ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ , ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ

Αριθμητική Αξιολόγηση Αλγορίθμων Κατάταξης Συναισθήματος , με Χρήση Γράφων Ν-Γραμμάτων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Γεώργιος Ι. Κανέλλης

Επιβλέπουσα : Θεοδώρα Βαρβαρίγου
Καθηγήτρια

Αθήνα, Νοέμβριος 2014

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ , ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Αριθμητική Αξιολόγηση Αλγορίθμων Κατάταξης Συναισθήματος , με Χρήση Γράφων Ν-Γραμμάτων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Γεώργιος Ι. Κανέλλης

Επιβλέπουσα : Θεοδώρα Βαρβαρίγου
Καθηγήτρια

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την ....η Νοεμβρίου 2014.

.....	.....	.....
Θ. Βαρβαρίγου	Β. Λούμος	Ε. Καγιάφας
Καθηγήτρια	Καθηγητής	Καθηγητής

Αθήνα, Νοέμβριος 2014

.....

Γεώργιος Ι. Κανέλλης

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

Περίληψη

Στην παρούσα διπλωματική εργασία λαμβάνει χώρα η επεξεργασία εκατομμυρίων ανεξάρτητων και περιορισμένου μήκους κειμένων του κοινωνικού δικτύου Twitter, ώστε να αποτελέσουν κατάλληλη είσοδο για αλγορίθμους ανάλυσης και κατάταξης συναισθημάτων στις τρεις βασικές κατηγορίες (θετικό, αρνητικό, ουδέτερο). Ο μηχανισμός ταξινόμησης υλοποιείται με το Weka, μία δημοφιλή βιβλιοθήκη γραμμένη σε Java, βιβλιοθήκη άκρως κατάλληλη για ταξινόμηση/πρόβλεψη. Οι αλγόριθμοι μηχανικής μάθησης του εργαλείου Weka, δρουν σε αριθμητικές τιμές , τις οποίες παρέχουν από τα αρχικά κείμενα οι μετρικές που προκύπτουν από την δημιουργία γράφων ν-γραμμάτων. Οι γράφοι ν-γραμμάτων παράγονται από τα κατάλληλα προσαρμοσμένα κείμενα. Περαιτέρω, οι αλγόριθμοι μηχανικής μάθησης εξετάζονται ως προς την απόδοσή τους στις προκαθορισμένες τιμές , και αξιολογούνται.

Λέξεις Κλειδιά:

Εξόρυξη Δεδομένων , Μηχανική Μάθηση , Ανάλυση Συναισθήματος , Δομικό Μοτίβο , Ταξινόμηση , Πρόβλεψη , Instance , Χαρακτηριστικό , Κλάση , Δέντρο , Κανόνας , Κοινωνικό Δίκτυο , Twitter , Γράφος Ν-Γραμμάτων , Weka .

Η σελίδα είναι σκόπιμα κενή.

Abstract

This thesis accomodates the processing of millions of independent and short messages of the Twitter Social Network, in order for them to be an appropriate input for sentiment analysis and classification algorithms to the three basic sentiment categories (positive, negative, neutral).The classification mechanism was implemented with Weka, a popular machine learning library written in Java, a library most suitable for classification/prediction. The machine learning algorithms of Weka act on arithmetic values, which are provided by metrics generated by n-gram graphs. The n-gram graphs are created based on the filtered input. Furthermore,the algorithms are being tested for their efficiency, set at their default values.

Λέξεις Κλειδιά:

Data Mining , Machine Learning , Sentiment Analysis , Structural Pattern , Classification , Prediction , Instance , Attribute , Class , Tree , Rule , Social Network , Twitter , N-Gram Graph , Weka .

Η σελίδα είναι σκόπιμα κενή.

Περιεχόμενα

Κατάλογος σχημάτων

Πηγή των σχημάτων αποτελεί το βιβλίο [1] και η δημοσίευση [1].

1 Εισαγωγή

1.1 Ανεπεξέργαστα Δεδομένα και Πληροφορία

Η σύγκλιση της πληροφορικής και των επικοινωνιών έχει δημιουργήσει μια κοινωνία που βασίζεται στις πληροφορίες. Ωστόσο, οι περισσότερες από τις πληροφορίες είναι σε ακατέργαστη μορφή: τα λεγόμενα ανεπεξέργαστα δεδομένα (raw data) . Όμως κατακλυζόμαστε από δεδομένα,των οποίων η ποσότητα συνεχώς αυξάνει. Η φθηνή επέκταση του χώρου αποθήκευσης αναβάλει συχνά κάθε απόφαση διαχείρισης των αποθηκευμένων δεδομένων. Και τα δεδομένα αυξάνονται επιπλέον λόγω του ενδιαφέροντος που τρέφουν για τις καταναλωτικές συμπεριφορές το εμπόριο και η βιομηχανία. Είναι πια σαφές ότι η παραγωγή νέων δεδομένων και η κατανόησή τους απομακρύνονται στην πράξη ανησυχητικά.

Αν χαρακτηριστούν τα data ως καταγεγραμμένα γεγονότα, τότε οι πληροφορίες είναι το σύνολο των μοτίβων , ή των προσδοκιών , που κρύβονται μέσα στα δεδομένα. Υπάρχει τεράστιος όγκος πληροφορίας κλειδωμένος στις βάσεις δεδομένων. Πληροφορίας που ενδέχεται να είναι σημαντική, αλλά ακόμη δεν έχει ανακαλυφθεί ή συντεθεί. Η αποστολή του πεδίου της εξόρυξης δεδομένων (data mining) είναι να την φέρει στο προσκήνιο.

1.2 Εξόρυξη Δεδομένων : Στόχοι (Μοτίβα) και Προβλήματα

Η εξόρυξη δεδομένων (Data Mining) είναι η εξαγωγή της σιωπηρής, προηγουμένως άγνωστης , και δυνητικά χρήσιμης πληροφορίας από τα δεδομένα. Η ιδέα είναι να υλοποιηθούν προγράμματα στους υπολογιστές που κοσκινίζουν αυτόματα τις βάσεις δεδομένων, ψάχνοντας για κανονικότητες ή μοτίβα. Η διαδικασία αυτή γίνεται με αυτόματο, ή πιο σύνηθες, με ημιαυτόματο τρόπο.

Αν βρεθούν ισχυρά μοτίβα, κατά πάσα πιθανότητα θα γενικευτούν για να γίνονται πιο ακριβείς προβλέψεις σε μελλοντικά δεδομένα. Φυσικά υπάρχουν προβλήματα. Πολλά μοτίβα είναι αδιάφορα. Άλλα είναι εικονικά, δηλαδή εξαρτώνται από τυχαίες συμπτώσεις στο συγκεκριμένο σύνολο δεδομένων (dataset) που χρησιμοποιήθηκε. Επίσης τα πραγματικά δεδομένα δεν είναι ιδανικά: μερικά κομμάτια είναι αλλοιωμένα ή λείπουν. Οτιδήποτε ανακαλύπτεται είναι ανακριβές: Υπάρχουν εξαιρέσεις σε κάθε κανόνα, και περιπτώσεις που δεν καλύπτονται από κανέναν κανόνα. Οι αλγόριθμοι πρέπει να είναι αρκετά ισχυροί ώστε να μπορούν να αντεπεξέλθουν με τα ατελή δεδομένα, και να εξάγουν κανονικότητες που είναι μεν ανακριβείς, αλλά χρήσιμες.

1.3 Μηχανική Μάθηση

Η μηχανική μάθηση (machine learning) παρέχει την τεχνική βάση του data mining. Είναι ένας κλάδος της τεχνητής νοημοσύνης και αφορά την κατασκευή και μελέτη συστημάτων τα οποία μπορούν να "μαθαίνουν" από τα δεδομένα (data instances). Με τη μηχανική μάθηση καθίσταται εφικτή η κατασκευή προσαρμόσιμων προγραμμάτων υπολογιστών τα οποία λειτουργούν με βάση την αυτοματοποιημένη ανάλυση συνόλων δεδομένων και όχι τη διαίσθηση των μηχανικών που τα προγραμμάτισαν. Η μηχανική μάθηση επικαλύπτεται σημαντικά με τη στατιστική, αφού και τα δύο πεδία μελετούν την ανάλυση δεδομένων.

Χρησιμοποιείται για να εξάγει πληροφορία από τα ακατέργαστα δεδομένα των βάσεων δεδομένων, πληροφορία που εκφράζεται σε κατανοητή μορφή και μπορεί να χρησιμοποιηθεί για διάφορους σκοπούς. Η διαδικασία είναι αφαιρετική: Την λήψη δεδομένων ακολουθεί η εύρεση της κρυφής δομής. Αποτελείται από τεχνικές που χρησιμοποιούνται σε πρακτικές εφαρμογές data mining, προς εύρεση και περιγραφή των διαρθρωτικών μοτίβων των δεδομένων.

Μερικές εφαρμογές data mining επικεντρώνονται στην πρόβλεψη: Προβλέπουν τι θα συμβεί σε νέες καταστάσεις κοιτώντας δεδομένα που περιγράφουν τι συνέβη στο παρελθόν, συχνά μαντεύοντας την ταξινόμηση νέων παραδειγμάτων. Είναι όμως εξίσου, ίσως και περισσότερο, ενδιαφέρουσες οι εφαρμογές των οποίων το αποτέλεσμα της μάθησης είναι μια πραγματική περιγραφή μιας δομής που μπορεί να χρησιμοποιηθεί για την ταξινόμηση παραδειγμάτων. Αυτή η δομική περιγραφή υποστηρίζει την επεξήγηση και την κατανόηση, καθώς και την πρόβλεψη. Σε αυτή την περιγραφή συμβάλουν οι γνώσεις των χρηστών, οι οποίες αποτελούν το κύριο ενδιαφέρον των πρακτικών εφαρμογών μηχανικής μάθησης. Αυτό αποτελεί ένα από τα μεγαλύτερα πλεονεκτήματα της μηχανικής μάθησης έναντι του κλασσικού στατιστικού μοντέλου.

Ως συνέπεια, η έννοια της μάθησης όταν εκτελούμε πρόβλεψη ταιριάζει με την έννοια της απόδοσης. Μπορεί κανείς να ελέγξει την μάθηση παρατηρώντας την τωρινή συμπεριφορά και συγκρίνοντάς την με την παρελθοντική. Έτσι η μάθηση μπορεί να αντικατασταθεί ως όρος με την προπόνηση, λόγω της ανάγκης ύπαρξης παρελθοντικής συμπεριφοράς.

1.4 Είδη Αλγορίθμων Ταξινόμησης

Οι αλγόριθμοι μηχανικής μάθησης κατηγοριοποιούνται ανάλογα με το επιθυμητό αποτέλεσμα του αλγορίθμου. Οι συνηθέστερες κατηγορίες είναι η επιτηρούμενη , η μη επιτηρούμενη και η ενισχυτική μάθηση.

Η επιτηρούμενη μάθηση, επιβλεπόμενη μάθηση ή μάθηση με επίβλεψη (supervised learning), θέλει τον

αλγόριθμο να κατασκευάζει μια συνάρτηση που απεικονίζει δεδομένες εισόδους (training set)(σύνολο εκπαίδευσης) σε γνωστές, επιθυμητές εξόδους , με απώτερο στόχο τη γενίκευση της συνάρτησης αυτής και για εισόδους (testing set)(σύνολο ελέγχου) με άγνωστη έξοδο . Οι συνθήκες υπό τις οποίες ένας machine learning αλγόριθμος αποδίδει σε υψηλό βαθμό στο να απεικονίζει ορθώς εισόδους δίχως δοσμένη (στον αλγόριθμο) έξοδο, στη σωστή έξοδο, αποτελεί βασικό αντικείμενο μελέτης της Υπολογιστικής Θεωρίας Μάθησης (computational learning theory).

Η μη επιτηρούμενη μάθηση, ανεπίβλεπτη μάθηση ή μάθηση χωρίς επίβλεψη (unsupervised learning), θέλει τον αλγόριθμο να κατασκευάζει ένα μοντέλο για κάποιο σύνολο εισόδων χωρίς να γνωρίζει επιθυμητές εξόδους για το σύνολο εκπαίδευσης. Στην ενισχυτική μάθηση (reinforcement learning), ο αλγόριθμος μαθαίνει μια στρατηγική ενεργειών για μια δεδομένη παρατήρηση.

Η ανάλυση των αλγόριθμων μηχανικής μάθησης είναι ένας κλάδος της στατιστικής που ονομάζεται θεωρία μάθησης. Απλοϊκά παραδείγματα μηχανικής μάθησης αποτελεί η προσπάθεια ενός αλγορίθμου να κατατάξει νέα emails σε spams ή μη, ενώ προηγουμένως εκπαιδεύτηκε πάνω σε κατάλληλο training set, καθώς και η οπτική αναγνώριση χαρακτήρων, κατά την οποία εκτυπωμένοι χαρακτήρες αναγνωρίζονται αυτόματα βάσει προγενέστερων παραδειγμάτων.

1.5 Δομικά Μοτίβα

Τα μοτίβα οφείλουν να μην οδηγούν σε τετριμμένες προβλέψεις επί των δεδομένων. Επίσης διαχωρίζονται στα διαφανή(ο τρόπος παρουσίασής τους μαρτυρά την εσωτερική δομή τους) και τα αδιαφανή. Υποθέτοντας ότι και οι δύο κατηγορίες είναι αποδεκτές, το ουσιαστικό ζητούμενο για ένα μοτίβο που εξωρύχθη είναι το κατά πόσο είναι δομημένο έτσι ώστε να μπορεί να εξεταστεί , να περιγραφεί λογικά, και να χρησιμοποιηθεί για μελλοντικές αποφάσεις. Αυτά είναι τα λεγόμενα διαρθρωτικά μοτίβα (structural patterns) , τα οποία προσφέρουν συγκεκριμένο τρόπο περιγραφής κάποιας ιδιότητας των δεδομένων.

1.6 Περιγραφή Δομικών Μοτίβων και Κανόνες

Τα διαρθρωτικά μοτίβα μπορούν να περιγραφούν από κανόνες (συνθήκη και συνέπεια) , είτε με τον δημοφιλή τρόπο αναπαράστασης μέσω δέντρων αποφάσεων, τα οποία καθορίζουν την ακολουθία αποφάσεων που οδηγούν σε συγκεκριμένη συνέπεια,είτε με άλλους τρόπους. Οι κανόνες δεν πηγάζουν από τη γενίκευση των δεδομένων, αλλά τα συνοψίζουν. Στις περισσότερες περιπτώσεις η είσοδος δεν είναι πλήρης, και συνεπώς οι κανόνες πρέπει να μπορούν να γενικεύουν τα δεδομένα προσδίδοντάς τους επεκτάσεις. Δηλαδή μπορούν να προκύψουν κανόνες από κάποια data-παραδείγματα , οι οποίοι δεν

χρησιμοποιούν στην συνθήκη τους πλήρως την είσοδο, άρα είσοδοι δίχως δεδομένα στο κομμάτι που παραλείπει ο κανόνας μπορούν να εκφραστούν από τον κανόνα, και ας είναι στο σύνολό τους ελλιπείς.

Οι κανόνες βέβαια κάνουν και λάθος συνεπαγωγές, λόγω των προβλημάτων του data mining που είδαμε παραπάνω (π.χ. λόγω θορύβου(noise) στα δεδομένα εισόδου).

