα μπορούσε εύκολα να επιστρέψει τα ίδια αποτελέσματα χρησιμοποιώντας ένα SQL ερώτημα. Ενώ αυτό αληθεύει για μια απλή περίπτωση, η διαφορά μεταξύ ενός ερωτήματος SQL κι ενός ερωτήματος στο Lucene είναι ότι στη δεύτερη περίπτωση τα αποτελέσματα αξιολογούνται και επστρέφονται κατά σειρά σχτικότητας με το ερώτημα, σε αντίθεση με το SQL όπου τα αποτελέσματα ταξινομούνται με βάση ενός ή περισσότερων στηλών. Με άλλα λόγια η αξιολόγηση των εγγράφων είναι αυτή που διαφοροποιεί την ανάκτηση πληροφορίας από τα κλασσικού τύπου ερωτήματα των σχεσιακών βάσεων που γνωρίζαμε μέχρι τώρα.



Εικόνα 4.4. Η δομή δεδομένων που χρησιμοποιείται στο IR είναι το ανεστραμμένο ευρετήριο

Μπορεί να είναι έκπληξη το γεγονός ότι οι μηχανές αναζήτησης, όπως της Google, χρησιμοποιούν ανεστραμμένα ευρετήρια για την αναζήτηση στο διαδίκτυο, στην πραγματικότητα η ανάγκη να δημιουργήσουμε ένα μεγάλης κλίμακας ανεστραμμένο ευρετήριο οδήγησε στην επινόηση του MapReduce. Όπως έχει αναφερθεί σε προηγούμενο κεφάλαιο το MapReduce είναι ένα προγραμματιστικό μοντέλο που κατανέμει πολύ μεγάλο αριθμό εργασιών επεξεργασίας δεδομένων σε ένα σύμπλεγμα διακομιστών με χρήση ενός αλγορίθμου δύο φάσεων: του map και του reduce. Οι ρίζες του MapReduce ανήκουν στο συναρτησιακό προγραμματισμό και υιοθετήθηκε απ΄τη Google για την ανάπτυξη του ανεστραμμένου ευρετηρίου της που θα ενίσχυε την αναζήτηση στο διαδίκτυο.

Κατά τη φάση του mapping παράγεται ένας μοναδικος όρος και το ID του εγγράφου όπου προκύπτει αυτός ο όρος, ενώ κατά τη φάση του reduce οι όροι ταξινομούνται έτσι ώστε όλα τα ζεύγη να σταλούν στην ίδια διαδικασία reducer για τον κάθε μοναδικό όρο. Στην συνέχεια αθροίζονται όλες οι συχνότητες για τον κάθε όρο για να δημιουργηθεί το ανεστραμμένο ευρετήριο. Το Apache Hadoop προσφέρει μια ανοιχτού κώδικα εφαρμογή του MapReduce και χρησιμοποιείται από το Apache Nutch για να αναπτυχθεί το ανεστραμμένο ευρετήριο του Lucene για αναζήτηση ευρείας κλίμακας χρησιμοποιώντας το Solr.

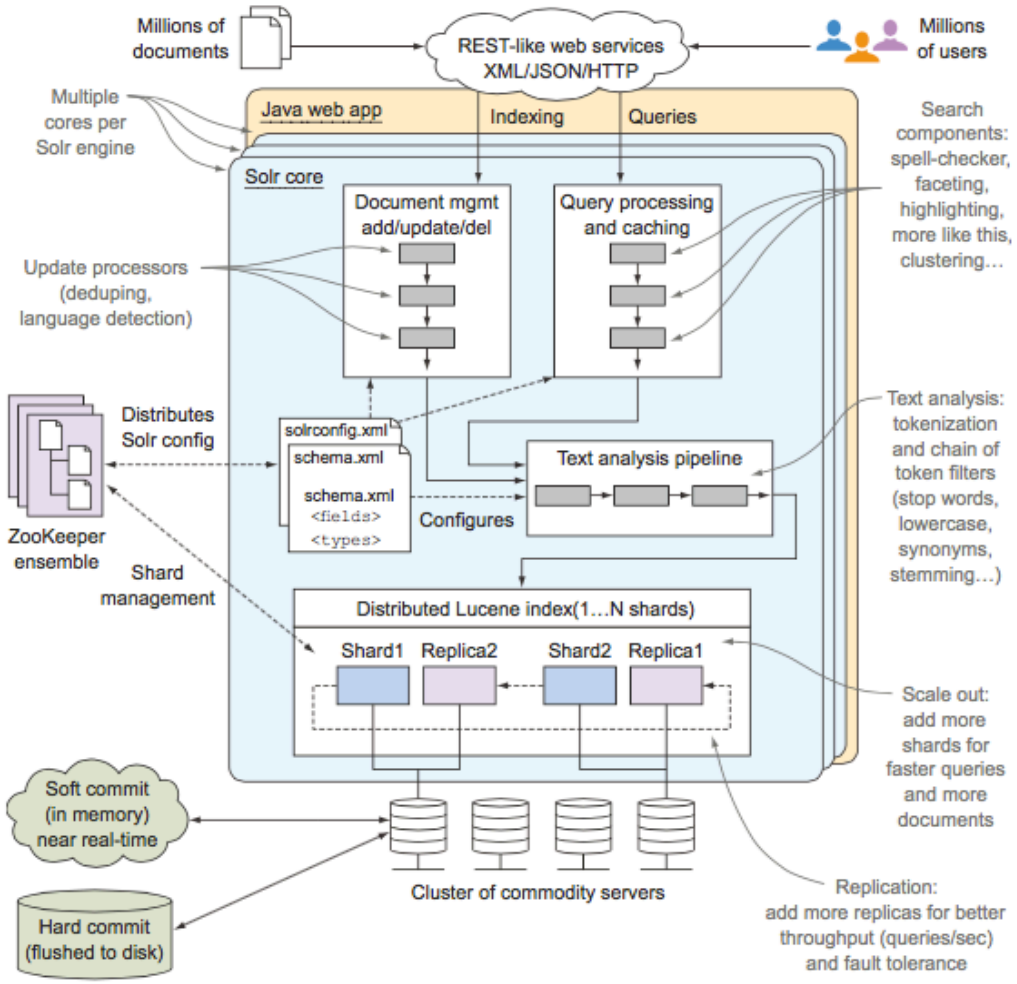
Αφού γνωρίσαμε τι προσφέρει η εσωτερική δομή του Lucene στην ευρετηριοποίηση και στην αναήτηση ας δούμε τι επιπλέον αξία προσθέτει το Solr ξεκινώντας από κάποια χαρακτηριστικά όπως το πως ορίζουμε τη δομή του ευρετηρίου. Παρόλο λοιπόν που το Lucene προσφέρει μια έτοιμη βιβλιοθήκη για την ευρετηριοποίηση των εγγράφων και την εκτέλεση ερωτημάτων αυτό που λείπει είναι ένας εύκολος τρόπος να ρυθμίσουμε πως θα δομείται το ευρετήριο. Με το Lucene θα πρέπει να γράψουμε κώδικα σε Java για να προσδιοριστούν τα πεδία και το πώς αυτά θα αναλυθούν. Το Solr έχει έναν απλό και δηλωτικό τρόπο για τον ορισμό του ευρετηρίου και αυτός είναι με ένα XML default αρχείο που ονομάζεται schema.xml. Θα δούμε παρακάτω στην εφαρμογή μας πως ορίσαμε και τι πεδία χρησιμοποιήσαμε στο schema.xml . Να σημειωθεί ότι ένα ευρετήριο που έχει δημιουργηθεί από το Solr είναι 100% συμβατό με ένα ευρετήριο που κατασκευάστηκε με στο Lucene με κώδικα.

Το Solr τρέχει ως μια Java web εφαρμογή και ενσωματώνεται με άλλες τεχνολογίες που χρησιμοποιούν standards όπως XML, JSON, HTTP. Για τη λειτουργία του χρειάζεται μια Java Servlet engine όπως είναι ο Jetty ή ο Tomcat ή έναν ολοκληρωμένο J2EE application server όπως ο JBoss ή ο Oracle AS. Το Solr προσφέρει απλές REST-like υπηρεσίες βασισμένο στο HTTP API του και έτσι είναι δυνατόν να χρησιμοποιήσουμε κατευθείαν από τη γραμμή διευθύνσεων εντολές όπως η POST ή η DELETE. Οι πιο γνωστές και συχνά χρησιμοποιούμενες γλώσσες όπως είναι οι Python, PHP, Java, .NET, Ruby όλες διαθέτουν έναν Solr client.

Σήμα κατατεθέν των σύγχρονων εφαρμογών είναι η ανάγκη για ευελιξία μιας και οι αλλαγές στις απαιτήσεις είναι ταχείς. Ένας τρόπος που το Solr βοηθά προς αυτή την κατεύθυνση είναι ότι δεν χρειάζεται να γίνουν όλα με ένα ευρετήριο γιατί το λογισμικό υποστηρίζει την ταυτόχρονη λειτουργία πολλών πυρήνων στην ίδια μηχανή. Κάθε πυρήνας διαθέτει ξεχωριστό ευρετήριο το οποίο μας επιτρέπει τη διαχείριση πολλών από αυτούς από τον ίδιο εξυπηρετητή έτσι ώστε οι πόροι και οι διαχειριστικές εργασίες και οι εργασίες συντήρησης να διαμοιράζονται.

Ενώ η επεκτασιμότητα και η ευελιξία είναι σημαντικές, η ικανότητα ανάνηψης από σφάλματα είναι καθοριστική για ένα σύγχρονο σύστημα,αν δηλαδή για παράδειγμα ένας ή περισσότεροι servers αποτύχουν. Σε τέτοια περίπτωση το Solr δε μπορεί να συνεχίσει να πραγματοποιεί ευρετηριοποίηση των εγγράφων ούτε να εκτελεί ερωτήματα. Για την αποφυγή τέτοιων καταστάσεων προστίθονται κλώνοι για κάθε server οπότε αν κάποιος αποτύχει προωθούνται τα ερωτήματα και η διαδικασία της ευρετηριοποίησης λαμβάνει χώρα στον κλώνο, και συνεπώς η συστάδα παραμένει ενεργή. Ένα μειονέκτημα από την όλη αποτυχία του server είναι ότι πλεόν η ευρετηριοποίηση και η εξυπηρέτηση των ερωτημάτων θα γίνεται πιο αργά λόγω φυσικά του ότι διαθέτουμε έναν server λιγότερο.

Παρακάτω φαίνονται σχηματικά τα κομμάτια που περιγράφηκαν για την έκδοση 4 του λογισμικού.



Εικόνα 4.5. Διάγραμμα των κύριων κομματιών του Solr 4

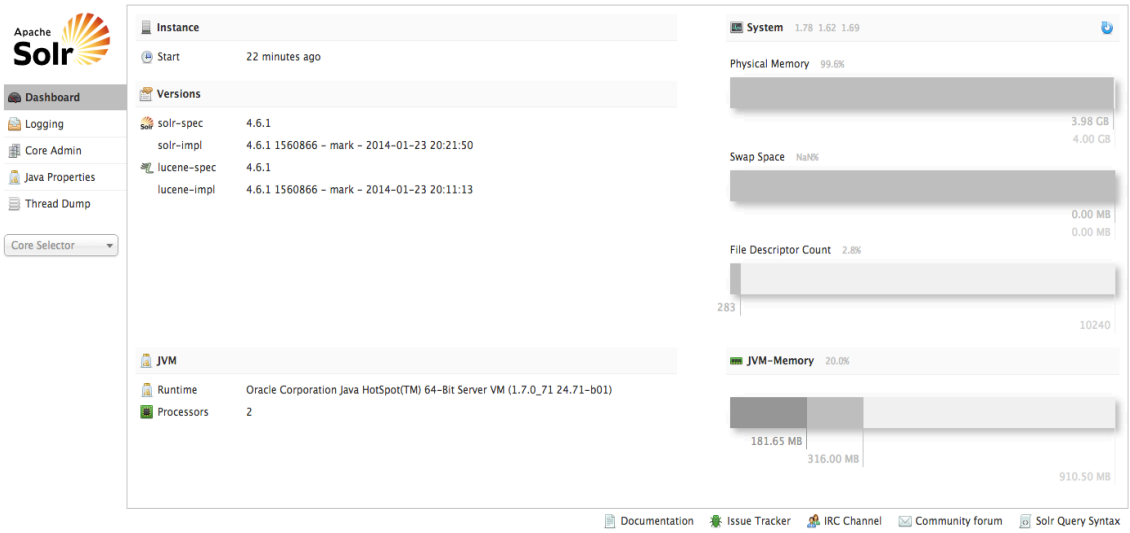
4.4 Εγκατάσταση και ρύθμιση του Solr για την εφαρμογή μας

Για τα πειράματα που πραγματοποιήσαμε στην παρούσα εργασία χρησιμοποιήθηκε η έκδοση του Solr 4.6.1 . Τα δεδομένα που χρησιμοποιήθκαν για να γίνει το ευρετήριο και να δημιουργηθούν τα έγγραφα τα αντλήσαμε από μια σχεσιακή βάση δεδομένων Postgres 9.4 που συνολικά περιείχε πάνω από 2 εκατομμύρια εγγραφές. Περισσότερα για τη βάση δεδομένων,μαζί με κάποια screenshots, θα εξεταστούν στο επόμενο κεφάλαιο.

Όπως έχει προαναφερθεί το Solr μπορεί να τρέξει τοπικά στον υπολογιστή μας σε standalone mode. Ο server ξεκινά πηγαίνοντας στο φάκελο που υπάρχει το start.jar και δίνοντας την εντολή :

```
java -jar start.jar
```

Περιμένουμε λίγο και πηγαίνοντας στη διεύθυνση του browser μας http://localhost:8983/solr/#/ βλέπουμε την παρακάτω εικόνα:



Εικόνα 4.6. Αρχική σελίδα του Solr 4.6.1 στον browser

Το Solr από τη διεπαφή του διαχειριστή (admin UI) προσφέρει έναν εύκολο και γρήγορο τρόπο να δημιουργηθούν πυρήνες , να ανιχνευτούν προβλήματα, να τεθούν ερωτήματα, να γίνει η ευρετηριοποίηση και γενικά να γίνει οποιαδήποτε άλλη εργασία κρίθεί απαραίτητο. Για την εφαρμογή μας δημιουργήσαμε κάποιους πηρήνες μέσω της διεπαφής τους οποίους και φορτώσαμε για να γίνει η ευρετηριοποίηση στη συνέχεια.

Για τη σύνδεση με την Postgres από όπου και θα “τραβήξουμε” τα δεδομένα γίνεται από τον ενσωματωμένο dataimport handler που τον ορίζουμε στο αρχείο solrconfig.xml ως εξής:

```
   <requestHandler name="/dataimport"
class="solr.DataImportHandler">
       <lst name="defaults">
        <str name="config">db-data-config.xml</str>
       </lst>
  </requestHandler>
```

και για την προσθήκη του οδηγού για σύνδεση με την Postgres προσθέτουμε το παρακάτω:

```
<luceneMatchVersion>4.6</luceneMatchVersion>
  <lib dir="/Users/L1/Desktop/Solr/solr-
4.6.1/example/solr/dih" regex=".*\.jar" />
```

Το αρχείο data-config.xml ,που ορίζει ποια πεδία από τη βάση θα αντιστοιχηθούν με του ευρετηρίου και ουσιαστικά πως θα παρουσιάζονται στο χρήστη, είναι ως εξής:

```
<dataConfig>
    <dataSource driver="org.postgresql.Driver"
   url="jdbc:postgresql://localhost:5432/nand"
user="postgres"
   password="hello" />
  <document>
```

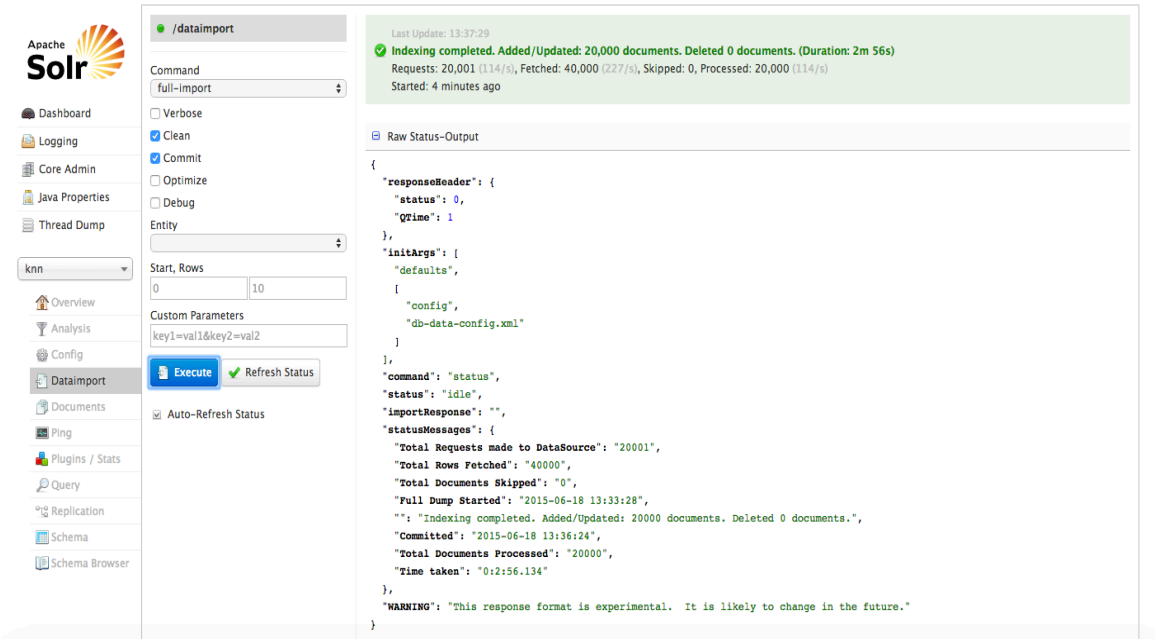
```
     <entity name="train_small" query="SELECT
catalog_product_id, pname, manufacturer_name,
catalog_vendor_id, descr, category_name FROM train_small">
       <field column="catalog_product_id" name="id" />
       <field column="pname" name="prodname" />
       <field column="manufacturer_name" name="manufname" />
       <field column="catalog_vendor_id" name="catvendid" />
       <field column="descr" name="description" />
       <field column="category_name" name="categname" />
       <entity name="catalog_product_infos" query="SELECT
image_url, vendor_catalog_url FROM catalog_product_infos
WHERE catalog_product_id=${train_small.catalog_product_id}">
         <field column="image_url" name="imgurl" />
         <field column="vendor_catalog_url"
name="vendcatalurl" />
       </entity>
     </entity>
  </document>
</dataConfig>
```

Τέλος το εξίσου σημαντικό και απαραίτητο αρχείο schema.xml ,για το οποίο έγινε λόγος και στις προηγούμενες ενότητες, το οποίο ορίζει όλα τα πεδία του ευρετηρίου, τον τύπο τους και άλλες σημαντικές ιδιότητες:

```
<field name="id" type="string" indexed="true" stored="true"
required="true" multiValued="false"/>
   <field name="categname" type="text_general"
indexed="true" stored="true" />
   <field name="text" type="text_general" indexed="true"
stored="true" multiValued="true" termVectors="true"/>

    <field name="prodname" type="text_general"
indexed="true" stored="true"/>
    <field name="manufname" type="text_general"
indexed="true" stored="true"/>
    <field name="catvendid" type="string" indexed="true"
stored="true"/>
    <field name="description" type="text_general"
indexed="true" stored="true"/>

    <field name="imgurl" type="string" indexed="true"
stored="true" />
    <field name="vendcatalurl" type="string" indexed="true"
stored="true"/>

    <field name="is_categorized" type="int" indexed="true"
stored="true"/>


    <uniqueKey>id</uniqueKey>
```

```
<copyField source="description" dest="text"/>
<copyField source="prodname" dest="text"/>
<copyField source="manufname" dest="text"/>
```

Αφού λοιπόν ολοκληρώσαμε με τις ρυθμίσεις και ορίσαμε τα πεδία που θα αντληθούν με τη βοήθεια του import handler από τη βάση κάνουμε την ευρετηριοποίηση των εγγράφων όπως φαίνεται και στην επόμενη εικόνα:



Εικόνα 4.7. Επιτυχημένη δημιουργία εγγράφων και ευρετηρίου

ΚΕΦΑΛΑΙΟ 5

Εφαρμογή

5.1 Σκοπός της εφαρμογής

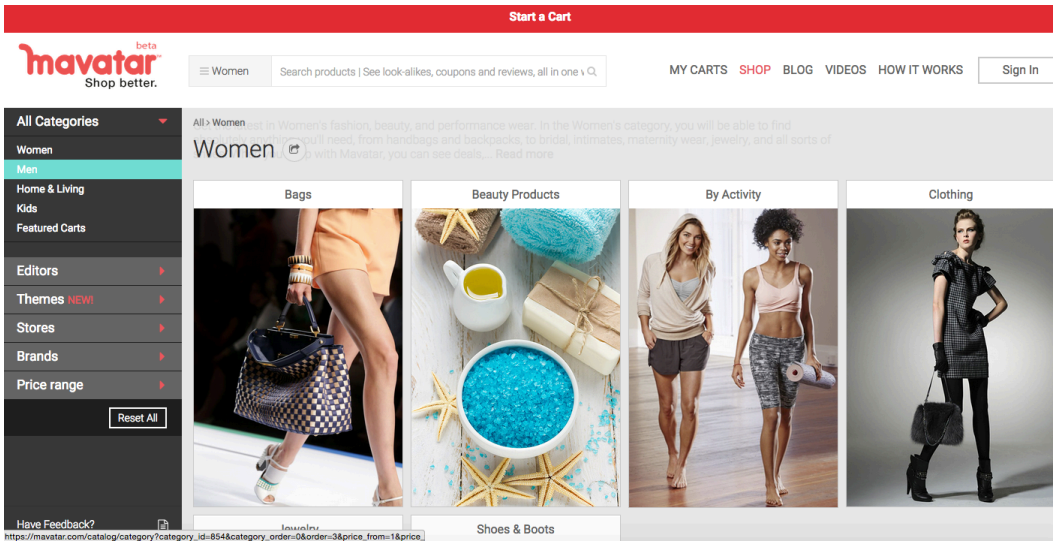
Σκοπός της εφαρμογής είναι η μελέτη δύο αλγορίθμων που περιγράφηκαν στο κεφάλαιο 2, του naive Bayes και των κ-εγγύτερων γειτόνων, για την κατηγοριοποίηση δεδομένων που αποτελούνται από κείμενο έτσι ώστε να εξαχθεί η ακρίβεια και το ποσοστό της ορθής κατηγοριοποιήσης σε μια σειρά δειγμάτων. Οι κατηγορίες που μπορεί να ανήκει ένα δείγμα είναι μία αλλά η επιλογή γίνεται μεταξύ 600 και άνω στο σύνολό τους. Η κατηγοριοποίηση τόσων πολλών προϊόντων χειροκίνητα θα ήταν πρακτικά αδύνατη, αφού θα απαιτούσε πάρα πολύ κόπο και χρόνο. Έτσι λοιπόν τώρα η διαδικασία μπορεί να αυτοματοποιηθεί και με βάση τα αποτελέσματα να εξαχθούν πολύ χρήσιμα συμπεράσματα. Ωστόσο για για να είναι χρήσιμη συνολικά η εφαρμογή θα πρέπει η ακρίβεια που δίνει να είναι αρκετά καλή. Σε αντίθετη περίπτωση οδηγεί σε λάθος κατηγοριοποιήσεις και σε λάθος συμπεράσματα. Είναι λογικό να υπάρχουν φυσικά και λανθασμένες, αλλά απαιτείται αυτές να είναι όσο το δυνατόν λιγότερες.

5.2 Δεδομένα εφαρμογής για εκπαίδευση και δοκιμή

Τα δεδομένα που χρησιμοποιήθηκαν για την εφαρμογή προέρχονται από το mavatar [16], το οποίο είναι μια ιστοσελίδα για έξυπνες αγορές μέσω διαδικτύου, όπως φαίνεται και στην εικόνα 5.1. Από τη συγκεκριμένη σελίδα είχαμε κάποια δεδομένα για διάφορα προϊόντα και κατηγορίες αποθηκευμένα σε μια βάση δεδομένων. Από αυτή τη βάση επεξεργαστήκαμε 3 πίνακες συνολικά, που φαίνονται στην εικόνα 5.2 και πήραμε με διαδοχικά joins τα πεδία: catalog_product_id , pname , manufacturer_name , catalog_vendor_id , descr , category_name , catalog_product_infos , image_url , vendor_catalog_url.

Τα δεδομένα μεταφέρθηκαν από τη βάση στο σύστημα αρχείων Hadoop, και συγκεκριμένα η έκδοση 2.6.0, έτσι ώστε να βρίσκονται στην κατάλληλη μορφή και να έχουν τον κατάλληλο χώρο(από 9GB στη βάση σε περίπου 100MB μέγιστο αρχείο στο Hadoop) για να εφαρμοστούν οι αλγόριθμοι. Για το σκοπό αυτο χρησιμοποιήθηκε το Sqoop 1.99.4 από το Hadoop Ecosystem για να μεταφερθούν σε μορφή csv(ή sequence files) με την ακολουθία εντολών που είναι διαθέσιμη στο παράρτημα.

Τα δεδομένα που θα χρησιμοποιηθούν για τα πειράματά μας θα χωριστούν σε 4 παρτίδες και το σετ για δοκιμή για την καθεμία θα είναι περίπου στο 10% από το σετ εκπαίδευσης, άρα θα έχουμε τα ζεύγη: 2.000 – 200 , 20.000 – 2.000 , 200.000 – 20.000, 1.000.000 – 100.00 έγγραφα.



Εικόνα 5.1. Mavatar shopping categories

	id [PK] se	name character varyi	parent intege	image_url character varying(255)	code character varying(	keywords text	created_at timestamp without	updated_at timestamp without	user_ integ	vault_ integ	sort_o intege	active boolean	type characte
1	991	Kids' Nurser		/assets/category/thumb/kids-an	kids-and-baby		2014-05-06 19:5	2014-05-06 19:5			3	TRUE	
2	992	Home & Livin		/assets/category/thumb/living.	living		2014-05-06 19:5	2014-05-06 19:5			4	TRUE	
3	993	Men's Fashio		/assets/category/thumb/men.jpg	men		2014-05-06 19:5	2014-05-06 19:5			2	TRUE	
4	994	Women's Fash		/assets/category/thumb/women.j	women		2014-05-06 19:5	2014-05-06 19:5			1	TRUE	
5	995	Lighting	992	/assets/category/thumb/accent-	accent-lightin	Lighting | lamp	2014-05-06 19:5	2014-05-06 19:5			0	TRUE	
6	996	Decor	992	/assets/category/thumb/accents	accents	decor | artwork | basket | clock | candl	2014-05-06 19:5	2014-05-06 19:5			0	TRUE	
7	997	Baby Gear	991	/assets/category/thumb/baby-ge	baby-gear	(baby | kid | child | infant) & (walker	2014-05-06 19:5	2014-05-06 19:5			0	TRUE	
8	998	Bath	992	/assets/category/thumb/bath.jp	bath	Bath | towels | soap | shower	2014-05-06 19:5	2014-05-06 19:5			0	TRUE	
9	999	Bath & Body	1051	/assets/category/thumb/bath-bo	bath-body	(woman | women | female) & (shower | bat	2014-05-06 19:5	2014-05-06 19:5			0	TRUE	

	id [PK] serial	catalog_product_id integer	catalog_category_id integer	type integer
1	1	313962	1445	100
2	2	313963	1445	100

Εικόνα 5.2. Πίνακες της βάσης δεδομένων που χρησιμοποιήθηκαν

5.3 Υλικό και Λογισμικό που χρησιμοποιήθηκε

Για την πραγματοποίηση των πειραμάτων χρησιμοποιήθηκε ,όσον αφορά το υλικό, υπολογιστής MacBook Pro 2.53 GHz Intel Core 2 Duo Mid 2009, ενώ για το λογισμικό εγκαταστάθηκαν: σχεσιακή βάση Postgres 9.4, το Hadoop 2.6.0 σε pseudo-distributed mode, το Sqoop 1.99.4 , το Mahout 0.9 και το Solr 4.6.1 .Μιας και δεν έχουμε αναφερθεί στο Mahout μέχρι στιγμής κι επειδή θα χρησιμοποιηθεί πάρα πολύ στα πειράματα που ακολουθούν, αξίζει να πούμε λίγα λόγια.

Το Mahout είναι μια βιβλιοθήκη αλγορίθμων μηχανικής εκμάθησης. Περιέχει αλγόριθμους που χρησιμοποιούνται για clustering, classification, recommendation και για πολλούς άλλους σκοπούς. Αυτό που κάνει το Mahout διαφορετικό από άλλες βιβλιοθήκες αλγορίθμων μηχανικής εκμάθησης είναι το γεγονός ότι οι περισσότεροι αλγόριθμοι

μηχανικής εκμάθησης που προσφέρει είναι κατανεμημένοι και κλιμακώσιμοι, γεγονός που τους καθιστά κατάλληλους για μεγάλο όγκο δεδομένων. Για να γίνει αυτό εφικτό, οι αλγόριθμοι αυτοί έχουν υλοποιηθεί σύμφωνα με το προγραμματιστικό μοντέλο MapReduce και εκτελούνται πάνω από το Hadoop. Πιο κάτω γίνεται εμβάθυνση στη χρήση του Mahout στην ταξινόμηση κειμένων, μιας και μόνο αυτό το κομμάτι του Mahout αφορά τη συγκεκριμένη εργασία.