Οι κανόνες έχει σημασία να ερμηνεύονται με τη σωστή σειρά. Όταν κάποιος δεν εφαρμόζεται, ακολουθεί ο επόμενος , κ.ό.κ. . Έτσι δημιουργείται μια λίστα αποφάσεων. Κανόνες που εκτελούνται ανεξάρτητα , αναγκαστικά περιγράφουν με λάθος τρόπο τα δεδομένα εισόδου. Όταν η λίστα αποφάσεων έχει κτιστεί έτσι, ώστε όλοι οι κανόνες της να συνεπάγονται μια τιμή για ένα τμήμα της εισόδου, τότε λέγονται κανόνες ταξινόμησης(πρόβλεψης). Όταν οι συνεπαγωγές δίνουν τιμές για διάφορα τμήματα της εισόδου, τότε συσχετίζουν αυτά τα τμήματα εισόδου, και οι αντίστοιχοι κανόνες λέγονται κανόνες συσχέτισης.

Δεν είναι όμως όλοι οι κανόνες πλήρεις και ντετερμινιστικοί, δηλαδή δεν δίνουν μοναδικό συμπέρασμα για κάθε νέα είσοδο. Συχνά υπάρχουν περιπτώσεις που κανένας κανόνας δεν εφαρμόζεται , ενώ σε άλλες πολλοί, με παραπάνω της μίας προτεινόμενης εξόδου. Τότε γίνεται χρήση πιθανοτήτων ή βαρών, για να επικρατήσει η έξοδος του πιο αξιόπιστου κανόνα.

Έχοντας επίγνωση του τρόπου περιγραφής των διαρθρωτικών , δηλαδή των χρήσιμων, μοτίβων , συνειδητοποιούμε ότι η περιγραφή της δομής των δεδομένων εισόδου, μέσω μοτίβων που ανακαλύπτει το data mining σε αυτά, υποδηλώνει ότι τέτοια μοτίβα παρέχουν γνώση. Επειδή η κανόνες που αναπαριστούν τα μοτίβα γίνονται εύκολα κατανοητοί από τον ανθρώπινο αναγνώστη, η γνώση αυτή, δηλαδή τα χρήσιμα μοτίβα, είναι εξίσου χρήσιμα με την δυνατότητα πρόβλεψης επί των δεδομένων.

1.7 Εφαρμογές Μηχανικής Μάθησης

1.7.1 Web Mining

Η μηχανική μάθηση είναι ένα ισχυρό εργαλείο, και συνεπώς εφαρμόζεται σε ποικίλους τομείς. Παραδείγματος χάριν, για Web Mining. Εταιρείες μηχανών αναζήτησης εξετάζουν συνδέσμους του διαδικτύου για να μετρήσουν το κύρος των ιστοσελίδων. Οι μετρικές της PageRank υπηρεσίας αξιολογούν την “ισχύ” μιας ιστοσελίδας στο διαδίκτυο μετρώντας τους συνδέσμους που δείχνουν αυτή την σελίδα, δίνοντας σημασία και στην ισχύ της σελίδας προορισμού του συνδέσμου. Το PageRank χρησιμοποιείται από τις μηχανές αναζήτησης για την ταξινόμηση των αποτελεσμάτων αναζήτησης πριν την εμφάνισή τους.

Εναλλακτικά, ένα σύνολο αναζητήσεων αξιολογείται από ανθρώπους και έπειτα ένας αλγόριθμος μάθησης προπονείται με τις αναζητήσεις και τις αξιολογήσεις τους.

Τα παραπάνω υποδηλώνουν το αυτονόητο: οι μηχανές αναζήτησης εξορύσσουν το περιεχόμενο του διαδικτύου και των αναζητήσεων των χρηστών. Βασικός τους σκοπός είναι το κέρδος από τις διαφημίσεις. Με την εξόρυξη και τα μοτίβα που ανακαλύπτουν, προβλέπουν ποια διαφήμιση θα αρέσει σε κάποιον διαδικτυακό χρήστη. Και επιθυμούν σωστές προβλέψεις, καθώς πληρώνονται από τους διαφημιστές μόνο εάν επιλεχθεί από κάποιον χρήστη η διαφήμισή τους προς ανάγνωση. Και όχι μόνο οι μηχανές αναζήτησης. Συμμετέχουν και ιστοσελίδες για τον κινηματογράφο, διαδικτυακά βιβλιοπωλεία , κ.ά. .

Προς τους παραπάνω σκοπούς βοηθούν και οι χρήστες, συχνά άθελά τους ή από αδιαφορία. Οι χρήστες μοιράζονται δημοσίως κάθε σκέψη , εμπειρία , τοποθεσία , εικόνα , βίντεο ή προτίμηση , μέσω των υπηρεσιών κοινωνικής δικτύωσης , blogs και forums.

1.7.2 Αποφάσεις που Προϋποθέτουν Κρίση

Σε περιπτώσεις που η λήψη αποφάσεων προϋποθέτει κάποιου είδους κρίση, υπάρχουν κάποια στάδια προσέγγισης. Παίρνοντας το παράδειγμα μιας εταιρείας δανειοδοσίας, παλαιότερα συνήθως , με αυτοματοποιημένο τρόπο, εφαρμόζετο μια στατιστική διαδικασία αποφάσεως, που αποφαινόταν για την πλειοψηφία των δανειοληπτών , ενώ οι εξαιρέσεις μελετούντο από κάποιον άνθρωπο. Σημειώνοντας με την πάροδο του χρόνου τα στοιχεία των δανειοληπτών και εάν αποπλήρωσαν τα δάνεια ή όχι, φτιάχτηκαν training είσοδοι, πάνω στις οποίες μπορεί ένας αλγόριθμος να προπονηθεί και να αξιολογήσει τους μελλοντικούς υποψήφιους δανειολήπτες. Προφανώς η μηχανική μάθηση σε τέτοιες περιπτώσεις ήδη εφαρμόζεται και προσφέρει χρήσιμα συμπεράσματα.

1.7.3 Διαρροές Πετρελαίου

Με τη χρήση δορυφόρων και μέσω των εικόνων που αυτοί παράγουν, οι επιστήμονες προσπαθούν να προβλέψουν μια οικολογική καταστροφή από διαρροή πετρελαίου ή παράνομες εξορύξεις. Οι εικόνες αυτές όμως εμπεριέχουν θόρυβο, λόγω των περιβαλλοντολογικών φαινομένων. Η δια ζώσης μελέτη του πιθανού επικίνδυνου σημείου που εμπεριέχεται σε μια εικόνα είναι μια ακριβή διαδικασία και απαιτεί εξειδικευμένο προσωπικό. Προφανώς εικόνες ήδη ελεγμένες στο παρελθόν μπορούν να προπονήσουν έναν αλγόριθμο μηχανικής μάθησης για την μελέτη νέων εικόνων άνευ σοβαρού κόστους, με υπαρκτή μια διαχειρίσιμη παράμετρο που ορίζει την έμφαση του αλγορίθμου στο να αποφεύγει λάθος επισημάνσεις ή αντίθετα στο να μην αποτυγχάνει να εντοπίσει ένα πρόβλημα. Απαραίτητη προϋπόθεση είναι η συμπερίληψη στην είσοδο του αλγορίθμου μιας πλήρους συλλογής πληροφοριών για την υπό εξέταση περιοχή.

1.7.4 Πρόβλεψη Ηλεκτρικού Φορτίου

Στην βιομηχανία παραγωγής ηλεκτρικής ενέργειας, είναι σημαντικό να μπορεί να αποφασιστεί η μελλοντική ζήτηση για ισχύ όσο πιο νωρίς γίνεται. Εάν μπορούν να επιτευχθούν ακριβείς προβλέψεις για το μέγιστο και ελάχιστο φορτίο για κάθε ώρα , μέρα , μήνα , εποχή και χρόνο, εξοικονομούνται χρήματα στην διαχείριση των αποθεματικών ενέργειας, στον σχεδιασμό της συντήρησης και στη διαχείριση των καυσίμων. Έχοντας ένας αλγόριθμος μηχανικής μάθησης στη διάθεσή του τα φορτία που απαίτησαν οι καταναλωτές σε βάθος πολλών ετών, την κατανομή τους μέσα στη μέρα ,βδομάδα κτλ., τις περιβαλλοντολογικές συνθήκες , τις ημέρες εορτών, και ένα μικρό πεπερασμένο σύνολο των πιο όμοιων παρελθοντικών ημερών με την χθεσινή ημέρα, μπορεί να προβλέψει την ημερήσια κατανομή φορτίου σε δευτερόλεπτα, έναντι των ωρών που χρειάζεται ο ανθρώπινος παράγοντας.

1.7.5 Διαγνώσεις

Η προληπτική συντήρηση ηλεκτρομηχανικών συσκευών ,όπως κινητήρες και γεννήτριες, μπορεί να προλάβει αστοχίες που διακόπτουν ή καθυστερούν τις βιομηχανικές διαδικασίες. Η συντήρηση γίνεται από ειδικούς ,όταν την αναλαμβάνει ο ανθρώπινος παράγοντας, και η μελέτη των μετρήσεων προς εύρεση αστοχιών και του είδους των αστοχιών είναι μια επίπονη διαδικασία,η οποία πρέπει να επαναληφθεί για κάθε είδος συσκευής. Η εφαρμογή μηχανικής μάθησης σε αυτήν την περίπτωση χρειάζεται είσοδο μόνο με περιπτώσεις λαθών , καθώς αναλαμβάνει την εύρεση του είδους του λάθους. Οι κανόνες αυτοί που θα προκύψουν θα πρέπει να συσχετιστούν με αυτούς του ειδικού,καθώς και με τις γνώσεις του,ώστε να επαληθευθεί η εγκυρότητά τους.

1.7.6 Marketing και Πωλήσεις

Είδαμε ήδη την ανάγκη των διαφημιστών για τη μηχανική μάθηση. Μπορούμε να προεκτείνουμε την εφαρμογή της και στο πεδίο των τραπεζών, με στόχο τώρα όχι την μελέτη των δανειοληπτών, αλλά την μελέτη των κατόχων λογαριασμών, ώστε να μπορεί να προβλεφθεί η μελλοντική κατάσταση του κεφαλαίου της τράπεζας. Παρόμοια εταιρείες κινητής τηλεφωνίας προωθούν στοχευμένες διαφημίσεις των υπηρεσιών τους σε συγκεκριμένους πελάτες, ώστε με ελάχιστο κόστος να διατηρήσουν την παλιά πελατεία τους.

Επιπλέον κανόνες συσχέτισης μπορούν να παραχθούν από τα δεδομένα στα ταμεία των σουπερμάρκετ, ώστε η κάθε ατομική αγορά διαφόρων προϊόντων να συμβάλει στην παραγωγή συμπερασμάτων για το ποια προϊόντα συνηθίζεται να αγοράζονται μαζί. Τα δεδομένα στην έξοδο των ταμείων αποτελούν για τους μεταπωλητές των προϊόντων μία από τις λίγες πηγές δεδομένων. Τα συμπεράσματα που προκύ-

πτουν χρησιμεύουν στα μαγαζιά πώλησης για την “έξυπνη” τοποθέτηση των προϊόντων(όπου προϊόντα που συνήθως αγοράζονται μαζί τοποθετούνται κοντά), τον περιορισμό προσφορών σε παραπάνω του ενός προϊόντος που ανήκουν στην ίδια ομάδα συσχετισμένων προϊόντων, την προσφορά κουπονιών για προϊόν που δεν αγοράστηκε αλλά ανήκει στην ίδια ομάδα με αυτό που αγοράστηκε.

Η αξία του να γνωρίζει το σουπερμάρκετ τις καταναλωτικές συνήθειες του πελάτη είναι πολύ μεγαλύτερη της έκπτωσης που θα του προσφέρει για κάποια προϊόντα. Εξού και η κυκλοφορία καρτών πίστωσης πόντων, που χρησιμεύουν όχι μόνο στην προμήθεια κυρίως των δημοφιλών προϊόντων ,αλλά και στη μείωση του κόστους αποστολής διαφημιστικών φυλλαδίων.

Μια τάση των σουπερμάρκετ είναι να δίνουν την αίσθηση στον καταναλωτή , ότι σε έναν κόσμο με αλόγιστα αυξανόμενες τιμές, ο καταναλωτής βγαίνει κερδισμένος με το να αγοράσει με σχετική έκπτωση προϊόντα που κανονικά δεν θα αγόραζε.

Τέλος , το προσανατολισμένο marketing στοχεύει αρχικά μέσω φυλλαδίων σε δημογραφικές περιοχές με πολλούς πιθανούς μελλοντικούς πελάτες, και αν κάποιοι τηλεφωνήσουν για κάποια από τις προσφορές σημειώνονται,ώστε σε επόμενες προσφορές βάζοντας τους σημειωμένους ενδιαφερόμενους ως είσοδο του αλγορίθμου μηχανικής μάθησης να βρίσκεται ανέξοδα ποιοι από αυτούς είναι πιθανόν να αγοράσουν κάποιο προϊόν ή υπηρεσία.

Οι εφαρμογές της μηχανικής μάθησης είναι πάρα πολλές. Μερικά παραδείγματα ακόμη είναι οι τομείς της Βιολογίας, της Χημείας, της Φαρμακοβιομηχανίας, της Αστρονομίας, οι υπηρεσίες εξυπηρέτησης πελατών, η ιατρική περίθαλψη ζωντανού χρόνου, τα λογισμικά προστασίας υπολογιστών ή δικτύων ζωντανού χρόνου.

1.8 Μηχανική Μάθηση και Στατιστική

Η μηχανική μάθηση και η στατιστική επικαλύπτονται σε τέτοιο βαθμό που δε μπορεί να βρεθεί μια διαχωριστική γραμμή. Και τα δύο αφορούν τεχνικές ανάλυσης δεδομένων. Κάποιες προέκυψαν από κλασσικές στατιστικές διεργασίες και κάποιες από τον τρόπο που εφαρμόστηκε η μηχανική μάθηση στους ηλεκτρονικούς υπολογιστές.

Ένας ανυπόστατος και ρηχός διαχωρισμός είναι η παρουσίαση της στατιστικής να ασχολείται με τον έλεγχο υποθέσεων και της μηχανικής μάθησης ως τη διαδικασία της γενίκευσης , ούσα η αναζήτηση διαμέσου υποθέσεων (μοτίβων δηλαδή). Παρόλα αυτά, η στατιστική είναι πολλά περισσότερα από απλές υποθέσεις, ενώ υπάρχουν τεχνικές μηχανικής μάθησης που δεν πραγματοποιούν αναζητήσεις.

Είναι πολλές οι κοινές τους μέθοδοι. Από τις σημαντικότερες είναι η ταξινόμηση μέσω της επαγωγής δέντρων αποφάσεων (δέντρων παλινδρόμησης). Εξίσου σημαντική είναι και η χρήση μεθόδων

“κοντινότερης-γειτνίασης” για ταξινόμηση. Αυτές οι μέθοδοι αποτελούν κλασσικές στατιστικές τεχνικές , οι οποίες χρησιμοποιούνται από ερευνητές της μηχανικής μάθησης για να βελτιώσουν την απόδοση της ταξινόμησης , αλλά και τον υπολογιστικό της χρόνο.