Το Mahout μπορεί να χρησιμοποιηθεί σε πολλές διαφορετικές εφαρμογές ταξινόμησης κειμένου παρουσιάζοντας συγκριτικό πλεονέκτημα έναντι των υπόλοιπων προσεγγίσεων, όταν απαιτείται να χρησιμοποιηθεί μεγάλος όγκος δεδομένων. Λόγω των κατανεμημένων και κλιμακώσιμων αλγορίθμων που είναι διαθέσιμοι, η χρήση πολλών εγγράφων δεν απαιτεί απαγορευτικά πολύ χρόνο και μνήμη. Η εικόνα 5.3 παρουσιάζει σε ποιες περιπτώσεις το Mahout είναι καταλληλότερο από τις υπόλοιπες προσεγγίσεις:

Αριθμός εγγράφων	Σχόλια
<100 χιλιάδες	Δε συνιστάται η χρήση του Mahout. Οι υπόλοιπες προσεγγίσεις υπερτερούν σε ταχύτητα.
100 χιλιάδες μέχρι 1 εκατομμύριο	Το Mahout αρχίζει να αποτελεί καλή λύση. Τα εργαλεία που προσφέρει κάνουν το Mahout επιθυμητό, αν και δεν υπερτερεί σε ταχύτητα.
1 μέχρι 10 εκατομμύρια	Το Mahout υπερτερεί σε ταχύτητα.
>10 εκατομμύρια	Το Mahout αποτελεί ιδανική λύση. Οι περισσότερες άλλες προσεγγίσεις αποτυγχάνουν.

Εικόνα 5.3. Mahout έναντι άλλων προσεγγίσεων

Όπως παρουσιάζεται στην εικόνα, η μόνη περίπτωση που δεν προτείνεται χρήση του Mahout είναι όταν η εφαρμογή απαιτεί λιγότερα από 100 χιλιάδες έγγραφα. Σε κάθε άλλη περίπτωση, η χρήση του Mahout συνιστάται. Ειδικότερα στις περιπτώσεις όπου ο αριθμός των εγγράφων που απαιτείται είναι πάνω από 1 εκατομμύριο, η χρήση του Mahout αποτελεί μια από τις λιγοστές λύσεις.

Το Mahout, προσφέρει μια πληθώρα αλγορίθμων όπως είναι ο Naive Bayes, ο Complementary Naive Bayes ,ο Random Forests, ο Support Vector Machines (SVM) και ο Stochastic Gradient Descent (SGD), αλλά οι 4 τελευταίοι δε θα μελετηθούν σ' αυτήν την εργασία. Η εικόνα 5.4 παρουσιάζει μέχρι ποιο αριθμό εκπαιδευτικών εγγράφων συστήνεται να χρησιμοποιηθεί ο Naive Bayes, ο Complementary Naive Bayes και ο Random Forests, έτσι ώστε η εκπαίδευση τους να μη χρειάζεται αρκετό χρόνο.

Αλγόριθμος	Μέγιστος συνιστώμενος αριθμός εκπαιδευτικών παραδειγμάτων
Naive Bayes	<100 εκατομμύρια
Complementary Naive Bayes	<100 εκατομμύρια
Random Forests	<10 εκατομμύρια

Εικόνα 5.4. Μέγιστος συνιστώμενος αριθμός εκπαιδευτικών εγγράφων για τους κατανεμημένους αλγορίθμους που προσφέρει το Mahout

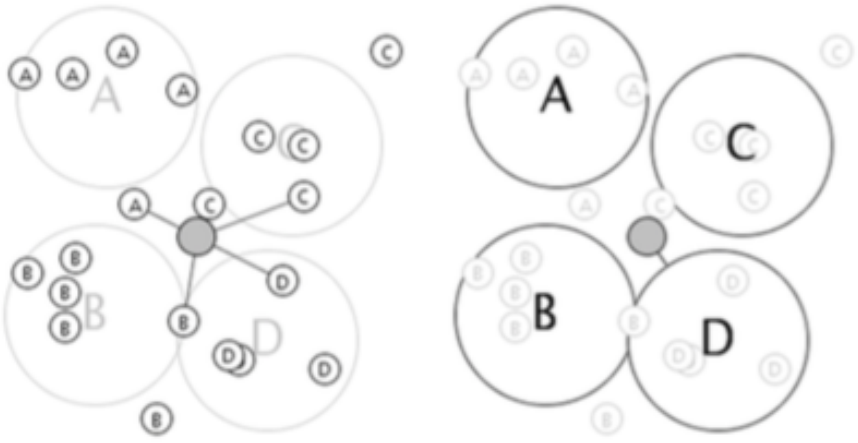
Όπως παρουσιάζεται, η χρήση του Random Forests δε συνίσταται στην περίπτωση που ο αριθμός των εκπαιδευτικών εγγράφων ξεπερνά τα 10 εκατομμύρια. Αυτό οφείλεται στο γεγονός ότι ο αλγόριθμος έχει μεγάλη υπολογιστική πολυπλοκότητα και επίσης βρίσκεται ακόμα σε στάδια υλοποίησης στο Mahout. Το γεγονός αυτό όμως, δεν καθιστά τον Random Forests ως κακή επιλογή. Ο Random Forests αποτελεί ένα από τους πιο

ακριβής ταξινομητές και μπορεί να χρησιμοποιηθεί και σε περιπτώσεις όπου οι σειριακοί αλγόριθμοι δεν είναι τόσο αποδοτικοί (περισσότερο από 1 εκατομμύριο έγγραφα). Από την άλλη, ο Naive Bayes και ο Complementary Naive Bayes μπορούν να χρησιμοποιηθούν ακόμη και στην περίπτωση που τα διαθέσιμα έγγραφα κοντεύουν τα 100 εκατομμύρια. Οι δύο αυτοί αλγόριθμοι είναι σχεδόν πανομοιότυποι, γι' αυτό σ' αυτήν την εργασία γίνεται ανάλυση και χρήση μόνο του Complementary Naive Bayes. Η διαφορά τους ωστόσο επισημαίνεται στην θεωρητική τους περιγραφή.

Όπως είδαμε κάποιοι αλγόριθμοι κατηγοριοποίησης ονομάζονται χωρικοί. Αυτοί χρησιμοποιούν το μοντέλο διανυσματικού χώρου (vector space model) για να αναπαραστήσουν το περιεχόμενο ενός εγγράφου. Οι αλγόριθμοι λοιπόν καθορίζουν την κατηγορία ενός εγγράφου μετρώντας την απόσταση ή τη γωνία μεταξύ του διανύσματος όρων του εγγράφυ που επρόκειτο να κατηγοριοποιηθεί και των άλλων διανυσμάτων που αντιπροσωπεύουν είτε έγγραφα είτε κατηγορίες.Δύο τέτοιοι αλγόριθμοι που θα μελετηθούν αι στα πειράματα είναι ο k-εγγύτερων γειτόνων και ο TF-IDF(term frequency-inverse document frequency).Καθένας από αυτούς τους 2 μεταχειρίζεται το έγγραφο προς κατηγοριοποίηση ως ένα ερώτημα και εκτελεί αναζητήσεις στο ευρετήριο ώστε να βρει παρόμοια έγγραφα.

Οι κατηγορίες των ανακτώμενων εγγράφων χρησιμοποιούνται για να καθορίσουν την κατηγορία του εγγράφου για το οποίο γίνεται το ερώτημα. Για τον αλγόριθμο των κ-εγγύτερων γειτόνων γλινεται αναζήτηση στο ευρετήριο των κατηγοριοποιημένων εγγράφων ενώ στην TF-IDF προσέγγιση γλινεται αναζλητηση στο ευρετήριο όπου κάθε έγγραφο αναπαριστά μία από τις κατηγορίες που θα αναθέσουμε. Και οι δύο έχουν πλεονεκτήματα σε ευκολία και επίδοση.

Το μοντέλο διανυσματικού χώρου είναι η καρδιά του Lucene, το οποίο έχει βελτιστοποιηθεί για να κάνει τους απαραίτητους υπολογισμούς απόστασης και για τους δύο αλγορίθμους πολύ γρήγορα. Με δεδομένο ένα ερώτημα παρόμοια έγγραφα επιστρέφονται εντός πολύ μικρού διαστήματος ακόμα κι αν το ευρετήριο περιέχει εκατομμύρια έγγραφα. Το σκορ που επιστρέφεται από το Lucene είναι το αντίστροφο της απόστασης μεταξύ δύο εγγράφων,συνεπώς όσο υψηλότερη τιμή τόσο πιο κοντινό είναι το έγγραφο στο διανυσματικό χώρο. Για κάθε αλγόριθμο τα έγγραφα που είναι πιο κοντά στο ερώτημα που τίθεται θα χρησιμοποιηθούν για να γίνει μια ανάθεση κατηγορίας σε αυτό.



Εικόνα 5.5. Σύγκριση κ-NN(αριστερά) και TF-IDF(δεξιά).

Στον αλγόρθμο κ-NN η παράμετρος κ αναφέρεται προφανώς στον αριθμό των γειτόνων που λαμβάνουμε υπ'όψιν για να παρθεί η απόφαση του ποια κατηγορία είναι η πιο κατάλληλη. Στον TF-IDF δημιουργούμε ένα μοναδικό έγγραφο για την κάθε κατηγορία που ψάχνουμε να αναθέσουμε. Μια άλλη προσέγγιση για αυτό θα ήταν να διαλέξουμε

έγγραφα – αντιπροσώπους χειροκίνητα. Η TF-IDF προσέγγιση πήρε αυτό το όνομα διότι η βαρύτητα που έχει η κάθε λέξη σε μια κατηγορία χρησιμοποιείται ως βάση για να παίρνονται οι αποφάσεις κατηγοριοποίησης. Η σχετική σημασία που έχει ένας όρος βασίζεται στον αριθμό των κατηγοριών που εμφανίζεται. Η διαφορά που περιγράφηκε μεταξύ του k-NN και του TF-IDF φαίνεται στην εικόνα 5.5 . Το έγγραφο που θα κατηγοριοποιηθεί είναι με γκρι χρώμα και το κ=5. Δύο γείτονες ανήκουν στην C κι από ένας στις A,B,D. Συνεπώς το έγγραφο θα κατηγοριοποιηθεί στην C κατηγορία. Στη δεξιά εικόνα οι μεγάλοι κύκλοι αντιπροσωπεύουν τα έγγραφα-κατηγορίες που χρησιμοποιούνται στον TF-IDF, που είναι η αλληλουχία καθενός από τα αρχικά έγγραφα με δεδομένη κατηγορία. Το αποτέλεσμα αυτού δείχνει ότι το έγγραφο προς κατηγοριοποίηση βρίσκεται πιο κοντά στην κατηγορία D, οπότε θα λάβει την ταμπέλα D, κι όχι τη C, όπως προηγουμένως.