Η μηχανική μάθηση ,ακόμα και από την αρχή, χρησιμοποιεί στατιστικές μεθόδους για να διαμορφώσει σε μια κατάλληλη μορφή την αρχική είσοδο, ώστε να χρησιμοποιηθεί σωστά από τον αλγόριθμο: απεικόνιση δεδομένων, επιλογή μερικών εκ των χαρακτηριστικών (κομματιών) της εισόδου, απόρριψη ακραίων τιμών, κ.ά. Αλλά και κατά την παραγωγή των κανόνων ή δένδρων, δηλαδή του ταξινομητή, χρησιμοποιούνται στατιστικοί έλεγχοι για την διόρθωση των ταξινομητών όταν είναι “overfitted” , δηλαδή όταν εξαρτώνται σε πολύ μεγάλο και ανεπιθύμητο βαθμό από λεπτομέρειες των δεδομένων της εισόδου, λεπτομέρειες που ίσως να μην εμφανίζονται σε μελλοντική είσοδο προς ταξινόμηση.

Τέλος, στατιστικές μέθοδοι χρησιμοποιούνται για να επικυρώνουν τη παρούσα δομή ενός ταξινομητή και να επαληθεύουν τα αποτελέσματα των αλγορίθμων μηχανικής μάθησης.

1.9 Εξόρυξη Δεδομένων και Ηθική

Οποιοσδήποτε πειραματίζεται με μεθόδους μηχανικής μάθησης σε δεδομένα που αποτελούν προσωπικά στοιχεία ανθρώπων , ή πιο βολικά πολιτών, οφείλει να δρα υπεύθυνα και μέσα στα όρια του νόμου. Υπεύθυνα συνειδητοποιώντας τα ηθικά ζητήματα που εγείρονται, και με νομιμότητα παράγοντας κανόνες που δεν διαχωρίζουν τους ανθρώπους με σεξιστικό , θρησκευτικό ή άλλον τρόπο. Το τελευταίο όμως είναι σύνθετο, άρα επαφίεται αποκλειστικά στο είδος της εφαρμογής της μηχανικής μάθησης. Τέτοιοι διαχωρισμοί είναι προφανείς για ιατρικές διαγνώσεις, αλλά απαράδεκτοι για επιλογή δανειοληπτών. Το ζήτημα αυτό γίνεται ακόμη πιο σύνθετο, καθώς κάποιες νόμιμες αναγνώσεις των δεδομένων (παραδείγματος χάριν ταχυδρομικός κώδικας) ταυτίζεται τοπικά σε πολυφυλετικές χώρες εκατομμυρίων κατοίκων με την φυλή των κατοίκων του συγκεκριμένου ταχυδρομικού κώδικα, ακόμα και αν αυτή ποτέ δεν αναζητήθηκε.

1.9.1 Re-identification

Η δουλειά που έχει λάβει χώρα στο πεδίο του reidentification ,δηλαδή της εξάλειψης (αντιστροφής) της ανωνυμίας, έχει κάνει ξεκάθαρο το μέγεθος της δυσκολίας της εφαρμογής ανωνυμίας στα δεδομένα. Εταιρείες ή κρατικοί οργανισμοί που σε παρελθοντικό χρόνο προσπάθησαν να ανωνυμοποιήσουν δεδομένα πριν τα δημοσιεύσουν απέτυχαν, καθώς κάποιοι εφάρμοσαν reidentification και του εξέθεσαν. Το δυστυχές είναι ότι εάν πραγματικά αφαιρέσει κανείς όλα τα προσωπικά δεδομένα που απαιτείται ώστε να αποτυγχάνει το reidentification, τότε δε θα μπορεί να παραχθεί κάποιο χρήσιμο μοτίβο από τα

δεδομένα αυτά.

1.9.2 Χρήση Προσωπικών Δεδομένων

Είναι ευρέως αποδεκτό, ότι προτού κανείς λάβει την απόφαση να προσφέρει προσωπικές πληροφορίες, πρέπει να γνωρίζουν πού θα χρησιμοποιηθούν και για ποίο λόγο, ποία στάδια θα ακολουθηθούν ώστε να προστατευθεί η εμπιστευτικότητα των δεδομένων του και η αξιοπρέπεια του ατόμου. Πρέπει να γνωρίζει ποίες είναι οι συνέπειες του να παράσχει ή να του κρατώνται τα δεδομένα, και όποια ενημέρωση λάβει επί των παραπάνω θεμάτων να είναι σαφής και όχι σε δυσνόητο νομικό κείμενο.

Αποσαφηνισμένο οφείλει να είναι και το βελησνεκές της παραγωγής μοτίβων. Δεν πρέπει να συμπεραίνονται δομές των στοιχείων του ατόμου, όταν δεν έχει συμφωνήσει για αυτές. Προφανώς η σαφής οριοθέτηση της παραγωγής μοτίβων είναι δύσκολη, και δύσκολη είναι και η τήρησή της από τον αλγόριθμο μηχανικής μάθησης.

Παράξενα συμπεράσματα μπορούν να προκύψουν με το data mining. Αναδύονται στην επιφάνεια μέσω του data mining στερεότυπα της καθημερινότητας που ούτως ή άλλως έχουν εντοπιστεί και χρησιμοποιούνται από εταιρείες (βλέπε ασφαλιστικές). Την ηθικότητα στη χρήση στερεοτύπων και την χρησιμότητα αυτών , ορίζει αναγκαστικά η ανθρώπινη κρίση.

Κατέχοντας προσωπικά δεδομένα, όποιος εξασκεί μηχανική μάθηση οφείλει να ορίσει με σαφήνεια ποίος μπορεί να έχει πρόσβαση σε αυτά. Υπάρχουν τακτικές προστασίας των δεδομένων οι οποίες διατηρούνται με την πάροδο του χρόνου, δίχως να είναι σαφές με την πρώτη ματιά η αιτία. Παραδείγματος χάριν, οι βιβλιοθήκες ποτέ δεν φανερώνουν ποίος δανείστηκε ένα δανεισμένο βιβλίο, ώστε να μην δεχτεί το άτομο πίεση να επιστρέψει το βιβλίο ή κατάκριση για τα αναγνωστικό του ενδιαφέρον. Όσοι δε προωθούν την αποδόμηση των φυσικών βιβλιοθηκών, σίγουρα θα ωφεληθούν από την καταγραφή των αναγνωστικών συνηθειών των αναγνωστών, και ίσως παρανόμως φροντίσουν να έχουν κέρδος πωλώντας αυτές τις πληροφορίες σε εκδότες!

1.10 Ευρύτερα Ζητήματα

Γίνεται σαφές από την ανάδυση αλλόκοτων συμπερασμάτων (π.χ. όσοι έχουν κόκκινο αμάξι δεν αποπληρώνουν τα δάνειά τους) ότι η εξόρυξη δεδομένων είναι ένα εργαλείο, και οι άνθρωποι που αποφασίζουν πρέπει να αξιολογήσουν ότι προέκυψε από το data mining, και να χρησιμοποιήσουν όπου μπορούν και άλλες γνώσεις.

Επίσης τα μοτίβα που προκύπτουν από τη μηχανική μάθηση δεν συνεπάγονται αυτόματα ότι σε αυτά βρίσκεται οφθαλμοφανής η βέλτιστη απόφαση. Ο ανθρώπινος παράγοντας πρέπει πάλι να δράσει. Στην

περίπτωση που είδαμε παραπάνω στα σουπερμάρκετ με τις ομάδες προϊόντων που συνήθως πωλούνται μαζί , ίσως είναι πιο κερδοφόρο για τον πωλητή να απομακρύνει χωρικά τα προϊόντα της ίδιας ομάδας ώστε στην αναζήτησή τους ο καταναλωτής να παρασυρθεί να ψωνίσει επιπλέον προϊόντα. Ή εναλλακτικά ίσως συμφέρει να τοποθετηθεί,για να εξαντληθεί ως απόθεμα,ένα κακής ποιότητας ή απλώς πιο ακριβό προϊόν κοντά με τα υπόλοιπα της ομάδας , αντί για ένα πιο συμφέρον για τον καταναλωτή όμοιό του.

2 Η Είσοδος

Η είσοδος ενός αλγορίθμου μηχανικής μάθησης παίρνει τη μορφή "εννοιών","στιγμιοτύπων" και "χαρακτηριστικών". Ονομάζουμε αυτό που πρέπει να μαθευτεί "έννοια", ενώ το αποτέλεσμα της μάθησης "περιγραφή μιας έννοιας". Η ιδέα της έννοιας είναι το αποτέλεσμα της διαδικασίας της μάθησης, και αυτό πρέπει να είναι κατανοητό, δηλαδή να μπορεί να μελετηθεί, να συζητηθεί και να αμφισβητηθεί. Πρέπει όμως να είναι και λειτουργικό, δηλαδή να μπορεί να εφαρμοστεί σε αληθινά παραδείγματα.

Η δομή της εισόδου είναι ένα σύνολο στιγμιοτύπων. Στην απλούστερη περίπτωση κάθε στιγμιότυπο της εισόδου είναι ένα ανεξάρτητο,ατομικό παράδειγμα της έννοιας προς μάθηση. Στις περισσότερες περιπτώσεις όμως η ίδια η φύση των παραδειγμάτων είναι τέτοια, που δεν είναι εφικτό με ένα μόνο στιγμιότυπο να αναπαρασταθεί ένα παράδειγμα. Πιο συγκεκριμένα, αν η συλλογή των δεδομένων γίνεται με τέτοιο τρόπο, ώστε κάθε συλλογή(π.χ. μέτρηση φυσικού φαινομένου) να μην έχει νόημα από μόνη της ,αλλά μόνο ως σύνολο,τότε έχω την περίπτωση μη-ανεξάρτητων στιγμιοτύπων.

Στην παρούσα διπλωματική εργασία, οι διάφορες είσοδοι είχαν στην πλειοψηφία τους ανεξάρτητα στιγμιότυπα, ενώ όπου δεν ήταν οφείλετο σε λάθος κατά τη συλλογή δεδομένων. Προφανώς διορθώθηκαν πριν χρησιμοποιηθούν, ώστε να κάθε στιγμιότυπο να αποτελεί ανεξάρτητο παράδειγμα.

Κάθε στιγμιότυπο χαρακτηρίζεται από τις τιμές των χαρακτηριστικών του, τα οποία σταθμίζουν κάθε πτυχή του στιγμιότυπου. Με απλά λόγια, κάθε στιγμιότυπο αποτελείται από "στήλες" , από τιμές δηλαδή που χωρίζονται με κάποιο σύμβολο (συνήθως το κόμμα), ενώ η "μεταβλητή" που παίρνει σε κάθε στιγμιότυπο κάποια τιμή δίνεται στην αρχή του αρχείου εισόδου στη μορφή στιγμιοτύπου, προς ευκολία του προγραμματιστή και δε δίνεται ως είσοδος στον αλγόριθμο μηχανικής μάθησης. Κάθε καλώς ορισμένο αρχείο εισόδου δίνει ως πρώτο στιγμιότυπο τη σημασία των χαρακτηριστικών , έστω και σε μορφή σχολίων.

Παρά την εννοιολογική σημασία των στιγμιοτύπων, ως τιμές λαμβάνουν συνήθως συγκεκριμένα ήδη περιεχομένου,σχετικά μικρά σε αριθμό στα πλαίσια της εξόρυξης δεδομένων: αριθμητικές τιμές ή ονομαστικές(κατηγορηματικές).

2.1 Έννοια και Είδη Μάθησης

Τα βασικά διαφορετικά ήδη μάθησης που εμφανίζονται είναι 4 σε πλήθος. Στην μάθηση ταξινόμησης , ένα σύνολο από παραδείγματα (training set) χρησιμοποιείται ως μέσο προπόνηση του αλγορίθμου μάθησης , έτσι ώστε να μάθει έναν τρόπο να ταξινομεί και άγνωστα παραδείγματα, το λεγόμενο (test set). Στη συσχετική μάθηση αναζητείται κάθε συσχέτιση μεταξύ των χαρακτηριστικών, όχι μόνο αυτές

που προσφέρουν την ονομαστική τιμή ενός συγκεκριμένου attribute. Στην συσταδοποίηση αναζητούνται ομάδες παραδειγμάτων που ανήκουν μαζί. Στην αριθμητική πρόβλεψη το αποτέλεσμα της πρόβλεψης είναι μια αριθμητική ποσότητα και όχι μια διακριτή κλάση.

Από εδώ και στο εξής, κλάση ονομάζουμε κάθε μία από τις πιθανές τιμές που μπορεί να λάβει ένα attribute που δέχεται ονομαστικές τιμές, με την προϋπόθεση τις τιμές αυτού του attribute να προβλέπει ο ταξινομητής.

Επιπροσθέτως να επισημάνουμε προς αποσαφήνιση, ότι η αριθμητική πρόβλεψη είναι και αυτή μια μάθηση ταξινόμησης, όπως προκύπτει από τον ορισμό της. Συνεπώς η τετράδα τρόπος μάθησης ταξινόμησης,συσχετική,συστάδες,αριθμητική , θα ήταν σωστή και ως: ταξινόμησης ονομαστικής , αριθμητικής , συσχετική , συστάδες

Διαγραμματικά:

“έννοια” (concept) ⟶ τρόπος εκμάθησης ταξινόμησης,συσχετική,συστάδες,αριθμητική ⟶

⟶ “περιγραφή μιας έννοιας”

Στην παρούσα διπλωματική εργασία θα πραγματοποιηθεί αριθμητική πρόβλεψη ταξινόμησης.

Τα μοτίβα στα δεδομένα εισόδου μπορούν να περιγράψουν τη δομή της εισόδου. Τα μοτίβα αυτά ευρίσκονται με τη συσχετική μάθηση. Συνεπώς η βασική διαφορά των συσχετικών κανόνων σε σχέση με τους κανόνες ταξινόμησης, είναι καταρχάς ότι οι πρώτοι μπορούν να προβλέψουν την τιμή κάθε attribute ,και όχι μόνο ενός, και επίσης ίσως προβλέψουν για κάθε attribute πολλές τιμές ανά πρόβλεψη. Συνεπώς υπάρχουν σημαντικά περισσότεροι σε πλήθος κανόνες συσχέτισης από ότι ταξινόμησης, και πρόκληση είναι να καταφέρουμε να μην προκύψουν υπερβολικά πολλοί.

Οι κανόνες συσχέτισης συνήθως παράγονται ορίζοντας δύο κατώφλια: να συσχετίζουν έναν ελάχιστο πλήθος εκ των παραδειγμάτων εισόδου, και να συσχετίζουν με ένα ελάχιστο ποσοστό ακρίβειας. Το δεύτερο κατώφλι είναι πιο υψηλό και σημαντικό. Δυστυχώς και πάλι προκύπτουν πολλοί. Τέλος, συνήθως εφαρμόζονται σε ονομαστικά attributes.

Όσον αφορά την συσταδοποίση, η βασική της χρησιμότητα είναι να ομαδοποιεί παραδείγματα που ταιριάζουν μαζί. Ως παράδειγμα, αν φανταστούμε μία είσοδο δεδομένων της οποίας ένα attribute απαλείφεται. Προφανώς η συσταδοποίηση θα φτιάξει τόσες συστάδες από την νέα είσοδο όσες και οι διαφορετικές τιμές που μπορεί να πάρει το συγκεκριμένο attribute που απαλείφθηκε. Δηλαδή τα παραδείγματα θα ομαδοποιηθούν ανά τιμή του απαλειμμένου attribute. Αν όμως υπάρχουν και άλλες ομοιότητες, ίσως φτιαχτούν παραπάνω ομάδες.

Συνεπώς ο στόχος είναι να φτιαχτεί ένας μηχανισμός συσταδοποίησης που παράγει συστάδες που μπορούν να φανούν χρήσιμες στον χρήστη, και που μπορεί να ταξινομεί μελλοντικά παραδείγματα στη

σωστή συστάδα. Το τελευταίο μπορεί να βελτιωθεί με χρήση ταξινόμησης.

2.2 Είδη Αλγορίθμων Ταξινόμησης Ξανά

Όπως είδαμε προηγουμένως, η μάθηση ταξινόμησης είναι κατά κόρον επιβλεπόμενη: Προπονείται ο αλγόριθμος σε ένα training set και ο ταξινομητής που προκύπτει θα δοκιμαστεί σε ένα test set. Η επιτυχία στην κατασκευή του ταξινομητή μετριέται συνήθως με το ποσοστό των σωστών ταξινομήσεων επί του test set. Αυτή είναι μια αντικειμενική προσέγγιση. Πιο υποκειμενικά, μπορεί κάποιος να αξιολογήσει και κατά πόσο η δομή του ταξινομητή , π.χ. οι κανόνες ενός δέντρου αποφάσεων, είναι εύκολα κατανοητοί από ανθρώπινους χειριστές.

Στην παρούσα διπλωματική εργασία θα χρησιμοποιήσουμε την αντικειμενική προσέγγιση αξιολόγησης του ταξινομητή.

2.3 Instances και Αποκανονικοποίηση

Είδαμε παραπάνω ότι η δομή της εισόδου είναι ένα σύνολο στιγμιοτύπων. Θεωρώντας την απλούστερη περίπτωση όπου κάθε στιγμιότυπο της εισόδου είναι ένα ανεξάρτητο,ατομικό παράδειγμα της έννοιας προς μάθηση, τότε κάθε dataset ορίζεται ως ένας πίνακας που αποτελεί ένα υποσύνολο τιμών του πολλαπλασιασμού (relation)(σχέση) των πεδίων ορισμού instances και attributes. Δηλαδή dataset <= (instances) x (attributes).

Αν και είναι περιοριστικό να αποτελούνται τα datasets από ανεξάρτητα instances, είναι ωστόσο απαραίτητο. Αν σκεφτούμε την περίπτωση των γενεαλογικών δέντρων , και ασχοληθούμε με μία συγκεκριμένη συγγένεια, π.χ. αν το εξεταζόμενο άτομο είναι αδερφή κάποιου άλλου, τότε η πιο ρηχή προσέγγιση είναι να βρούμε όλους τους πιθανούς συνδυασμούς. Επειδή όμως ένας τέτοιος πίνακας συνδυασμών απαιτεί οπτική χρήση του δέντρου, ενώ εγώ θέλω η είσοδος να έχει μια πληρότητα πριν τη χρησιμοποιήσει ο αλγόριθμος μάθησης, τότε ο απλούστερος ανεξάρτητος πίνακας που μπορεί να προκύψει είναι ένας που εμπεριέχει και τα ονόματα των γονέων τους εξεταζόμενου ατόμου. Ανεξάρτητος ως προς το δέντρο μόνο. Ο πίνακας αυτός θα γινόταν ακόμη πιο ελκυστικός αν περιείχε μόνο τις καταφατικές περιπτώσεις. Θα ήταν μικρός σε έκταση και πλήρης. Αυτή η υπόθεση κλειστού κόσμου όμως, δηλαδή η υπόθεση ότι ξέρω όλες τις καταφατικές απαντήσεις ή γενικότερα ότι καλύπτονται όλες οι περιπτώσεις, είναι συνήθως ουτοπική.

Ο νέος ιδανικός πίνακας δεν περιέχει όμως ανεξάρτητα instances,κάτι απαραίτητο για την ταξινόμηση. Τα ονόματα των ατόμων που εξετάζονται ως προς τη συγγένεια “αδερφή” συγκρίνονται δύο φορές όταν είναι και τα δύο κορίτσια. Με την δημιουργία ενός νέου πίνακα με πλήρη τα στοιχεία των γονέων

και των δύο συγκρινόμενων ατόμων επιτυγχάνεται η ανεξαρτησία των instances, ως προς τη συγγενική σχέση “αδερφή”. Άρα έχτισα ένα ορθό dataset.

Η σχέση μεταξύ αδερφών και η σχέση μεταξύ γονέων έγιναν μία μεγάλη πεπλατυσμένη. Η διαδικασία αυτή στις βάσεις δεδομένων λέγεται αποκανονικοποίηση (denormalization). Είναι πάντα εφικτή σε ένα πεπερασμένο σύνολο από σχέσεις. Ένα προφανές πρόβλημα της αποκανονικοποίησης , και γενικότερα της ανάγκης για ανεξάρτητα instances, είναι ότι για εύρεση συγγενικής σχέσης στο παράδειγμά μας μεταξύ ατόμων που απέχουν γενεαλογικά πολύ παραπάνω του ενός βήματος, η τελική σχέση θα είναι τόσο πλατιά που υπολογιστικά και κοστολογικά θα είναι απαγορευτική.

Ένα άλλο ελάττωμα της αποκανονικοποίησης είναι ότι προκύπτουν στην τελική σχέση κάποια μοτίβα που είναι ψευδεπίγραφα ή τετριμμένα.

2.3.1 Άλλα Είδη Παραδειγμάτων

Υπάρχουν και άλλα ήδη παραδειγμάτων , όπως τα δομημένα παραδείγματα, όπου γράφοι ή δέντρα μπορούν να θεωρηθούν ειδικές περιπτώσεις σχέσεων, οι οποίες συχνά απεικονίζονται σε ανεξάρτητα instances, εξάγοντας από τη δομή των δομημένα παραδείγματα τοπικά ή γενικά χαρακτηριστικά ώστε να χρησιμοποιηθούν ως attributes. Παρομοίως, ακολουθίες αντικειμένων ή τα επιμέρους αντικείμενά τους μπορούν να περιγραφούν από ένα σύνολο από ιδιότητες , οι οποίες θα λειτουργήσουν ως attributes.

Γυρνώντας πίσω στα παραδείγματα που αντιπροσωπεύονται από πολλά instances, να σημειώσουμε ότι εμφανίζονται σε πολλές εφαρμογές του πραγματικού κόσμου. Παραδείγματος χάριν στις βάσεις δεδομένων όταν θέλουμε να συσχετίσουμε μία σειρά μίας σχέσης με πολλές σειρές μίας άλλης. Η τελική σχέση θα έχει εξαρτήσεις, οι οποίες μπορούν να παρακαμφθούν μέσω ταξινόμησης ανά τιμή του χαρακτηριστικού που διαφοροποιούσε τις σειρές της μίας παλιάς σχέσης. Προκύπτουν λοιπόν έτσι ανεξάρτητα παραδείγματα – instances.

2.4 Attributes και Είδη Τιμών

Επιστρέφοντας στα “χαρακτηριστικά” του instance, δηλαδή τα attributes, είναι σημαντικό να αναφερθεί ότι κάποια instances ίσως είναι κάπως γενικευμένα (π.χ. μέσο μεταφοράς αντί για μοτοσυκλέτα) με αποτέλεσμα να μην έχουν όλα τα attributes νόημα για κάθε instance. Σε αυτήν την περίπτωση επιλέγεται από τεχνικής σκοπιάς ένας συμβολισμός που υποδηλώνει κάποιο attribute ως αδιάφορο, συνήθως ένα “?”. Το ίδιο πράττουμε όταν η ύπαρξη της τιμής κάποιου attribute εξαρτάται από την τιμή κάποιου άλλου, ή όταν απλώς δεν είναι προσβάσιμη μία τιμή ως δεδομένο (δηλαδή όταν είναι άγνωστη).

Οι ονομαστικές τιμές ενός attribute ορίζονται ως ένα σταθερό σύνολο από διακριτά σύμβολα (π.χ.

λέξεις). Η μόνη εργασία που μπορεί να λάβει χώρα μεταξύ δύο ονομαστικών τιμών είναι ο έλεγχος ισότητας. Αριθμητικές πράξεις ή ποσοτική σύγκριση είναι άτοπες.

Έναντι του να θεωρηθούν αριθμητικές τιμές μόνο οι ακέραιοι και οι πραγματικοί, έχουν προταθεί επίπεδα για τη μέτρηση, που ακολουθούν παρακάτω:

Οι τακτικές (ordinal) τιμές κρύβουν στο εννοιολογικό περιεχόμενό τους μία έννοια σύγκρισης. Π.χ. Το χλιαρό είναι πιο ζεστό από το κρύο. Πάλι όμως δε μπορώ να αποφανθώ για το πόσο πιο ζεστό είναι το χλιαρό. Προαπαιτούμενο όμως είναι να είναι ξεκάθαρη η εννοιολογική διάκριση των τακτικών τιμών, προκειμένου να μην πέσουν στην κατηγορία των ονομαστικών.

Οι με-διάστημα (interval) τιμές είναι αυτές που έχουν αριθμητική τιμή, αλλά νόημα έχουν μόνο οι σχετικές πράξεις (αφαίρεση, μέσος όρος) ,αλλά όχι άλλες, καθώς δεν είναι σαφές ποιο είναι το μηδέν στο φαινόμενο που περιγράφουν. Π.χ. Οι ημερομηνίες. Στην διάρκεια της ιστορίας η χρονολόγηση συνεχώς αλλάζει.

Οι αναλογικές (ratio) τιμές έχουν σημείο μηδέν να αναφερθούν, συνεπώς είναι οι αντίστοιχοι πραγματικοί αριθμοί.

Ειδική περίπτωση της σκάλας των ονομαστικών τιμών είναι οι boolean τιμές, όπου στη μηχανική μάθηση σημαίνουν ότι το πεδίο ορισμού του συγκεκριμένου attribute είναι σύνολο 2 μόνο στοιχείων.

2.5 Προετοιμάζοντας την Είσοδο

Ένα μεγάλο μερίδιο του χρόνου της διαδικασίας του data mining αφορά την προετοιμασία της εισόδου δεδομένων προκειμένου να αποκτήσει την κατάλληλη μορφή για να χρησιμοποιηθεί από τον αλγόριθμο μηχανικής μάθησης. Πρώτο μέλημα είναι σαφώς η συλλογή των δεδομένων σε ένα τελικό σύνολο από instances. Σε αυτό το σημείο και επειδή οι πηγές συνήθως ποικίλουν, είναι απαραίτητη η ομογενοποίηση των τιμών στα ίδια attributes ώστε οι ίδιες τιμές να έχουν τον ίδιο συμβολισμό. Και τα ίδια τα ονόματα των attributes χρειάζονται διόρθωση. Επίσης κάποια attributes ίσως πρέπει να αφαιρεθούν , ενώ κάποια άλλα να έχουν "άγνωστη τιμή" στο τελικό dataset.

Υπενθυμίζουμε και την διαδικασία της αποκανονικοποίησης που μελετήσαμε παραπάνω με το παράδειγμα του γενεαλογικού δέντρου.

Δεν είναι όμως πάντα θέμα συμβολισμών. Η διαφορετική μεθοδολογία συγκέντρωσης δεδομένων προκαλεί και ολικά διαφορετική δομή στην κάθε πηγή δεδομένων, με αποτέλεσμα η εξυγίανση των δεδομένων να γίνεται πιο σύνθετη. Πόσο μάλλον όταν κάθε δημιουργός πηγής δεδομένων έχει κάνει τα δικά του λάθη, με μόνη λύση την επίπονη, χειρωνακτική μελέτη των πηγών και διόρθωσή τους, με τη δυσκολία να αυξάνει μαζί με το μέγεθος των δεδομένων.

Στην παρούσα διπλωματική εργασία αντιμετωπίστηκαν όλα τα παραπάνω προβλήματα , δε χρειάστηκε όμως η διαδικασία της αποκανονικοποίσης.

Η λήψη αποφάσεων απαιτεί κάποιες φορές τον διαχωρισμό κάποιων συναθροισμένων στοιχείων πίσω σε μία πιο ατομική μορφή (aggregation), με σκοπό τη μελέτη της μορφής αυτής. Τέτοιες ανάγκες πρέπει να λαμβάνονται υπόψιν κατά τη δημιουργία του τελικού dataset, δηλαδή να έχει ήδη αποφασιστεί σε ποιο βαθμό θα συναθροιστούν ή το αντίθετο τα δεδομένα των πηγών.

Διαγραμματικά έχει ως εξής:

data assembly ⟶ integration ⟶ cleaning ⟶ aggregating ⟶ general preparation

2.5.1 Τιμές που Λείπουν

Είδαμε προηγουμένως ότι οι τιμές που λείπουν (Missing Values), με την έννοια ότι δεν υπάρχουν οι επιθυμητές, έχουν αντικατασταθεί με μία όντως κενή θέση ή σύμβολο που δηλώνει έλλειψη (? ή -) , ή με ακραίες τιμές που δεν είναι επιτρεπτές στο υπό εξέταση φαινόμενο (π.χ. αρνητικοί αριθμοί όταν επιτρέπονται μόνο θετικοί ή μηδέν σε φαινόμενο που το μηδέν δεν ορίζεται).

Η έλλειψη τιμής δεν συνεπάγεται αυτόματα ότι είναι ασήμαντη. Μπορεί να οφείλεται σε ελαττωματικό εξοπλισμό, σε αλλαγές στην πειραματική διαδικασία κατά τη συλλογή δεδομένων ή σε ένωση πολλών παρόμοιων,αλλά όχι πανομοιότυπων datasets. Ερωτηθέντες κατά την συμπλήρωση ερωτηματολογίων αρνούνται να δώσουν σε όλα απαντήσεις, ένα αρχαιολογικό εύρημα μπορεί να έχει φθαρεί υπερβολικά ή πειραματόζωα μπορεί να πεθάνουν πρόωρα.

Για να δοθεί κατάλληλη ερμηνεία στο τι σημαίνει να λείπει κάποια τιμή, πρέπει να αποφανθεί κάποιος με γνώση επί του αντικειμένου ή και του λόγου που οι τιμές λείπουν. Σε περιπτώσεις όμως που διαφορετικά είδη τιμών λείπουν, γίνεται εύκολα αντιληπτό ότι υπάρχει κάποιο σοβαρό λάθος στο data mining ή στην προετοιμασία του τελικού dataset. Μια ιδιαίτερη περίπτωση είναι η κατηγορία με περιπτώσεις παρόμοιες με αυτήν των ιατρικών διαγνώσεων, όπου όταν ο γιατρός επιλέξει έξυπνα ένα σύνολο εξετάσεων για τον ασθενή, είτε θα επαληθεύσουν ότι όντως έχει μια συγκεκριμένη ασθένεια, είτε αφαιρετικά για τα συγκεκριμένα συμπτώματα θα καταλάβει την ασθένεια του ασθενή, ακόμη και αν δεν έχει ακόμη επαληθευτεί.

2.5.2 Ανακριβείς Τιμές

Στην πλειοψηφία των περιπτώσεων τα δεδομένα δεν συλλέγονται για σκοπούς data mining. Κατά τη συλλογή τους κάποιες τιμές ίσως να ήταν αδιάφορες, ή μικρής σημασίας η αποφυγή λαθών. Επιπλέον, τυπογραφικά λάθη αλλάζουν τελείως το ποια είναι τιμή, ειδικά αν είναι ονομαστική, και έτσι αυξάνει το

πλήθος των πιθανών τιμών. Άλλες φορές γράφεται η ίδια τιμή με δύο τρόπους, όπως π.χ. όταν αλλάζουν τα ονόματα προϊόντων για λόγους marketing.