Οι αλγόριθμοι kNN και TF-IDF μοιράζονται κοινό κώδικα, αφού και οι δύο δημιουργούν ένα ευρετήριο από τα δεδομένα. Συγκεκριμένα και στον κώδικα που χρησιμοποιήσαμε και από το Lucene API δημιουργείται ένας IndexWriter, όπου ρυθμίζει πώς αναλύεται το κείμενο και φτιάχνονται τα αντικείμενα Documents. Το περιεχόμενο των Documents ποικίλλει με βάση τον αλγόριθμο. Το ελάχιστο δυνατό που μπορεί να περιέχει είναι ένα πεδίο με την κατηγορία που ανήκει. Ο κώδικας που χρησιμοποιήθηκε για το διάβασμα, την επεξεργασία, την προσθήκη σε ευρετήριο, την κατηγοριοποίηση και την αξιολόγηση των αλγορίθμων είναι κοινός.

Για τον καθορισμό των "καλύτερων" λέξεων που βοξθούν στην κατηγοριοποίηση μπορούν να χρησιμοποιηθούν μετρικές όπως η συχνότητα εμφάνισης των όρων και η συχνότητα εγγράφων, οι οποίες υπολογίζονται ήδη ως μέρος της διαδικασίας δημιουργίας του ευρετηρίου.Το αντίστροφο της συχνότητας εγγράφων χρησιμοποιείται για την απομάκρυνση κάποιων λέξεων με μικρή σημασία, κι έτσι καταλήγουμε με μια λίστα από όρους που περιέχονται στο ευρετήριο και είναι σημαντικοί, οπότε δεν αναλώνουμε το χρόνο μας εκτελώντας queries για όρους που έχουν μικρή σημασία στο να καθορίσουν την κατηγορία ενός αντικειμένου. Το Lucene έχει ενσωματωμένο κώδικα για την επιλογή των σημαντικών όρων και ορίζει για αυτό ένα τύπο query που ονομάζεται MoreLikeThisQuery . Αυτός ο τύπος χρησιμοποιείται για την εύρεση παρόμοιων αντικειμένων με αυτό που πρόκειται να κατηγοριοποιηθεί.

5.4 Πειράματα και Συμπεράσματα

Σε αυτή την ενότητα θα παρουσιάσουμε κάποια πειράματα που έγιναν για σύγκριση των επιδόσεων των αλγορίθμων που χρησιμοποιήθηκαν. Τα δεδομένα ανήκουν σε 50 κατηγορίες συνολικά και ανάλογα με την ποσότητά τους χωρίστηκαν σε 4 παρτίδες. Να σημειωθεί ότι τα δεδομένα για εκπαίδευση και για δοκιμή στον αλγόριθμο του Bayes διαχωρίστηκαν με την εντολή split που υπάρχει στο mahout. Αυτή λειτουργεί κάνοντας τυχαία επιλογή δειγμάτων με δεδομένο το ποσοστό επί τοις εκατό στα δείγματα δοκιμής που δίνεται από το χρήστη. Θα είχε μεγάλο ενδιαφέρον να διαχωριστούν τα δεδομένα από εμάς ,χωρίς τη χρήση της split, έτσι ώστε να έχουμε ακριβώς τα ίδια δεδομένα για εκπαίδευση και δοκιμή σε όλους τους αλγορίθμους που μελετάμε. Βέβαια αυτή η διαδικασία απαιτεί αρκετό επιπρόσθετο χρόνο και δεν πραγματοποιήθηκε.

Η αξιολόγηση των πειραμάτων μετριέται με την ακρίβεια (accuracy), αλλά και με ένα πίνακα "αταξίας" (confusion matrix), όπως στην εικόνα 5.6, από όπου μπορούμε να υπολογίσουμε τις ποσότητες true positive και false negative, όπως φαίνεται ενδεικτικά στην εικόνα 5.7. Γενικά για την αξιολόγηση τα μεγέθη που χρησιμοποιούνται πιο συχνά φαίνονται παρακάτω, στην 5.8 .

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	<--Classified as		
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	| 15	a	= Aprons
3	109	1	2	0	1	2	0	2	0	0	0	0	0	0	| 120	b	= Artwork
0	0	72	0	0	0	0	0	2	0	0	0	0	0	1	| 75	c	= Backpacks
0	0	0	42	0	1	0	0	0	0	0	0	0	0	0	| 43	d	= Beds
0	0	0	0	179	0	1	0	0	0	0	0	0	0	0	| 180	e	= Belts
0	0	1	1	0	68	1	0	0	0	0	0	0	0	4	| 75	f	= Blankets
0	0	0	0	0	0	249	0	2	0	0	0	0	0	0	| 251	g	= Blazers
0	0	1	0	0	0	0	7	2	0	0	0	0	0	0	| 10	h	= Bookcases
0	0	2	1	2	0	2	1	1132	0	0	3	0	0	1	| 1144	i	= Boots
0	0	0	0	0	1	0	0	0	3	0	0	0	0	0	| 4	j	= Bouncers
0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	| 2	k	= Refillables
0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	| 3	l	= Sharpeners
0	0	0	0	0	0	0	0	1	0	0	0	14	0	0	| 15	m	= Soap
0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	| 2	n	= Swings
0	0	0	0	0	1	0	0	0	0	0	0	0	0	64	| 65	o	= Throws

Εικόνα 5.6. Confusion Matrix από ένα πείραμά μας

	Indexer assigns the category (class positive C+)	Indexer doesn't assign the category (Class negative C-)
System assigns the category (Assigned positive R+)	a (True Positive)	b (False Positive)
System doesn't assign the category (Assigned negative R-)	c (False Negative)	d (True Negative)

Εικόνα 5.7. Πίνακας ενδεχομένων

IR	AI	Formula
Recall (R)	Sensitivity	$\frac{a}{a+c}$
Precision (P)	Predictive_value(+)	$\frac{a}{a+b}$
Fallout	Predictive_value (-)	$\frac{b}{b+d}$
	Accuracy	$\frac{a+c}{a+b+c+d}$
Error_rate		$\frac{b+c}{a+b+c+d}$

Εικόνα 5.8. Μέτρα επίδοσης στην κατηγοριοποίηση όπως ορίζονται στην Τεχνητή Νοημοσύνη(AI) και στην Ανάκτηση Πληροφορίας(IR)

5.4.1 Πείραμα 1

Σε αυτό το πείραμα χρησιμοποιήθηκαν δεδομένα για εκπαίδευση της τάξης των 2000 δειγμάτων και για δοκιμή περίπου 200. Ακολουθούν τα αποτελέσματα για τον knn και στη συνέχεια για τον Bayes:

```
kNN: 60%   Vs   Complementary NaiveBayes: 70.098%


736  INFO  mlt.MoreLikeThisCategorizer  - Loaded 50 categories
from index
=======================================================
Summary
-------------------------------------------------------
Correctly Classified Instances          :        120
60%
Incorrectly Classified Instances        :         80
40%
Total Classified Instances              :        200

=======================================================

INFO test.TestNaiveBayesDriver: Complementary Results:
=======================================================
Summary
-------------------------------------------------------
Correctly Classified Instances          :        143
70.098%
Incorrectly Classified Instances        :         61
29.902%
Total Classified Instances              :        204

=======================================================

Statistics
-------------------------------------------------------
Kappa                                       0.4824
Accuracy                                    70.098%
Reliability                                52.9372%
Reliability (standard deviation)            0.4295
Weighted precision                          0.7431
Weighted recall                              0.701
Weighted F1 score                           0.7007
```

Όπως παρατηρούμε τα ποσοστά σωστής κατηγοριοποίησης και των δύο είναι αρκετά χαμηλά, γεγονός που είναι αναμενόμενο καθώς οι κατηγορίες είναι πολλές για την ποσότητα των δεδομένων που χρησιμοποιήσαμε για εκπαίδευση. Ο αλγόριθμος του Bayes συμπεριφέρεται καλύτερα μεταξύ των δύο.

5.4.2 Πείραμα 2

Σε αυτό το πείραμα χρησιμοποιήθηκαν δεδομένα για εκπαίδευση της τάξης των 20.000 δειγμάτων και για δοκιμή περίπου 2.000 .

```
kNN: 64.95%   Vs   Complementary NaiveBayes: 70.8788%
```

```
524  INFO  mlt.MoreLikeThisCategorizer  - Loaded 50 categories
from index
=======================================================
Summary
-------------------------------------------------------
Correctly Classified Instances          :       1299
64.95%
Incorrectly Classified Instances        :        701
35.05%
Total Classified Instances              :       2000

=======================================================


INFO test.TestNaiveBayesDriver: Complementary Results:
=======================================================
Summary
-------------------------------------------------------
Correctly Classified Instances          :       1105
70.8788%
Incorrectly Classified Instances        :        454
29.1212%
Total Classified Instances              :       1559

=======================================================

Statistics
-------------------------------------------------------
Kappa                                       0.6012
Accuracy                                   70.8788%
Reliability                                49.8328%
Reliability (standard deviation)            0.3838
Weighted precision                          0.7554
Weighted recall                             0.7088
Weighted F1 score                           0.7182
```

Εδώ φαίνεται ότι με το 10πλασιασμό των δεδομένων για εκπαίδευση το ποσοστό σωστής κατηγοριοποίησης για τον kNN αυξήθηκε περίπου 5%, ενώ για τον αλγόριθμο του Bayes η διαφορά είναι ανεπαίσθητη. Αυτό οφείλεται περισσότερο στο είδος των δεδομένων που περιλαμβάνει το training set μας. Θα μπορούσε δηλαδή να βρίσκονται μέσα σε αυτό αρκετές εγγραφές για κάποιες κατηγορίες και λιγότερες για κάποια άλλη, πράγμα που φυσικά οδηγεί τον αλγόριθμο σε σφάλματα. Το μόνο σίγουρο και για τους δύο είναι ότι χρειάζονται ακόμα πολύ περισσότερα δεδομένα για την εκπαίδευση.