Τυπογραφικά λάθη ή λάθη μετρήσεων σε αριθμητικές τιμές συχνά παράγουν ακραίες τιμές, οι οποίες μπορούν να εντοπιστούν σχηματίζοντας ένα διάγραμμα ανά attribute του dataset. Ή όταν δεν είναι ακραίες έχουν μεγάλη απόκλιση από τις υπόλοιπες. Και εδώ όμως ο εντοπισμός του εάν η απόκλιση σημαίνει ανακρίβεια απαιτεί την κρίση κάποιου ειδικού στο φαινόμενο από όπου προέρχονται τα δεδομένα.

Οι διπλές εγγραφές αποτελούν άλλη μία πηγή λαθών. Οι περισσότεροι αλγόριθμοι μηχανικής μάθησης δίνουν διαφορετικά αποτελέσματα όταν υπάρχουν επαναλαμβανόμενα δεδομένα, επειδή η επανάληψη δίνει αυξημένη βαρύτητα σε μία τιμή.

Επιπροσθέτως, πολλοί από όσους παρέχουν στοιχεία που καταλήγουν σε βάσεις δεδομένων κάνουν σκόπιμα λάθη. Βάζουν λάθος διευθύνσεις για να μην λαμβάνουν ταχυδρομείο διαφημιστικού περιεχομένου,ή πιο συμπεριφέρονται πιο ακραία όπως να αλλάζουν το όνομά τους αν απορρίφθηκε παλαιότερα η αίτησή τους για ασφάλιση. Προφανώς έχουν σχεδιαστεί συστήματα εισόδου δεδομένων που να αποφεύγουν τέτοιες καταγραφές. Παρομοίως οι ταμίες των σουπερμάρκετ χρησιμοποιούν τις δικές τους κάρτες πίστωσης πόντων κατά την πληρωμή πελάτη που δεν παρέχει μία, είτε ως πολιτική του μαγαζιού για να εξυπηρετήσει τους πελάτες , είτε ως πρωτοβουλία του ταμία να κερδίζει πολλούς περισσότερους πόντους.

Τέλος, τα δεδομένα παλιώνουν. Με την πάροδο ετών , διευθύνσεις , προσωπικά νούμερα και emails είναι πολύ πιθανό να αλλάζουν. Γεγονός που πρέπει να ληφθεί υπόψιν για να προκύψουν χρήσιμα αποτελέσματα αν εφαρμοστεί μηχανική μάθηση.

2.6 Γνωριμία με τα Δεδομένα Εισόδου

Επειδή δε μπορεί να βρεθεί πάντα κάποιος ειδικός να εξηγήσει τις ανωμαλίες στα δεδομένα, απλά εργαλεία μπορούν να βοηθήσουν στην μελέτη της εισόδου. Εργαλεία που δείχνουν ιστογράμματα της κατανομής των τιμών ονομαστικών attributes, και γράφους των τιμών των αριθμητικών attributes. Οι γραφικές αναπαραστάσεις δεδομένων καθιστούν εύκολο τον εντοπισμό ακραίων τιμών ή απόκρυφων συμβάσεων για την κωδικοποίηση ασυνήθιστων περιπτώσεων, για τις οποίες δεν υπήρξε ενημέρωση.

Η δημιουργία σχεδιαγράμματος με δύο αριθμητικούς άξονες δύο διαφορετικά attributes, και κουκκίδες δύο χρωμάτων ή συμβόλων σε κάθε υπαρκτή τιμή είναι επίσης πολύ αποκαλυπτική.

Τέλος, επειδή η είσοδος συνήθως είναι υπερβολικά μεγάλη, ο μόνος τρόπος να ελεγχθούν τα δεδομένα, είναι η μελέτη ενός επαρκούς και αντιπροσωπευτικού δείγματος από instances.

3 Έξοδος : Αναπαράσταση της Γνώσης

Δόθηκε παραπάνω ο λόγος ύπαρξης των μοτίβων που ανακαλύπτονται από τους αλγορίθμους μηχανικής μάθησης και ο ορισμός τους. Εδώ θα δούμε πώς αναπαρίστανται , και υπάρχουν πολλοί τρόποι για αυτό. Κάθε ένας υποδεικνύει και την τεχνική που θα χρησιμοποιηθεί για τη δόμηση των μοτίβων. Ήδη μιλήσαμε για δέντρα αποφάσεων και κανόνες. Οι δε κανόνες δε χρειάζεται να είναι απλοί: μπορούν να επιτρέπουν εξαιρέσεις ή να εκφράζουν σχέσεις μεταξύ των τιμών των attributes διαφορετικών instances.

Επίσης, εφόσον κάποια προβλήματα ταξινόμησης εκφράζονται σε αριθμητικές κλάσεις (δηλαδή μάθηση με αριθμητική πρόβλεψη) τα μοτίβα τους αναπαρίστανται με γραμμικά μοντέλα. Τα δε γραμμικά μοντέλα μπορούν να χρησιμοποιηθούν και για την επίλυση δυαδικής ταξινόμησης. Για αριθμητική πρόβλεψη μπορούν ,αντί για linear models, να χρησιμοποιηθούν δέντρα σε ειδική μορφή.

Η αναπαράσταση με "έμφαση στα instances" (instance-based) επικεντρώνεται στα ίδια τα instances παρά στους κανόνες που διέπουν τις τιμές των attributes.

3.1 Πίνακες Αποφάσεων

Ο πιο απλός , στοιχειώδης τρόπος αναπαράστασης της εξόδου της μηχανικής μάθησης είναι να αναπαρίσταται όμοια με την είσοδο, ως πίνακας αποφάσεων (Decision Table). Εφαρμόζεται και σε αριθμητικές τιμές και ο πίνακας εξόδου ονομάζεται τότε πίνακας παλινδρόμησης. Απλώς κοιτά κανείς αν υπάρχουν οι απαραίτητες συνθήκες για να ληφθεί μια απόφαση. Στην πράξη, πίνακας αποφάσεων είναι και η ίδια η είσοδος όταν έχουν συλλεχθεί σε μία κατάλληλα διαμορφωμένη τελική είσοδο τα δεδομένα του προβλήματος! Η απαλοιφή κάποιων στηλών (attributes) θα προσέφερε έναν καταλληλότερο πίνακα, αλλά η όλη δυσκολία έγκειται στη σωστή επιλογή στηλών που δε θα επηρεάσουν το τελικό αποτέλεσμα.

3.2 Γραμμικά Μοντέλα

Η έξοδος του γραμμικού μοντέλου (Linear Model), δηλαδή η τιμή της κλάσης, είναι απλώς το άθροισμα των τιμών των attributes, με ειδικά βάρη να δίνονται στο κάθε attribute πριν την άθροιση. Δηλαδή πραγματοποιούμε ταξινόμηση με γραμμική παλινδρόμηση. Εδώ φαίνεται το άστοχο του όρου "παλινδρόμηση". Στη στατιστική σημαίνει απλώς πρόβλεψη αριθμητικής τιμής. Η δυσκολία έγκειται στη σωστή επιλογή των βαρών. Εδώ και είσοδος και η έξοδος αποτελούνται μόνο από αριθμητικές τιμές.

Το μοτίβο δίνει έξοδο μέσω μιας γραμμικής σχέσης. Δηλαδή μία εξίσωση $cn^*xn + ... + c1^*x1 + c0$ όπου c είναι τα βάρη και x τα attributes. Αν το μοτίβο χρειάζεται μόνο 2 attributes, τότε το αποτέλεσμα είναι μια διγραμμική σχέση που μπορεί να αναπαρασταθεί σε χ-y άξονες ως μια ημιευθεία με αρχή ανυψωμένη

λόγω της σταθεράς. Για 3 attributes η ημιευθεία-σύνορο γίνεται καμπύλη για παραπάνω υπερπεδίο. Προφανώς αλλαγή στην μορφή του συνόρου επιτυγχάνεται με αλλαγή στα βάρη. Τα βάρη βρίσκονται με κατάλληλες μεθόδους, όπως η μέθοδος ελαχίστων τετραγώνων.

Δυαδική ταξινόμηση επιτυγχάνεται μέσω ενός συνόρου-ημιευθείας που διαχωρίζει τελείως 2 κλάσεις. Δηλαδή, στο χ-y πεδίο, από την κάθε μεριά του συνόρου βρίσκει κανείς μόνο σημεία με την ίδια τιμή για το προς πρόβλεψη attribute, δηλαδή μόνο μια κλάση. Από την άλλη μεριά βρίσκεται η άλλη κλάση.

3.3 Δέντρα

Η διαίρει και βασίλευε προσέγγιση στην εκμάθηση από ένα σύνολο από instances οδηγεί στη μορφή των δέντρων αποφάσεων. Στους κόμβους τους ελέγχονται οι τιμές των attributes, συγκρινόμενες με μία σταθερά. Στους κόμβους-φύλλα των δέντρων βρίσκονται οι αποφάσεις. Κάθε φύλλο έχει είτε την κλάση ταξινόμησης ενός instance, είτε ενός συνόλου από instances , είτε μία κατανομή πιθανοτήτων για όλες τις κλάσεις. Μία νέα instance για να ταξινομηθεί , έχει σε κάθε κόμβο που συναντά ένα attribute της να ελέγχεται ως προς ένα κατώφλι , δρομολογούμενη έτσι προς το κατάλληλο κόμβο-φύλλο.

Σε ονομαστικά attributes δεν υπάρχει η έννοια του κατωφλίου. Κάθε κόμβος μπορεί να οδηγεί σε το πολύ τόσους όσες και οι πιθανές τιμές του. Αν η κάθε τιμή ενός ονομαστικού attribute αντιμετωπίζεται μεμονωμένα, τότε δε θα επανεμφανιστεί παρακάτω σε κόμβο. Αν τα attributes ελέγχονται μέσω υποσυνόλων των τιμών τους, τότε ίσως κάποιες ονομαστικές τιμές ελεγχθούν παραπάνω της μία φοράς. Για αριθμητικές τιμές υπάρχει κατώφλι και τα παιδιά ενός κόμβου είναι συνήθως δύο. Αν υπάρχει ποικιλία στις πιθανότητες γίνεται να προκύπτουν παραπάνω. Τα αριθμητικά attribute προφανώς συγκρίνονται πολλές φορές όσο προχωράει το δέντρο, με άλλο κατώφλι κάθε φορά. Μία εναλλακτική για τους ακέραιους αριθμούς (που δεν έχει επίδραση στους πραγματικούς οι οποίοι συγκρίνονται ως διαστήματα) είναι η παραγωγή 3 κλάδων μικρότερο,ίσο,μεγαλύτερο αντί για δύο.

Εάν τιμές που λείπουν αναγνωρισθούν ως δυνατή τιμή του attribute, τότε προκύπτει παραπάνω παρακλάδι για τον αντίστοιχο κόμβο. Εάν το ότι λείπει τιμή δεν έχει σημασία, μία εναλλακτική είναι να ακολουθεί η πορεία του ελέγχου το πιο δημοφιλές παρακλάδι για αυτόν τον κόμβο. Μια πιο σύνθετη προσέγγιση σπάει το instance σε υποσύνολα από τα attributes του, τα οποία τα ταξινομεί μέσω του δέντρου, με τον διαχωρισμό να γίνεται μέσω βαρών που υπακούν στις αναλογίες της δημοφιλίας κάθε παρακλαδιού προς τα όλα instances. Σε κάθε κατώτερο κόμβο η διαδικασία επαναλαμβάνεται. Οι κόμβοι-φύλλα που συναντώνται λαμβάνονται υπόψιν με βαρύτητα αυτήν των βαρών τους.

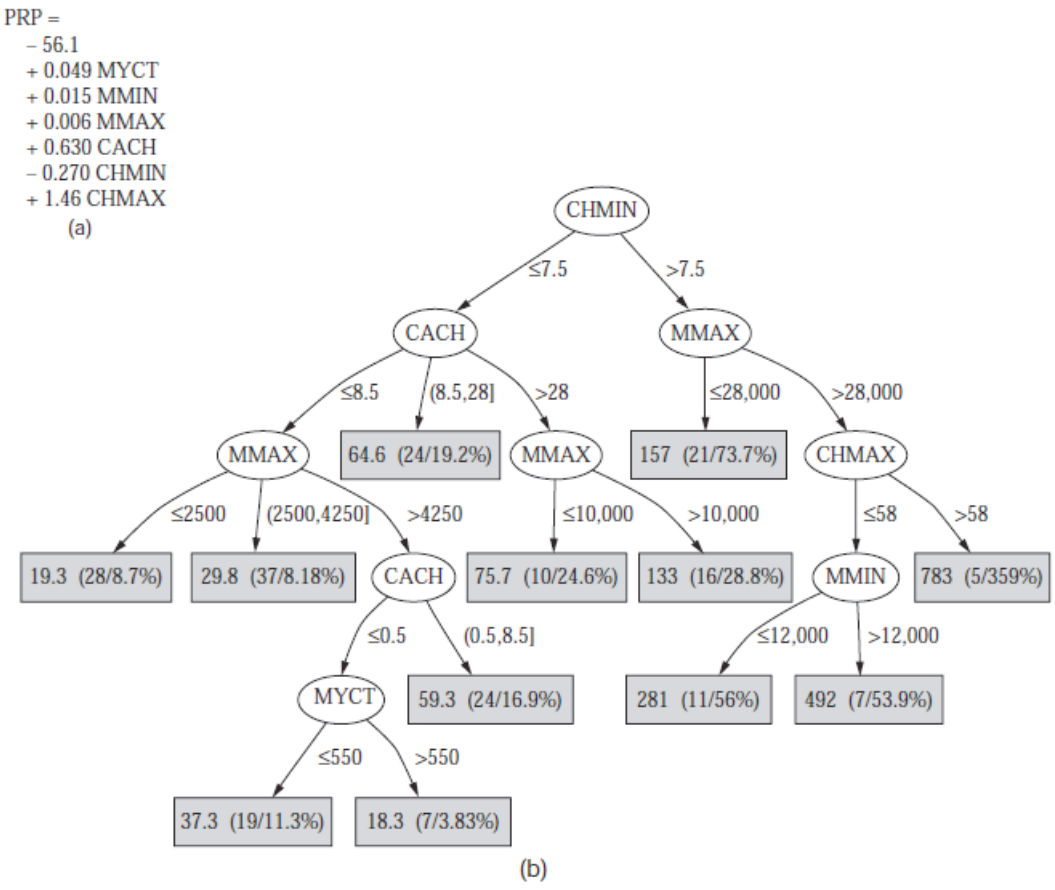
Ο έλεγχος με κατώφλια σταθερές δεν είναι ο μοναδικός τρόπος για τους κόμβους. Εναλλακτικά μερικά δέντρα συγκρίνουν δύο attributes μεταξύ τους, ή άλλα χρησιμοποιούν συναρτήσεις με είσοδο όλα

τα attributes προς παραγωγή του τελικού κατωφλίου. Είναι επίσης πιθανόν να επιθυμούμε να ελέγχονται πολλές εκδοχές μαζί: Ένας κόμβος έχει πολλές επιλογές και ακολουθούνται όλες, χτίζοντας την τελική απόφαση με κάποιον τρόπο, π.χ. τι αποφασίζει η πλειοψηφία των επιλογών.

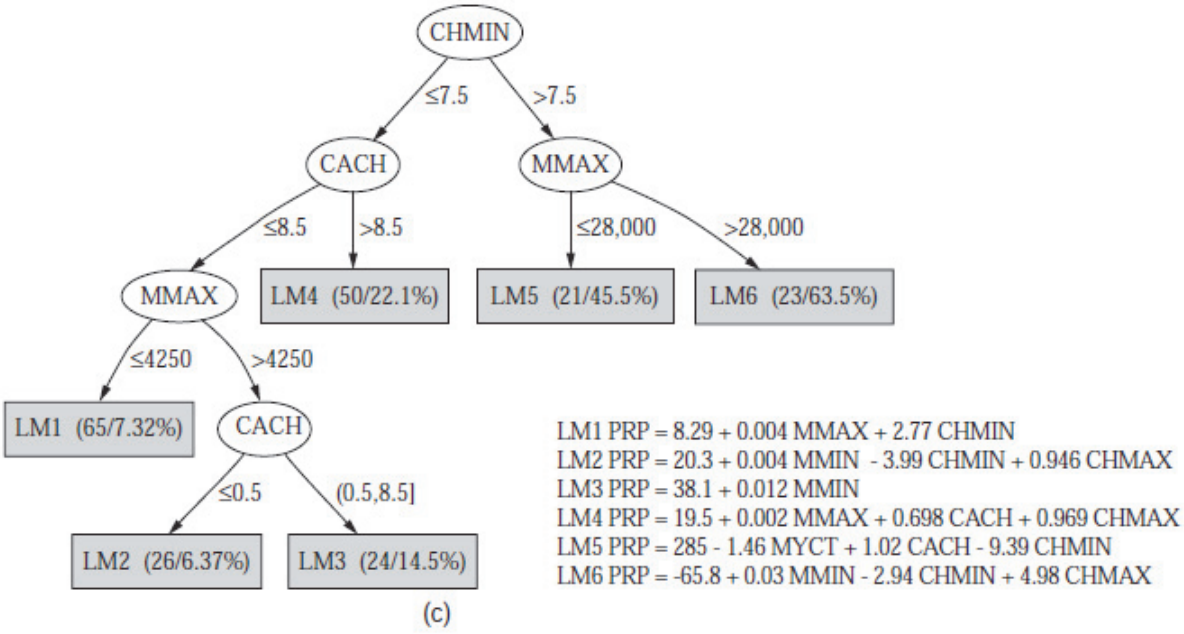
Τι συμβαίνει όμως σε ένα αριθμητικό dataset που τα instances έχουν στα attributes τους τιμές πολύ κοντά μεταξύ τους και ανά instance και ανά attribute? Τότε σε ένα δέντρο χωρίς όρια στο ύψος του θα δημιουργείτο overfitting, υπερβολική εμβάθυνση στις λεπτομέρειες του συγκεκριμένου training dataset. Το δε ύψος του θα ήταν πολύ μεγάλο, καθώς θα έπρεπε να γίνουν πολλές συγκρίσεις ανά μικρό εύρος τιμών των attributes. Συνεπώς τα δέντρα κτίζονται με ένα μέγιστο όριο ύψους, και σε κάθε κόμβο-φύλλο εμπεριέχεται ο μέσος όρος των τιμών της κλάσης των training instances που κατέληξαν σε αυτό. Τα δέντρα αριθμητικών αποφάσεων λέγονται και δέντρα παλινδρόμησης.

Σαν πιο σύνθετο, συνήθως το δέντρο είναι πιο ακριβές στις προβλέψεις του από το γραμμικό μοντέλο, αλλά με μειονέκτημα το μεγάλο μέγεθός του. Εναλλακτικά του μέσου όρου, στα φύλλα-κόμβους μπορεί να χρησιμοποιηθεί το γραμμικό μοντέλο για τις κλάσεις που καταλήγουν εκεί, δίνοντας μια πιο εκλεπτυσμένη και ακριβή μεθοδολογία πρόβλεψης, παρέχοντας συγχρόνως δέντρο μικρότερου μεγέθους.

Σχήμα 1: Παράδειγμα (α) γραμμικής παλινδρόμησης , (β) δέντρου αποφάσεων



Σχήμα 2: Παράδειγμα (c) δέντρου αποφάσεων με γραμμική παλινδρόμηση στα φύλλα



3.4 Κανόνες

Οι κανόνες είναι μία δημοφιλής εναλλακτική των δέντρων. Η συνθήκη ενός κανόνα είναι μια σειρά ελέγχων , όπως αυτοί στους κόμβους των δέντρων. Το συμπέρασμα του κανόνα δίνει την κλάση για τα instances που ικανοποιεί ή μια κατανομή πιθανοτήτων σε όλες τις κλάσεις. Οι έλεγχοι της συνθήκης συνδέονται με λογικό ΚΑΙ και προφανώς πρέπει να ισχύουν όλοι σε μία νέα instance για να πυροδοτηθεί ο κανόνας. Οι κανόνες προσπαθούμε να συνδέονται με λογικό Ή, αλλά αυτό δεν είναι πάντα εφικτό, με προβλήματα να προκύπτουν όταν πυροδοτούνται πολλοί μαζί.

3.4.1 Κανόνες Ταξινόμησης

Είναι εύκολο να διαβαστούν κανόνες ταξινόμησης από ένα δέντρο αποφάσεων. Για κάθε φύλλο του δέντρου παράγεται ένας κανόνας. Η συνθήκη του κανόνα εμπεριέχει όλες τις συνθήκες των κόμβων από το φύλλο έως και την κορυφή, και συνέπεια του κανόνα είναι η πρόβλεψη που εμπεριέχει το φύλλο. Οι κανόνες που παράγονται έτσι είναι ανεξάρτητοι και σαφείς, με την ουσιαστική ιδιότητα ότι μπορούν να εκτελεστούν εκτός σειράς,άρα όντως συνδέονται με λογικό Ή. Όπως είχαμε δει στην αναπαράσταση μοτίβων, οι κανόνες θέτονται και ως λίστες κανόνων, όπου προφανώς αν εκτελεστούν εκτός σειράς δημιουργούνται προβλήματα. Ωστόσο , κανόνες που προκύπτουν άμεσα από δέντρα είναι πολύ πιο σύνθετοι από όσο χρειάζεται, και συνεπώς “κλαδεύονται” ώστε να μην κάνουν επαναλαμβανόμενους ελέγχους.

Το πώς θα μεταφραστεί ένα σύνολο κανόνων σε δέντρο δεν είναι οφθαλμοφανές. Αν αυτοί οι κανόνες

είναι και ανεξάρτητοι της σειράς εκτέλεσης(π.χ. Έχουν στις συνθήκες τους άλλα attributes) και δεν κάνουν επαναλαμβανόμενους ελέγχους, τότε μάλλον θα κάνουν στο δέντρο: για να κτίσω το δέντρο θα διαβάσω έναν προς έναν τους κανόνες και τη δομή τους. Όσο σε κάθε νέο βήμα στο δέντρο δεν τελειώνει κάποιος , τόσο αναγκάζομαι να αρχίσω να σχεδιάζω ξανά στο δέντρο και όλους τους υπόλοιπους. Αυτό είναι το πρόβλημα των επαναλαμβανόμενων υποδέντρων. Το πρόβλημα γίνεται εντονότερο για attributes με ευρύτερο σύνολο πιθανών τιμών. Δηλαδή όπως και πριν, για κανόνες που δεν έχουν στη συνθήκη τους ίδια attributes , το να παίρνουν τα attributes 3 πιθανές τιμές αντί για 2 προκαλεί ένα κατά πολύ μεγαλύτερο δέντρο εξόδου με πληθώρα επαναλήψεων στα υποδέντρα. Αντιθέτως , οι καλώς ορισμένοι κανόνες αναδεικνύουν κάθε συμμετρία στο πρόβλημα με συνοπτικότατο τρόπο, αφού δεν υποφέρουν από επαναλήψεις.

Οι ανεξάρτητοι κανόνες ομοιάζουν ιδανικοί, αφού απεξαρτημένοι από τη σειρά εκτέλεσης δείχνουν να αντιπροσωπεύουν ένα ακέραιο κομμάτι γνώσης και να μπορούν να προστίθενται κι άλλοι δίχως να επηρεάζουν τους προηγούμενους, ενώ προσθήκη σε δέντρο θα απαιτούσε επανασχεδιασμό του δέντρου. Αυτή η ανεξαρτησία είναι όμως μία ουτοπία, καθώς αρχικά ένας αλγόριθμος μάθησης δε θα ήξερε από που να ξεκινήσει. Επίσης δεν δίνει απάντηση στο τι αποφασίζω όταν δυο διαφορετικοί κανόνες δίνουν διαφορετική πρόβλεψη για το ίδιο instance. Ούτε εάν κανένας κανόνας δεν ταξινομεί κάποιο instance. Τα δύο τελευταία δεν ισχύουν για ανεξάρτητους κανόνες που προκύπτουν από ανάγνωση δέντρων,γιατί οι επαναλήψεις των υποδέντρων απαλείφουν το συγκεκριμένο πρόβλημα .Ήδη όμως είπαμε ότι οι κανόνες από τα δέντρα είναι σύνθετοι,και άρα μη πρακτικοί.

Το πρόβλημα των πολλαπλών ταξινομήσεων μπορεί να αντιμετωπιστεί είτε με το να μην ταξινομηθεί καθόλου το instance , είτε με το να κρατηθεί το συμπέρασμα του πιο συχνά πυροδοτημένου από τους πυροδοτημένους κανόνες. Οι δύο στρατηγικές μπορεί να οδηγήσουν σε πολύ διαφορετικά αποτελέσματα. Αυτό της καθόλου-ταξινόμησης λύνεται είτε με πάλι με το να μην ταξινομηθεί καθόλου το instance είτε με το να του δοθεί ως πρόβλεψη πιο συχνή κλάση. Και αυτές οι στρατηγικές μπορεί να οδηγήσουν σε πολύ διαφορετικά αποτελέσματα.

Τα προβλήματα των ανεξάρτητων κανόνων απαλείφονται εύκολα με την υπόθεση ότι όταν δεν ισχύουν κάποιες τιμές ως κλάσεις , τότε το συμπέρασμα βρίσκεται στις άλλες κλάσεις. Άρα δε μπορεί να κτιστεί σύνολο κανόνων με διφορούμενες αποφάσεις ή καθόλου αποτέλεσμα. Αυτή η υπόθεση κλειστού κόσμου στην πράξη δύσκολα εξασφαλίζεται (σε boolean τιμές) και συνεπώς στη μηχανική μάθηση οι λίστες κανόνων, άρα και η εξάρτηση από τη σειρά εκτέλεσης είναι μονόδρομος, γεγονός που καθιστά τις εκ των υστέρων τροποποιήσεις απαγορευτικές.

3.4.2 Κανόνες Συσχέτισης

Όπως είδαμε, οι κανόνες συσχέτισης σε διαφέρουν πολύ από αυτούς της ταξινόμησης, μόνο που μπορούν να προβλέψουν οποιοδήποτε attribute, ή και συνδυασμούς από attributes. Επίσης δεν προορίζονται να χρησιμοποιηθούν ως σύνολο, αλλά καθένας τους βρίσκει διαφορετικά μοτίβα στο dataset και συνεπώς προβλέπουν διαφορετικά πράγματα. Είδαμε ότι μπορούν να προκύψουν πάρα πολλοί ακόμα και από ένα μικρό dataset, συνεπώς το ενδιαφέρον μας επικεντρώνεται σε αυτούς που εφαρμόζονται σε ένα λογικά μεγάλο κομμάτι από instances και έχουν μία φυσιολογικά μεγάλη ακρίβεια πρόβλεψης σε αυτές.

Η "κάλυψη" ή "υποστήριξη" ενός κανόνα συσχέτισης είναι το πλήθος των instances για τα οποία κάνει σωστές προβλέψεις. Η "ακρίβεια" ή αλλιώς "σιγουριά" είναι η κάλυψη ως ποσοστό. Είναι συνήθης τακτική να ορίζουμε ένα ελάχιστο επιτρεπτό όριο κάλυψης και ακριβείας, και να επιλέγουμε μόνο τους κανόνες συσχέτισης που δεν πέφτουν κάτω από κάποιο όριο. Η ύπαρξη αυτών των ορίων συνεπάγεται ότι κανόνες σύνθετοι δεν είναι σωστό να σπάνε σε απλούστερους, γιατί στους νέους απλούστερους κανόνες αυξάνουν η ακρίβεια και η σιγουριά.

Βάσει του παραπάνω σκεπτικού, και στο φαινόμενο όπου κάποιοι κανόνες συνεπάγονται άλλους, κρατάμε τον πιο εξειδικευμένο, και όχι τους απλούστερους.

3.4.3 Κανόνες με Εξαιρέσεις

Επιστρέφοντας στους κανόνες ταξινόμησης, μία φυσική τους προέκταση είναι οι εξαιρέσεις. Με αυτές μπορούν να λάβουν χώρα αυξητικές τροποποιήσεις σε ένα σύνολο κανόνων προσθέτοντάς τες στους κανόνες αντί να ανακατασκευαστεί όλο το σύνολο. Αν παραδείγματος; χάριν κάτι καινούργιο παρουσιαστεί στο υπό εξέταση φαινόμενο, τότε ένας ειδικός μπορεί να εξηγήσει ποιες τροποποιήσεις πρέπει να γίνουν σε ποιους κανόνες, προφανώς υπό τη μορφή εξαιρέσεων.

Εξαιρέσεις μπορεί να προκύψουν και για τις εξαιρέσεις. Μια αλυσιδωτή διασύνδεση εξαιρέσεων μπορεί κάλλιστα να οδηγήσει σε δομή δέντρου. Συνεπώς θα μπορούσε να αναπαρασταθεί η εννοιολογική περιγραφή των δεδομένων όχι ως απλή λίστα κανόνων, αλλά ως ένας αρχικός κανόνας με πολλά παρακλάδια εξαιρέσεων. Σε έναν τέτοιον πολυδαίδαλο κανόνα πρέπει να δοθεί μια αρχική εκδοχή, η οποία θα συνοδευτεί από την δενδροειδή μορφή εξαιρέσεων. Ως αρχική εκδοχή συνηθίζεται να τίθεται η πιο συχνή για τα instances.

Προκύπτει δηλαδή το εξής:

- *default: (κλάση)*
- *except if (...) and ...*
- *then (κλάση)*
- *except ...*
- *else*
- *...*

Η αντιμετώπιση των προβλημάτων συνηθίζεται να υλοποιείται από τους ανθρώπους ως κανόνες, εξαιρέσεις και περαιτέρω εξαιρέσεις σε αυτές, συνεπώς η παραπάνω δομή είναι πιο φιλική στον χρήστη/μελετητή των δεδομένων εισόδου. Κάθε then δήλωση οδηγεί σε μία πρόβλεψη, και είναι άμεσο και εύκολο να εντοπιστούν οι εξαιρέσεις και οι συνθήκες που οδήγησαν στην πρόβλεψη αυτή, ενώ στις λίστες αποφάσεως όλοι οι κανόνες πρέπει να μελετηθούν ώστε να φανεί πώς συντέλεσαν στο τελικό συμπέρασμα. Παρόλα αυτά, η ευκολία που προσφέρει αυτή η δομή στην τροποποίηση των κανόνων αποτελεί το βασικό της πλεονέκτημα.

Η δομή αυτή έχει τις εξαιρέσεις που εφαρμόζονται σε περισσότερα instances να συναντώνται στη δομή πιο νωρίς από ότι αυτές σε μεγαλύτερο βάθος. Αυτό είναι λογικό, καθώς νέες εξαιρέσεις που προκύπτουν ενσωματώνονται συνήθως σε μεγαλύτερο βάθος της δεντρικής δομής, αφού μάλλον προκύπτουν από μικρής έκτασης αλλαγή στο φαινόμενο,άρα εφαρμόζονται και σε λιγότερα instances.