5.4.3 Πείραμα 3

Σε αυτό το πείραμα χρησιμοποιήθηκαν δεδομένα για εκπαίδευση της τάξης των 200.000 δειγμάτων και για δοκιμή περίπου 20.000 .
kNN: 73.4597% Vs Complementary NaiveBayes: 78.5592%

```
498  INFO  mlt.MoreLikeThisCategorizer  - Loaded 50 categories
from index
=======================================================
Summary
-------------------------------------------------------
Correctly Classified Instances          :      14689
73.4597%
Incorrectly Classified Instances        :       5307
26.5403%
Total Classified Instances              :      19996

=======================================================


INFO test.TestNaiveBayesDriver: Complementary Results:
=======================================================
Summary
-------------------------------------------------------
Correctly Classified Instances          :       8266
78.5592%
Incorrectly Classified Instances        :       2256
21.4408%
Total Classified Instances              :      10522

=======================================================

Statistics
-------------------------------------------------------
Kappa                                        0.7412
Accuracy                                    78.5592%
Reliability                                 33.8549%
Reliability (standard deviation)             0.4082
Weighted precision                           0.8708
Weighted recall                              0.7856
Weighted F1 score                            0.8089
```

Όπως παρατηρούμε τα ποσοστά σωστής κατηγοριοποίησης και των δύο αυξήθηκαν αρκετά. Ακόμη βέβαια οι κατηγορίες είναι αρκετές για ην ποσότητα των δεδομένων, παρόλα αυτά σημειώθηκαν σημαντικά ποσοστά.

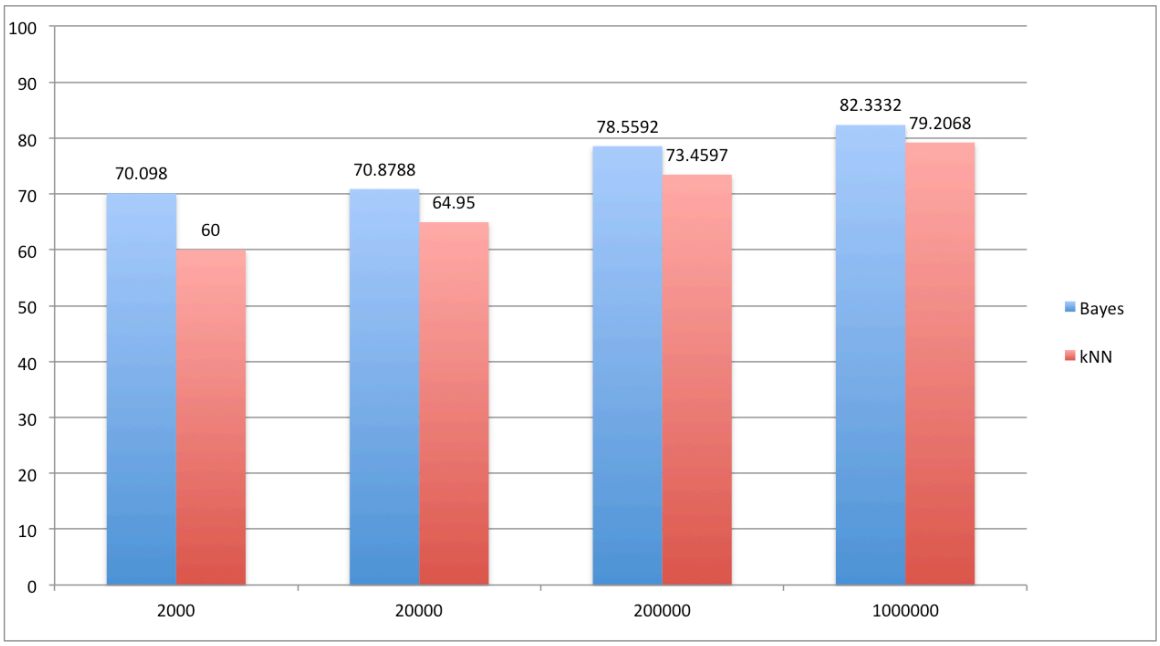
5.4.4 Πείραμα 4

Σε αυτό το πείραμα χρησιμοποιήθηκαν δεδομένα για εκπαίδευση της τάξης των 1.000.000 δειγμάτων και για δοκιμή περίπου 100.000.

```
kNN: 79.2086%   Vs   Complementary NaiveBayes: 82.3332%


2465 INFO  mlt.MoreLikeThisCategorizer  - Loaded 50 categories
from index
=======================================================
Summary
-------------------------------------------------------
Correctly Classified Instances          :      74388
79.2086%
Incorrectly Classified Instances        :      19526
20.7914%
Total Classified Instances              :      93914

=======================================================


INFO test.TestNaiveBayesDriver: Complementary Results:
=======================================================
Summary
-------------------------------------------------------
Correctly Classified Instances          :      27538
82.3332%
Incorrectly Classified Instances        :       5909
17.6668%
Total Classified Instances              :      33447

=======================================================

Statistics
-------------------------------------------------------
Kappa                                       0.7916
Accuracy                                   82.3332%
Reliability                                28.1717%
Reliability (standard deviation)            0.4082
Weighted precision                          0.9136
Weighted recall                             0.8233
Weighted F1 score                           0.8473
```

Όπως παρατηρούμε τα ποσοστά σωστής κατηγοριοποίησης και των αυξήθηκαν κι άλλο,πλησιάζοντας ,για τον knn, και ξεπερνώντας ,για τον Bayes, το 80%. Ο Bayes εξακολουθεί να συμπεριφέρεται καλύτερα αν και ο knn έχει κλείσει τη διαφορά του 10% που είχε στην αρχή. Αυτό είναι λογικό αφού ο knn χρειάζεται πολλά δεδομένα για να εκπαιδευτεί μέχρις ώτου αρχίσει να πετυχαίνει υψηλά ποσοστά. Θα μπορούσαν φυσικά να ακολουθήσουν ακόμα περισσότερα πειράματα με περισσότερα δεδομένα για αυτές τις 50

κατηγορίες, αν φυσικά τα είχαμε στη διάθεσή μας. Συγκεντρωτικά τα αποτελέσματα των πειραμάτων φαίνονται πιο καθαρά στο διάγραμμα που ακολουθεί:



5.4.5 Μερικά πειράματα ακόμη

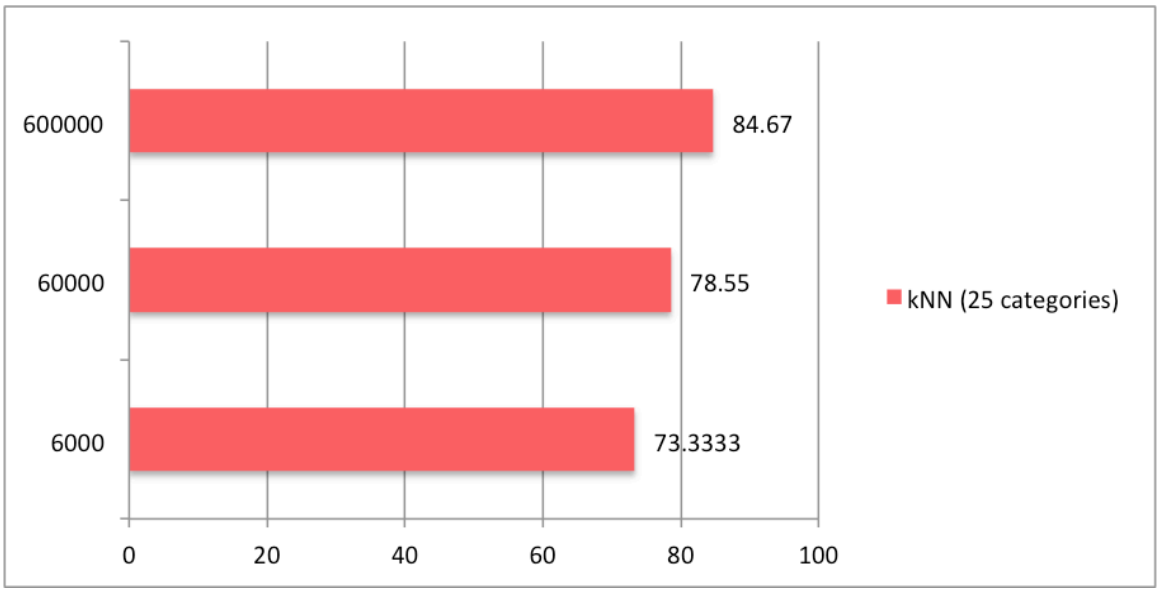
Οι επόμενες δοκιμές που ακολουθούν έγιναν για σύγκριση με τα πειράματα που προηγήθηκαν.

Πραγματοποιήθηκαν 3 προσομοιώσεις για τον αλγόριθμο knn, με 25 κατηγορίες προϊόντων αυτή τη φορά και με τα εξής ζεύγη δεδομένων εκπαίδευσης-δοκιμής: 6.000 – 600 , 60.000 – 6.000, 600.000 – 60.000.

```
kNN1: 73.3333%  Vs   kNN2: 78.55%  Vs   kNN3: 84.675%


970  INFO  mlt.MoreLikeThisCategorizer  - Loaded 25 categories
from index
=======================================================
Summary
-------------------------------------------------------
Correctly Classified Instances          :        440
73.3333%
Incorrectly Classified Instances        :        160
26.6667%
Total Classified Instances              :        600

=======================================================
```

```
920  INFO  mlt.MoreLikeThisCategorizer  - Loaded 25 categories
from index
=======================================================
Summary
-------------------------------------------------------
Correctly Classified Instances          :       4713
78.55%
Incorrectly Classified Instances        :       1287
21.45%
Total Classified Instances              :       6000

=======================================================


542  INFO  mlt.MoreLikeThisCategorizer  - Loaded 25 categories
from index
=======================================================
Summary
-------------------------------------------------------
Correctly Classified Instances          :      50396
84.675%
Incorrectly Classified Instances        :       9121
15.325%
Total Classified Instances              :      59517

=======================================================
```

Παρατηρούμε ιδιαίτερα αρκετά αυξημένα ποσοστά για τον knn σε σύγκριση με τα ίδιας τάξης μεγέθους, σε ποσότητα δεδομένων εκπαίδευσης, των προηγούμενων πειραμάτων των 50 κατηγοριών. Η διαφορά αυτή κυμαίνεται από 6% - 10% περίπου, άνοδο στα ποσοστά σωστής κατηγοριοποίησης. Συγκεντρωτικά τα αποτελέσματα των πειραμάτων φαίνονται πιο καθαρά στο διάγραμμα που ακολουθεί:

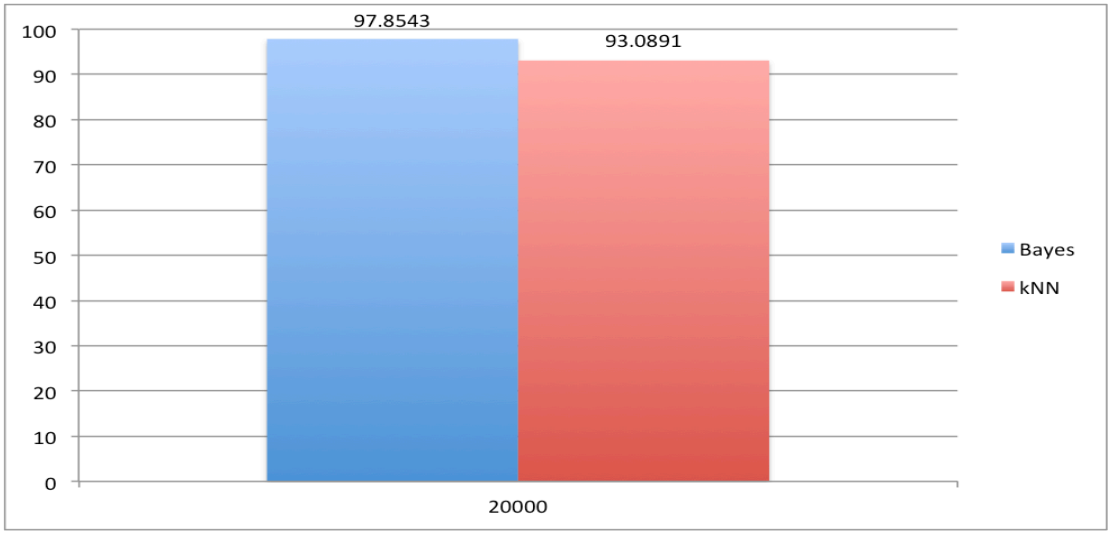


Το επόμενο πείραμα που ακολουθεί μεταξύ των δύο αλγορίθμων που εξετάσαμε, περιλαμβάνει 15 κατηγορίες αρκετά διαφορετικές μεταξύ τους , με ποσότητα δεδομένων εκπαίδευσης 20.000 έγγραφα και περίπου 2.000 έγγραφα για δοκιμή.

kNN: 93.0891% Vs Complementary NaiveBayes: 97.8543%

```
854  INFO  mlt.MoreLikeThisCategorizer  - Loaded 15 categories
from index
=======================================================
Summary
-------------------------------------------------------
Correctly Classified Instances          :       1347
93.0891%
Incorrectly Classified Instances        :        100
6.9109%
Total Classified Instances              :       1447

=======================================================


INFO test.TestNaiveBayesDriver: Complementary Results:
=======================================================
Summary
-------------------------------------------------------
Correctly Classified Instances          :       1961
97.8543%
Incorrectly Classified Instances        :         43
2.1457%
Total Classified Instances              :       2004

=======================================================
=======================================================
Statistics
-------------------------------------------------------
Kappa                                       0.9368
Accuracy                                   97.8543%
Reliability                                 88.098%
Reliability (standard deviation)            0.2517
Weighted precision                          0.9798
Weighted recall                             0.9785
Weighted F1 score                           0.9787
```

Παρατηρούμε φυσικά εκπληκτικές επιδόσεις και για τους δύο αλγορίθμους, με πάνω από 93% . Μεγάλο ρόλο παίζει τόσο η μείωση των κατηγοριών επιλογής, όσο και το είδος των κατηγοριών, αφού υπάρχουν και κατηγορίες που δεν ανήκουν στην ένδυση, συνεπώς οι λέξεις που χρησιμοποιούνται και αξιολογούνται είναι διαφορετικές μεταξύ των κατηγοριών, γεγονός που διευκολύνει τη σωστή επιλογή κατηγορίας από τους αλγορίθμους. Συγκεντρωτικά τα αποτελέσματα των πειραμάτων φαίνονται πιο καθαρά στο διάγραμμα που ακολουθεί:

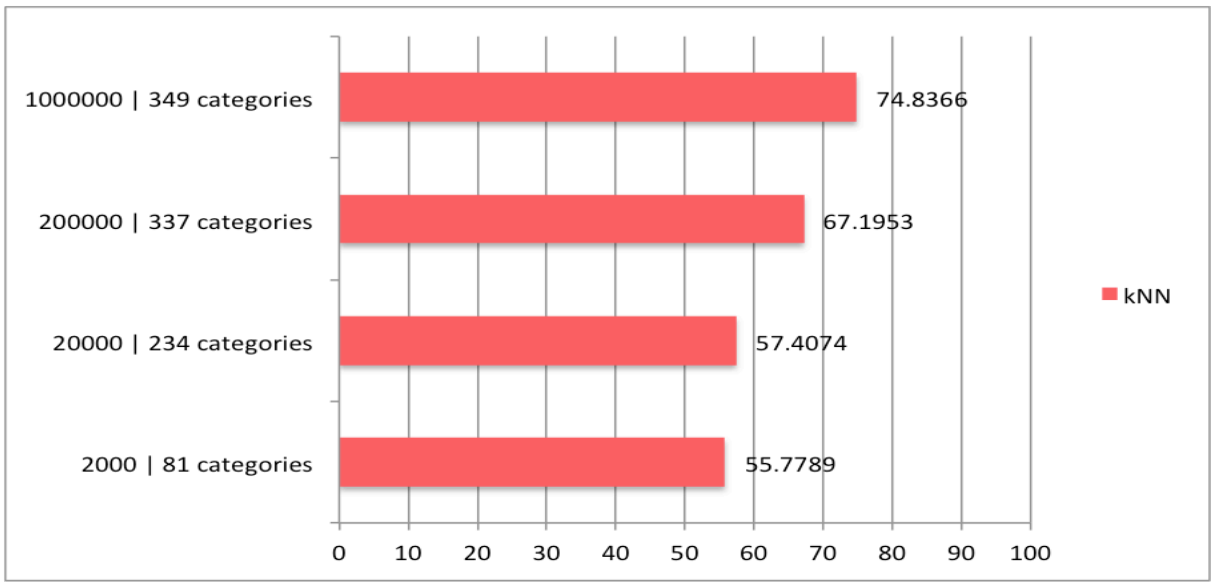


Το επόμενο πείραμα που ακολουθεί εξετάζει τη συμπεριφορά του knn και περιλαμβάνει πάρα πολλές κατηγορίες για την καθεμιά από τις 4 παρτίδες δεδομένων.Συγκεκριμένα:

```
kNN: 55.7789%   [81 categories, training: 2.000 testing:200]

1448 INFO  mlt.MoreLikeThisCategorizer  - Loaded 81 categories
from index
=======================================================
Summary
-------------------------------------------------------
Correctly Classified Instances          :        111
55.7789%
Incorrectly Classified Instances        :         88
44.2211%
Total Classified Instances              :        199

   =======================================================

kNN: 57.4074%  [234 categories, training: 20.000 testing:2.000]

484  INFO  mlt.MoreLikeThisCategorizer  - Loaded 234 categories
from index
=======================================================
Summary
-------------------------------------------------------
Correctly Classified Instances          :       1147
57.4074%
Incorrectly Classified Instances        :        851
42.5926%
Total Classified Instances              :       1998

     =======================================================
```

```
kNN: 67.1953%  [337 categories, training: 200.000 testing:20.000]


397  INFO  mlt.MoreLikeThisCategorizer  - Loaded 337 categories
from index
=======================================================
Summary
-------------------------------------------------------
Correctly Classified Instances          :      13431
67.1953%
Incorrectly Classified Instances        :       6557
32.8047%
Total Classified Instances              :      19988

      =======================================================

kNN:74.8366% [349 categories,training: 1.000.000 testing:100.000]


710  INFO  mlt.MoreLikeThisCategorizer  - Loaded 349 categories
from index
=======================================================
Summary
-------------------------------------------------------
Correctly Classified Instances          :      74773
74.8366%
Incorrectly Classified Instances        :      25142
25.1634%
Total Classified Instances              :      99915

      =======================================================
```

Παρατηρούμε ότι ακόμα κι αν οι δυνατότητες επιλογής κατηγορίας του αλγορίθμου γίνουν πάρα πολλές προσφέροντας περισσότερα δεδομένα καταφέρνουμε να βελτιώσουμε την επίδοσή του. Υπάρχουν ακόμα όμως πολλά περιθώρια βελτίωσης, όπως έχει περιγραφεί ως τώρα. Συγκεντρωτικά τα αποτελέσματα των πειραμάτων φαίνονται πιο καθαρά στο διάγραμμα που ακολουθεί:



ΚΕΦΑΛΑΙΟ 6

Επίλογος

6.1 Γενικά Συμπεράσματα

Ολοκληρώνοντας την παρούσα εργασία, θα θέλαμε να επικεντρωθούμε επιγραμματικά σε μερικά σημεία:

- Η χρήση του Mahout μας βοήθησε πάρα πολύ, αφού διαθέτει μια πλούσια βιβλιοθήκη αλγορίθμων μηχανικής εκμάθησης κι έτσι εξοικονομήσαμε χρόνο και κόπο.

- Ο Naïve Bayes που είναι υλοποιημένος στο Mahout είναι κατανεμημένος και συνίσταται για την κατηγοριοποίηση κειμένου όταν τα διαθέσιμα έγγραφα ξεπερνούν τις 500χιλιάδες.

- Τα ευρετήρια όχι μόνο διευκολύνουν την αναζήτηση αλλά βρίσκουν και πολύ μεγάλη εφαρμογή σε προβλήμα μηχανικής μάθησης ,μεταξύ άλλων.

- Γενικά ισχύει ότι "Περισσότερα δεδομένα νικούν έναν πιο έξυπνο αλγόριθμο".Παρόλα αυτά η χρήση μεγάλου όγκου δεδομένων δεν αποτελεί πάντα λύση στην αύξηση της ακρίβειας και αυτό το είδαμε στην παρούσα εργασία.

- Η συλλογή και ο καθαρισμός των δεδομένων έτσι ώστε αυτά να μπορούν να αξιοποιηθούν κατάλληλα αποτελεί μια επίπονη διαδιακασία.