3.4.4 Εκφραστικότεροι Κανόνες

Όπως είδαμε και στα φυλλώματα των δέντρων αποφάσεων, η σύγκριση ενός attribute με σταθερά δεν είναι μονόδρομος. Υπάρχουν κλάσεις που για να προσδιοριστούν πρέπει να λαμβάνονται υπόψιν πάνω τους ενός attribute στις συνθήκες των κανόνων. Υπολογιστικά όμως για κάποιους αλγορίθμους μηχανικής μάθησης, οι πράξεις ή συγκρίσεις μεταξύ attributes ίσως κοστίζουν πολύ. Μία λύση για να προσπεραστεί αυτό είναι οι υπολογισμοί να γίνουν εξωτερικά και το αποτέλεσμα , π.χ. μιας σύγκρισης, να προστεθεί ως επιπλέον attribute. Προφανώς αυτή η τακτική θα οδηγήσει και σε πιο απλούς και κατανοητούς κανόνες.

Το μειωμένο υπολογιστικό κόστος φαίνεται καλύτερα,εάν υποθέσουμε ότι οι κανόνες αυτοί υλοποιούνται σε πρόγραμμα επαγωγικής λογικής (υποπεδίο της μηχανικής μάθησης). Αν αυτό μελετά ένα σύνολο από instances, τότε οι επιμέρους ανά instance προβλέψεις γίνονται με αναδρομικό τρόπο.

3.5 Αναπαράσταση με Βάση τα Instances

Ο απλούστερος τρόπος μάθησης είναι η απομνημόνευση. Εφόσον ένα σύνολο από instances έχει απομνημονευτεί, για κάθε νέο instance γίνεται αναζήτηση στη μνήμη για το training instance που αναπαριστά πληρέστερα το καινούργιο. Αντί να κτίζουμε κανόνες από το dataset εισόδου, αποθηκεύουμε τα instances του στη μνήμη και κάνουμε νέες προβλέψεις βασιζόμενοι κατευθείαν σε αυτά τα instances. Υλοποιούμε δηλαδή αναπαράσταση με βάση τα Instances (instanced-based learning).

Στους προηγούμενους τρόπους αναπαράστασης της γνώσης ξεκινούσαμε μεν από τα instances, αλλά στη μνήμη αποθηκεύαμε μία περιγραφή τους (δέντρα,κανόνες,κτλ.) και όχι τα ίδια. Στο instance-based representation, η εργασία γίνεται τη στιγμή που προκύπτει νέο instance προς ταξινόμηση και όχι κατά την ανάγνωση του training dataset. Δηλαδή πρόκειται περί ενός οκνηρού σχήματος μάθησης, ενώ τα υπόλοιπα που είδαμε βιάζονται να παράξουν μία γενίκευση για τα δεδομένα. Κάθε νέο instance συγκρίνεται με τα ήδη υπάρχοντα με χρήση μιας μετρικής απόστασης, και το κοντινότερο instance χρησιμοποιείται για να δώσει την κλάση του στο καινούργιο. Αυτή είναι η τεχνική ταξινόμησης κοντινότερης-γειτνίασης.

Μερικές φορές χρησιμοποιούνται περισσότεροι τους ενός γείτονες (k-nearest-neighbor μέθοδος). Στην k-nearest, για ονομαστικές κλάσεις χρησιμοποιείται η πλειοψηφία των κλάσεων των k γειτόνων ως πρόβλεψη, ενώ για αριθμητικές κλάσεις ένας μέσος όρος με χρήση πρώτα βαρών.

Στα αριθμητικά attribute,αν υπάρχει μόνο ένα ανά instance τότε η απόσταση είναι η τετριμμένη αφαίρεση των δύο τιμών των υπό σύγκριση instances. Για παραπάνω αριθμητικά attributes χρησιμοποιείται η ευκλείδεια απόσταση, η οποία όμως προαπαιτεί τα attributes να είναι κανονικοποιημένα και ισάξιας σημασίας. Ο εντοπισμός όμως των σημαντικών attribute αποτελεί μία ουσιαστική δυσκολία της μάθησης. Η μεγαλύτερη σημασία κάποιων attribute σε σχέση με κάποια άλλα σκιαγραφείται στα βάρη που λαμβάνει εκ των προτέρων το κάθε attribute.

Στην περίπτωση ονομαστικών τιμών πρέπει να εφεύρουμε μία μετρική "απόστασης" μεταξύ των τιμών των attribute. Η απλούστερη υλοποίηση δίνει τιμή απόστασης "1" σε ίδιες τιμές και "0" σε διαφορετικές. Ευφυέστερες προσεγγίσεις δίνουν τιμή απόστασης ανάλογη με την εννοιολογική απόσταση των ονομαστικών τιμών. Παραδείγματος χάριν, η απόσταση του πορτοκαλί από το κίτρινο είναι μικρότερη από ότι με το γαλάζιο.

Όπως μαρτυρά και το όνομα της μεθόδου k-nearest-neighbor , ίσως κάποια από τα instances να μη χρειάζεται να αποθηκευτούν. Ένας λόγος είναι ότι ο αλγόριθμος ίσως αργεί πάρα πολύ με όλα τα instances εισόδου. Ή ίσως χρησιμοποιεί υπερβολικά μεγάλο κομμάτι χώρου αποθήκευσης. Στη γενική περίπτωση, κάποιες περιοχές τιμών ανά attribute είναι πιο συνεπείς και σταθερές ως προς την πρόβλεψη κλάσης, συνεπώς μερικά δείγματα-instances από αυτές τις περιοχές επαρκούν. Η εύρεση των περιοχών

αυτών αποτελεί άλλη μια δυσκολία/πρόβλημα στο instance-based learning.

Φαινομενικά, ένα πρόβλημα αυτού του είδους μάθησης είναι ότι δεν τηρεί το προαπαιτούμενο του να προσφέρει γνώση για τη δομή των δεδομένων εισόδου. Δεν είναι αληθές όμως, καθώς σε συνδυασμό με τις αποστάσεις, τα instances μπορούν να σχεδιαστούν σε ένα πεδίο, τηρώντας τις αποστάσεις αυτές. Έτσι σχεδιάζονται σύνορα που εμπεριέχουν instances με ίδια κλάση. Στην περίπτωση των 2 κλάσεων, ένας τρόπος είναι να επιλέγονται βάσει των μικρότερων αποστάσεων τα συνοριακά instances των 2 κλάσεων και να σχεδιάζονται για μικρό μήκος οι μεσοκάθετοι ανά 2 γειτονικά αντικριστά instances. Προκύπτει έτσι ένα πολύγωνο που διαχωρίζει σαφώς τις 2 κλάσεις, συνεπώς αποκτούμε γνώση για τα δεδομένα μας, ως προς αυτήν την κλάση.

Η επιλογή μόνο συνοριακών instances για το σχεδιασμό του συνόρου-πολυγώνου αποτελεί στην ουσία την επιλογή instances βάσει περιοχών. Αν όμως έχω επιλέξει με σωστό τρόπο τα instances, ακόμα και αν σχηματίσω το πολύγωνο βάσει μεσοκαθέτων από όλα τα instances, το πολύγωνο δε θα αλλάξει αισθητά, καθώς τα υπόλοιπα instances ανήκουν σε περιοχές χαμηλότερης σημασίας. Δηλαδή αποτελεί συνήθως γεγονός, πώς ο αντιπροσωπευτικός διαχωρισμός δύο συνόλων instances ως προς τις κλάσεις τους απαιτεί, τα περισσότερα instances που θα κρατηθούν στη μνήμη να είναι κοντά στα σύνορα των δύο συνόλων, ώστε όταν εφαρμοστεί ο near-neighbor κανόνας να μην δείξει λάθος κλάση λόγω έλλειψης παραδειγμάτων.

Μερικές instance-based αναπαραστάσεις ως πρώτο βήμα γενικεύουν το χώρο των instances και σχεδιάζουν ορθογώνια σε αυτόν, με κάθε ένα εξ αυτών να εμπεριέχει instances μόνο της ίδιας κλάσης. Όποιο instance δεν ανήκει σε κάποιο ορθογώνιο, ταξινομείται σε κάποιο με χρήση της μεθόδου k-nearest. Προκύπτουν όμως έτσι διαφορετικά σύνορα μεταξύ των instances ίδιας κλάσης , από όταν εφαρμόζεται κατευθείαν η μέθοδος k-nearest.

Τα ορθογώνια αυτά μοιάζουν με κανόνες, των οποίων οι συνθήκες συγκρίνουν αριθμητικά attributes με ένα πάνω και ένα κάτω κατώφλι. Η κάθε διάσταση αντιστοιχεί σε άλλο attribute και όλα αυτά συγκρίνονται με τα κατώφλια τους και συνυπάρχουν στην συνθήκη του κανόνα με χρήση λογικού ΚΑΙ. Οι κανόνες αντίστοιχοι των ορθογωνίων είναι πιο συντηρητικοί από αυτούς που παράγονται από τις μεθόδους παραγωγής κανόνων, αφού οι των ορθογωνίων έχουν πράγματι instances στα κατώφλια(ή πολύ κοντά σε αυτά) που χρησιμοποιούν για τις συγκρίσεις. Ο συντηρητισμός αυτός είναι επιτρεπτός, αφού είδαμε ότι όποιο νέο instance δεν ικανοποιεί έναν τέτοιον κανόνα ορθογωνίου, ταξινομείται πάραυτα με την k-nearest μέθοδο. Στις μεθόδους μάθησης με κανόνες , σε τέτοιες περιπτώσεις είδαμε παραπάνω ότι το instance είτε δεν ταξινομείται, είτε του δίνεται η πιο δημοφιλής τιμή.

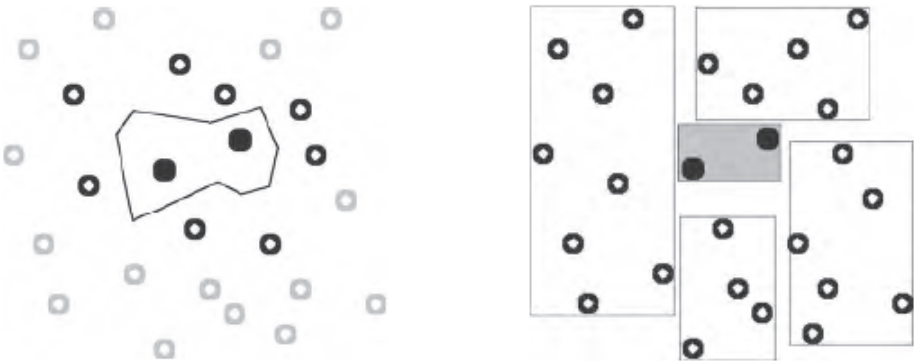
Ένα ουσιώδες πλεονέκτημα των συντηρητικών κανόνων που προκύπτουν από τα ορθογώνια, είναι

ότι είναι πιο γενικοί και διαυγείς, και δεν εμβαθύνουν πολύ στο συγκεκριμένο training dataset, όπως φαίνεται και από το γεγονός ότι τα ορθογώνια είναι σαφώς διακριτά μεταξύ τους. Επιπλέον, αυτή η μη-επικάλυψη των ορθογωνίων εξασφαλίζει ότι μόνο ένας κανόνας θα ενεργοποιείται από κάθε νέο instance, δηλαδή απαλείφει ένα σημαντικό πρόβλημα που είδαμε στην αναπαράσταση μέσω κανόνων.

Αν αντί μόνο διακριτών ορθογωνίων επιτρέπεται να ορίσουμε ορθογώνια μέσα σε ορθογώνια, όπου και πάλι όμως κάθε ορθογώνιο εμπεριέχει instances της ίδιας κλάσης, τότε οι αντίστοιχοι κανόνες που θα προκύψουν θα έχουν τη δομή κανόνων με εξαιρέσεις. Αν έχω ορθογώνια σε εσωτερικό ορθογώνιο, σημαίνει ότι έχω εξαίρεση στην εξαίρεση,κόκ.

Τέλος να σημειωθεί, ότι η αναπαράσταση instance-based μάθησης αφορά όχι μόνο τις αριθμητικές, αλλά και τις ονομαστικές τιμές.

Σχήμα 3: Instance-based διαχωρισμός κλάσεων, αριστερά με μεσοκάθετες και επιλεγμένα instances, δεξιά με διακριτές ορθογώνιες περιοχές.



3.6 Συστάδες

Όταν υλοποιώ συσταδοποίηση αντί για ταξινόμηση, η έξοδος έχει τη μορφή ενός διαγράμματος που δείχνει πως τοποθετούνται τα instances στις συστάδες (Clusters). Μια απλή περίπτωση τέτοιου διαγράμματος είναι ένας χώρος 2 διαστάσεων , όπου τοποθετούνται όλα τα instances και το επίπεδο χωρίζεται με καμπύλες που έχουν ένα κοινό σημείο. Δημιουργούνται έτσι χωρία στο επίπεδο αυτό, με το καθένα να αντιστοιχεί σε μία συστάδα.

Μερικοί αλγόριθμοι επιτρέπουν σε κάθε instance να ανήκει σε πολλές συστάδες, άρα αναγκαστικά το διάγραμμα είναι τύπου Venn: κάθε κύκλος αντιστοιχεί σε ένα σύνολο-συστάδα, όπου επιτρέπεται οι κύκλοι να επικαλύπτονται. Στα υποσύνολα επικάλυψης βρίσκονται προφανώς τα instances που ανήκουν σε πολλές συστάδες.

Άλλοι αλγόριθμοι δρουν βάσει πιθανοτήτων, αντί κατηγοριών. Σε έναν πίνακα με στήλες τις συστά-

δες και σειρές τα instances, το περιεχόμενο κάθε θέσης του πίνακα δηλώνει την πιθανότητα του συγκεκριμένου instance να ανήκει στη συγκεκριμένη συστάδα. Τα στοιχεία κάθε σειράς πρέπει αθροιστικά να δίνουν μονάδα, αφού είναι πιθανότητες.

Κάποιοι άλλοι αλγόριθμοι παράγουν δενδρογράμματα: δηλαδή ιεραρχικές δομές συστάδων με δομή δενδροειδή. Ένα μεγάλο cluster στο οποίο ανήκουν όλα τα από κάτω του, χωρίζεται σε υπο-συστάδες, αυτές σε άλλες, κόκ., έως ότου να καταλήξουμε σε συστάδες με ένα μέλος, δηλαδή τα ίδια τα instances.

Η συσταδοποίηση ακολουθείται από παραγωγή δέντρων αποφάσεων ή λίστες αποφάσεων(κανόνων) ,οι οποίοι τοποθετούν τα instances στις συστάδες. Συνεπώς η συσταδοποίηση σαν διαδικασία συμβάλει και αυτή στη δομική περιγραφή των δεδομένων εισόδου.

4 Αλγόριθμοι : Βασικές Μέθοδοι

Σε αυτήν την ενότητα θα εξηγήσουμε τις βασικές ιδέες που κρύβονται πίσω από τις τεχνικές του πρακτικού data mining. Στην άφθονη ποικιλία πιθανών datasets υπάρχουν πολλές κατηγορίες δομών που μπορούν να προκύψουν, και ένα εργαλείο εξόρυξης δεδομένων,όσο ικανό και αν είναι, κατά την αναζήτηση ενός μοτίβου στα δεδομένα ίσως προσπεράσει κάποια άλλα μοτίβα,ακόμη και στοιχειώδη. Ως αποτέλεσμα ίσως προκύψει μία αδιαφανής ταξινόμηση βασισμένη σε ένα είδος μοτίβου, αντί μίας εκλεπτυσμένης και ρητής ενός άλλου. Συνεπώς, οι διάφορες περιπτώσεις datasets απαιτούν και την κατάλληλη επιλογή ταξινομητή.