6.2 Επεκτάσεις

Η περαιτέρω διερεύνηση του αντικειμένου της εργασίας θα μπορούσε να περιλαμβάνει συγκρίσεις μεταξύ περισσότερων κατανεμημένων αλγορίθμων που προσφέρει το Mahout. Συγκεκριμένα, οι συγκρίσεις αυτές θα μπορούσαν να λάβουν υπόψη τόσο την ακρίβεια ταξινόμησης, όσο και το χρόνο εκτέλεσης. Γενικά, οι συγκρίσεις αυτές μπορούν να προσφέρουν πολύ χρήσιμα συμπεράσματα. Για παράδειγμα, αν και ο Random Forests έχει μεγαλύτερη χρονική πολυπλοκότητα από το Naive Bayes, αυτό δε σημαίνει πως θα χρειάζεται μεγαλύτερο χρόνο από το Naive Bayes στην περίπτωση μιας εφαρμογής. Ο λόγος που μπορεί να συμβαίνει αυτό είναι γιατί ο Random Forests μπορεί να απαιτεί πολύ λιγότερα εκπαιδευτικά έγγραφα από τον Naive Bayes για να πετύχει παρόμοια ακρίβεια ή ακόμη και μεγαλύτερη. Μια ακόμη ενδιαφέρουσα έρευνα για τη βελτίωση της ακρίβειας του ταξινομητή θα ήταν η εφαρμογή τεχνικών όπως η αφαίρεση συμβόλων και αριθμών, η

αφαίρεση stopwords, η κανονικοποίηση, η χρήση n-gram, το stemming, οι ελάχιστες και βέλτιστες εμφανίσεις κι ακόμα ο πειραματισμός για την εύρεση της βέλτιστης παραμέτρου k, στον κ-NN. Τέλος μια ακόμα επέκταση, που δεν αφήνεται ως ανοιχτό ζήτημα αλλά θα υλοποιηθεί στο άμεσο μέλλον, είναι η δημιουργία μιας online web εφαρμογής που θα τρέχει σε cloud based περιβάλλον και θα περιέχει μια φιλική προς το χρήστη διεπαφή με τη δυνατότητα να εισάγεται ένα προϊόν με την περιγραφή του σε ένα πεδίο και με επιλογή αλγορίθμου από τους διαθέσιμους αυτό να κατηγοριοποιείται από το σύστημα.

Παράρτημα

--------- Ακολουθία εντολών για τη μεταφορά των δεδομένων από Postgres στο Hadoop με το Sqoop ---------

```
sh ./sqoop2-server start
sh ./sqoop2-shell
create link -c 1
Creating link for connector with id 1
Please fill following values to create new link object
Name: hadoop

Link configuration

HDFS URI: hdfs://localhost:9000
New link was successfully created with validation status OK
and persistent id 7
sqoop:000> create link -c 2
Creating link for connector with id 2
Please fill following values to create new link object
Name: psgr

Link configuration

JDBC Driver Class: org.postgresql.Driver
JDBC Connection String:
jdbc:postgresql://localhost:5432/nand
Username: postgres
Password: *****
JDBC Connection Properties:
There are currently 0 values in the map:
entry# protocol=tcp
There are currently 1 values in the map:
protocol = tcp
entry#
New link was successfully created with validation status OK
and persistent id 8
sqoop:000> create job -f 8 -t 7
Creating job for links with from id 8 and to id 7
Please fill following values to create new job object
Name: textF_train

From database configuration

Schema name: public
Table name: train_small
Table SQL statement:
Table column names:
Partition column name: catalog_product_id
```

```
Null value allowed for the partition column:
Boundary query:

ToJob configuration

Output format:
  0 : TEXT_FILE
  1 : SEQUENCE_FILE
Choose: 0
Compression format:
  0 : NONE
  1 : DEFAULT
  2 : DEFLATE
  3 : GZIP
  4 : BZIP2
  5 : LZO
  6 : LZ4
  7 : SNAPPY
  8 : CUSTOM
Choose: 0
Custom compression format:
Output directory: /user/L1/train_small_txt

Throttling resources

Extractors:
Loaders:
New job was successfully created with validation status OK
and persistent id 5
sqoop:000> start job -j 5
Submission details
Job ID: 5
Server URL: http://localhost:12000/sqoop/
Created by: L1
Creation date: 2015-05-21 23:53:33 EEST
Lastly updated by: L1
External ID: job_1432212374464_0001
    http://Elena-Tsiligiannis-MacBook-
   Pro.local:8088/proxy/application_1432212374464_0001/
```

--------- Μεταφορά των sequence files των δεδομένων ,που δημιουργήθηκαν από το Lucene, στο Hadoop File System ---------

```
package test;

import java.io.IOException;
import java.net.URI;

import org.apache.hadoop.conf.Configuration;
```

```
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IOUtils;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.SequenceFile;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.io.Writable;
import org.apache.hadoop.util.ReflectionUtils;

/**
** @author Eleni Tsiligianni  (elentsilig@hotmail.com)
**/
public class seqWrite {

          static long sum=1;
       public   static   void   main(String[]   args)   throws
IOException {
              String uri = args[0]; //from
              String uri2 = args[1];
              Configuration conf = new Configuration();
              FileSystem                fs                =
FileSystem.get(URI.create(uri), conf);
              Path path = new Path(uri);

              //String uri2 = args[1]; //to
              Configuration conf2 = new Configuration();
              FileSystem                fs2               =
FileSystem.get(URI.create(uri2), conf2);
              Path path2 = new Path(uri2);
//SequenceFile.createWriter( fs, conf, path, key.getClass(),
value.getClass());
//Ta Writable key kai value ektos try?

              SequenceFile.Reader reader = null;
              SequenceFile.Writer writer = null;
              try {
                reader  =  new  SequenceFile.Reader(fs,  path,
conf);
                Writable key = (Writable)

ReflectionUtils.newInstance(reader.getKeyClass(), conf);
                Writable value = (Writable)

ReflectionUtils.newInstance(reader.getValueClass(), conf);
                writer    =    SequenceFile.createWriter(fs2,
conf2, path2,
                    key.getClass(), value.getClass());
                Writable value2;
                Text key2=new Text();

                long position = reader.getPosition();
```

```
                while (reader.next(key, value)) {
                  String syncSeen = reader.syncSeen() ? "*"
: "";
                  System.out.printf("[%s%s]\t"+"/%s\t%s\n",
position, syncSeen, key, value);
                  //key2=key;
                  String id = String.valueOf(sum++);
                  String posId = String.valueOf(position);
                  key2.set("/"+key+"/"+posId);
                  value2=value;
                  System.out.printf("%s\t%s\n", key, value);
                 // System.out.printf("End of Write");
                  writer.append(key2, value);
                  position   =   reader.getPosition();   //
beginning of next record
                    }
               } finally {
                IOUtils.closeStream(reader);
                IOUtils.closeStream(writer);
                System.out.printf("End of Write");
                        }
      }
}
```

--------- Μετατροπή ευρετηρίου από την έκδοση Lucene 3.6 στην 4.6 για να μπορεί να διαβαστεί από την τρέχουσα δική μας ---------

```
//Convert LUCENE 3.6 index -> LUCENE 4.6 index
//Tha prepe i morfi na einai idia
//lucene-core-3.6 jar added

package test;

import java.io.File;

import org.apache.lucene.index.IndexUpgrader;
import org.apache.lucene.store.Directory;
import org.apache.lucene.store.FSDirectory;
import org.apache.lucene.util.Version;



public class IndexUpgrade {
     public static void main(String[] args) throws Exception
{
          File indexDir = new File("knn-index");
```

```
        Directory            myIndexDirectory            =
FSDirectory.open(indexDir);
          IndexUpgrader        upgrader        =        new
IndexUpgrader(myIndexDirectory, Version.LUCENE_46);
          upgrader.upgrade();
          System.out.println("All OK");
     }

}
```

--------- Bash script για τη δημιουργία της κατάλληλης μορφής των δεδομένων (καθαρισμός και απαραίτητα πεδία) που πήραμε από την Postgres, πριν εφαρμόσουμε τους αλγορίθμους k-NN και TF-IDF του Lucene ---------

```
while IFS=";" read a b c d e f; do
     echo "'$f'\t$b,$c,$e" >> "$f"
done < test_small_pstgr.csv
```

Βιβλιογραφία

[1] http://en.wikipedia.org/wiki/Machine_learning

[2] Jason Bell, "Machine Learning: Hands-On for Developers and Technical Professionals", Wiley, 2014

[3] Alex Smola, S.V.N. Vishwanathan, "Introduction to Machine Learning", Cambridge University Press, 2008

[4] Jan van Leeuwen, "Approaches to machine learning", Philips SOIA, 2002

[5] http://en.wikipedia.org/wiki/Statistical_classification

[6] Mohammed J. Zaki, Wagner Meira Jr, "Data Mining and Analysis: Fundamental Concepts and Algorithms", Cambridge University Press, 2014

[7] Tom M. Mitchell, "Machine Learning", McGraw-Hill, 1997

[8] Jiawei Han, Micheline Kamber, Jian Pei, "Data Mining - Concepts and Techniques", Morgan Kaufmann, 2011

[9] Ian H. Witten, Eibe Frank, Mark A. Hall, "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, 2011

[10] http://en.wikipedia.org/wiki/Overfitting

[11] Kevin P. Murphy, "Machine Learning:A Probabilistic Perspective", The MIT Press, 2012

[12] http://en.wikipedia.org/wiki/Big_data

[13] Carlo Vaccari, PhD Thesis: "Big Data in Official Statistics", University of Camerino , 2014

[14] Eric Hatcher, Otis Gospodnetic, Michael McCandless, "Lucene in Action", Manning, 2010

[15] Trey Grainger, Timothy Potter, "Solr in Action", Manning, 2014

[16] Thilina Gunarathne , "Hadoop MapReduce v2 Cookbook", Packt, 2015

[17] https://mavatar.com/

[18] https://mahout.apache.org/

[19] Piero Giacomelli, "Apache Mahout Cookbook", Packt, 2013

[20] Grant S. Ingersoll, Thomas S. Morton, Andrew L. Farris "Taming Text", Manning, 2013