4.1 Στοιχειώδεις Κανόνες – OneR

Υπάρχουν datasets που ένα μόνο attribute είναι επαρκές για να εξάγονται από αυτό σωστές προβλέψεις, ενώ όλα τα υπόλοιπα είναι είτε άσχετα, είτε περιττά. Συνεπώς, ένας πολύ διδακτικός κανόνας για αρχή, είναι ότι οι απλές ιδέες συνήθως λειτουργούν πολύ καλά. Ένας εύκολος τρόπος παραγωγής πολύ απλών κανόνων ταξινόμησης από ένα σύνολο από instances είναι ο αλγόριθμος OneR(one rule, 1R, 1-rule), ο οποίος παράγει ένα δέντρο αποφάσεων μονού επιπέδου, εκφρασμένο ως ένα σύνολο κανόνων οι οποίοι εξετάζουν το ίδιο συγκεκριμένο attribute. Προκύπτει συχνά, ότι η απόδοση του OneR είναι πολύ υψηλή, ίσως επειδή τα πραγματικά datasets συχνά κρύβουν στοιχειώδη μοτίβα, και ένα attribute αρκεί για να βρεθεί η σωστή κλάση με υψηλή ακρίβεια.

Ξεκινώντας με τις ονομαστικές τιμές, για κάθε attribute του dataset κτίζουμε ένα σύνολο κανόνων. Οι κανόνες ανά κάθε τέτοιο σύνολο είναι ένας ανά τιμή του attribute. Για κάθε κανόνα, κοιτάμε για αυτήν την τιμή ποία κλάση εμφανίζεται στα instances εισόδου πιο συχνά, και την προσάπτουμε ως συμπέρασμα του κανόνα αυτού. Έπειτα μετράμε για κάθε εμφάνιση αυτής της τιμής του συγκεκριμένου attribute στο dataset, πόσες φορές δεν εμφανίζεται η κλάση που έχει επιλεχθεί για τον κανόνα, και διαιρούμε με το πλήθος εμφανίσεων της τιμής αυτής. Έτσι προκύπτει το ποσοστό λάθους του κανόνα. Επαναλαμβάνουμε για κάθε attribute και τιμή attribute.

Έπειτα για κάθε attribute επιλέγουμε τον κανόνα (άρα και τιμή) με το μικρότερο ποσοστό λάθους. Έπειτα από όλα τα attribute επιλέγουμε αυτό με τον κανόνα με το μικρότερο ποσοστό λάθους, και προκύπτει έτσι το τελικό ζητούμενο attribute. Ισοπαλίες στα ποσοστά λαθών αντιμετωπίζονται με τυχαία επιλογή ενός συγκρινόμενου από όλους τους ισόπαλους.

Οι τιμές που λείπουν σε ένα dataset αντιμετωπίζονται συνολικά ανά attribute: έστω και μιά φορά να λείπει τιμή σε ένα attribute, αυτό αποκτά και μία νέα τιμή στο πεδίο ορισμού του, την “missing”. Ο

χειρισμός δε των αριθμητικών τιμών , υλοποιείται μέσω της διαδικασίας της διακριτοποίησης: πρώτα ταξινομούνται τα instances ως προς το επιλεχθέν από τον αλγόριθμο attribute, και έπειτα χωρίζεται το εύρος των τιμών σε διακριτές περιοχές, οι οποίες μπορούν να ονοματιστούν, έτσι ώστε η διαδικασία του OneR που θα ακολουθήσει να είναι ολόιδια με όταν έχω ονομαστικές τιμές.

Ένας τρόπος διαχωρισμού είναι όποτε αλλάζει στην ταξινομημένη ακολουθία η τιμή της κλάσης. Το πότε αλλάζει όμως δεν είναι σαφές. Επιλέγουμε συνήθως λοιπόν να θέσουμε το σημείο διαχωρισμού στον μέσο όρο των τιμών του επιλεχθέντος attribute, των συνοριακών instances των δύο γειτονικών διαστημάτων. Όταν όμως οι τιμές είναι ίδιες, μετακινούμε το σημείο διαχωρισμού ξεπερνώντας το πρόβλημα, αλλά αποκτώντας ένα μικτό διάστημα ως προς την κλάση των instances του.

Το σοβαρότερο πρόβλημα που έχει να αντιμετωπίσει ο OneR, είναι το overfitting. Όταν το επιλεχθέν attribute έχει πολλές πιθανές τιμές, τότε δημιουργούνται πολλοί κανόνες για το attribute, όμως στην περίπτωση που οι τιμές του attribute είναι ομοιόμορφα κατανεμημένες στο dataset, τότε κάθε κανόνας θα έχει μικρό ποσοστό λάθους, γιατί διευκολύνεται η επιλογή στην κλάση να αποτελεί την μεγάλη πλειοψηφία. Στην οριακή περίπτωση, ένα attribute αποτελείται από τιμές-δείκτες-αναγνωριστικά για τα instances, με αποτέλεσμα να προκύπτουν τόσοι κανόνες όσα και instances. Προφανώς, ο τελικός κανόνας που θα δώσει ο OneR θα είναι ακραία εξαρτώμενος από το training dataset, άρα μη-αξιοποιήσιμος.

Το παραπάνω πρόβλημα overfitting προφανώς προκύπτει για αριθμητικές τιμές. Η λύση είναι να δίνεται ένα ελάχιστο όριο στο πλήθος των instances με κλάση αυτήν του κανόνα(κλάση πλειοψηφίας), στα διαστήματα της διακριτοποίησης. Σύμφωνα με τη δημοσίευση του Holte [3], μία προτεινόμενη τιμή για το όριο,η οποία προέκυψε πειραματικά, είναι το έξι.

Ακόμα και με την εφαρμογή αυτού του ορίου, αλλά και γενικότερα, όταν ένα instance έχει κλάση αυτή της πλειοψηφίας του γειτονικού της διαστήματος, μεταφέρεται αυτό το instance στο γειτονικό διάστημα, καθώς δε θα επηρεάσει το ποία είναι η κλάση πειοψηφίας. Αυτό γίνεται, γιατί επιθυμούμε όσο το δυνατόν λιγότερα διαστήματα. Στο ίδιο σκεπτικό, δύο γειτονικά διαστήματα με ίδια κλάση πλειοψηφίας συγχωνεύονται σε μία ολική με ίδια πάλι κλάση πλειοψηφίας.

Τέλος, εάν προκύψουν αριθμητικές τιμές που λείπουν, τότε δημιουργείται η τιμή “missing” , αλλά η διακριτοποίηση εφαρμόζεται μόνο στα instances που έχουν τιμή.

Εναλλακτικά, ένας πιο εκφραστικός τρόπος υλοποίησης του OneR, είναι να δημιουργείται ένας κανόνας ανά κλάση. Κάθε κανόνας είναι μια σύζευξη από ελέγχους, ένας για κάθε attribute. Στις αριθμητικές τιμές , κάθε έλεγχος κοιτά εάν η τιμή του αντίστοιχου attribute σε ένα νέο instance είναι εντός ενός διαστήματος, ενώ στις ονομαστικές εάν είναι εντός ενός υποσυνόλου τιμών , για το attribute αυτό. Κάθε αριθμητικό διάστημα αντιστοιχεί σε μία κλάση και έχει στα άκρα του το μέγιστο και το ελάχιστο

των τιμών του συγκεκριμένου attribute που εμφανίζονται για την κλάση αυτήν. Ομοίως το ονομαστικό υποσύνολο δεν έχει μεν ακρότατα, αλλά περιέχει όλες τις τιμές του attribute για αυτήν την κλάση.

Σχεδιαστικά έχει ως εξής:

1 κλάση <==> 1 κανόνας <==> 1 σύνολο ελέγχων των N attributes

o	κλάση(κανών)_1	κλάση(κανών)_2	...	κλάση(κανών)_M
o attribute_1	διάστημα(test)_1-1	...	...	...
o attribute_2	διάστημα(test)_1-2	...	...	...
o	...			
o attribute_N	διάστημα(test)_1-N	...	...	διάστημα_M-N(test_M-N)

Κανόνες διαφορετικών κλάσεων συνήθως επικαλύπτονται, δηλαδή πυροδοτούνται συγχρόνως, και τη στιγμή της πρόβλεψης επιλέγεται αυτός με τους περισσότερους επιτυχημένους ελέγχους. Αυτός ο εναλλακτικός τρόπος είναι πολύ γρήγορος και μπορεί να δουλέψει σε μεγάλες ποσότητες δεδομένων εισόδου.

4.2 Στατιστική Μοντελοποίηση – NaiveBayes

Μια άλλη απλή τεχνική αποτελεί η χρήση όλων των attributes, επιτρέποντας τη συμμετοχή τους στο τελικό αποτέλεσμα με ίδια βαρύτητα και ανεξάρτητα μεταξύ τους, με την κλάση δοσμένη. Η δοσμένη κλάση προφανώς δεν εμφανίζεται σε πραγματικά datasets, αλλά η μεθοδολογία και χωρίς δοσμένη κλάση είναι αρκετά αποδοτική. Η μεθοδολογία αυτή ονομάζεται NaiveBayes, όπου το Naive(αφελής) οφείλεται στην υπόθεση της ανεξαρτησίας των attributes. Ακόμα και σε περιπτώσεις άνισης βαρύτητας μεταξύ των attributes (π.χ. υπάρχουν επαναλαμβανόμενα attributes, δηλαδή πλεονάζοντα, ή απλώς περιττά) , υπάρχουν τεχνικές τροποποίησης του dataset εισόδου ώστε να εξαλειφθούν τα προβλήματα και η μέθοδος να δουλέψει αποδοτικά.

Η μέθοδος βασίζεται στον τύπο του Bayes για δεσμευμένες πιθανότητες, όπου για μία υπόθεση H και ένα γεγονός E ισχύει: *Pr[H|E] = Pr[E|H]*Pr[H]/Pr[E]* , όπου Pr[H] είναι η πιθανότητα να ισχύει η υπόθεση H, Pr[E] η πιθανότητα να ισχύει το γεγονός E, Pr[E|H] η πιθανότητα να ισχύει το γεγονός E εάν ισχύει η υπόθεση H, και Pr[H|E] η πιθανότητα να ισχύει η υπόθεση H εάν ισχύει το γεγονός E. Στη μέθοδό μας υπόθεση είναι μία κλάση και γεγονός ένας συνδυασμός τιμών των attributes. Εάν E1,E2,...,En είναι οι τιμές-γεγονότα των attributes και είναι ανεξάρτητα δοθείσας της κλάσης H, τότε ο τύπος δίνεται και ως: *Pr[H|E] = Pr[E1|H]*Pr[E2|H]*...*Pr[En|H]*Pr[H]/Pr[E]*, όπου Pr[E1|H]*...*Pr[En|H] είναι η συνδυασμένη Pr[E|H] πιθανότητα των γεγονότων.

Η ιδέα είναι να δημιουργηθούν οι πιθανότητες Pr[Ei|H] βάσει του δοσμένου dataset, και στη συνέχεια

για κάθε νέο instance να βρίσκονται οι πιθανότητες να ανήκει σε κάθε μία από τις κλάσεις. Ο απλούστερος τρόπος για τη δημιουργία των Pr[Ei|H] είναι να βρίσκεται ο λόγος του πλήθος των instances με γεγονός Ei και κλάση H προς το πλήθος των instances με κλάση H. Άρα προκύπτει ένας πίνακας μορφής:

o	attribute 1	...	attribute N	
o		class1 ... class M		class1 ... class M
o	att1_value1		attN_value1	
o	...	[P[Ei|Hj]]	...	[P[Ei|Hj]]
o	att1_valuek1		attN_valuekN	
o	(i=1,...,k1 , j=1,...,M)		(i=1,...,kN , j=1,...,M)	

Όταν προκύψει νέο instance, τότε για κάθε κλάση Hj , κοιτάμε ποία γεγονότα Ei ισχύουν, βρίσκουμε τα Pr[Ei|Hj] και το Pr[Hj] από τον πίνακα που έχουμε φτιάξει και παράγουμε το γινόμενο τους Pr[E|Hj]*Pr[Hj]. Ομοίως παράγουμε τα γινόμενα για κάθε κλάση. Για να γίνει το γινόμενο αυτό η πιθανότητα Pr[Hj|E] να ανήκει το νέο instance στην κλάση Hj που μελετάμε,πρέπει πρώτα να βρούμε το άθροισμα S όλων των Pr[E|Hj]*Pr[Hj] γινομένων. Τότε *Pr[Hj|E] = Pr[E|Hj]*Pr[Hj] / S*. Αυτό το βήμα είναι μια κανονικοποίηση. Ομοίως βρίσκουμε τις πιθανότητες για τις υπόλοιπες κλάσεις.

Υπάρχει ένα σοβαρό πρόβλημα όταν δημιουργεί κανείς τα P[Ei|Hj] βάσει συχνοτήτων εμφάνισης. Έστω και μία τιμή ενός attribute να μην εμφανίζεται καθόλου, τότε το γινόμενο του τύπου Bayes μηδενίζεται και ασκείται έτσι ένα "βέτο" στην παραγωγή της πιθανότητας μιας κλάσης. Σύμφωνα με την τεχνική του "Laplace estimator", μπορούμε να μετατρέψουμε κάθε Pr[Ei|Hj] από *nom/den* σε *[nom+(μ/ki)] / [den + μ]*. Επεξηγώντας τα παραπάνω, nom = πλήθος των instances με γεγονός Ei και κλάση Hj και den = πλήθος των instances με κλάση H. Επίσης "μ" είναι μία σταθερά, την οποία μπορούμε να προσαρμόσουμε όπως επιθυμούμε. Η σταθερά μ αποτελεί μια εικονική προσαύξηση στο πλήθος των instances με κλάση Hj, η οποία ισομοιράζεται σε όλα τα κλάσματα των τιμών του ίδιου attribute στην κλάση Hj (εξού και το μ/ki, όπου ki = πλήθος διαφορετικών τιμών του attribute i).

Όσο μεγαλώνει η σταθερά μ ενός attribute, τόσο εξαρτάται το αποτέλεσμα μιας νέας πρόβλεψης από το training dataset. Όσο μικραίνει, τόσο πιο ανεξάρτητη γίνεται. Επιπλέον, αν αντί για *[nom+(μ/ki)]* θέσω *[nom+(μ*pki)]*, όπου pki ένα ποσοστό που αφορά μόνο μία τιμή του attribute της μ σταθεράς, τότε μπορώ να προσαρμόζω ξεχωριστά και τη βαρύτητα της κάθε τιμής του attribute. Τέλος , μπορώ να θέτω διαφορετική σταθερά μ σε κάθε διαφορετικό attribute.

Στα πλεονεκτήματα της Bayes τεχνικής έγκειται το ότι διατυπώνεται με αυστηρό τρόπο, όμως μειονεκτεί στο ότι δεν είναι εύκολο να επιλεχθούν οι κατάλληλες σταθερές. Στην πράξη είναι αρκετό να

αρχικοποιεί κανείς όλες τις μετρήσεις στο 1 αντί για το 0 που είναι λογικό, με την προϋπόθεση πάντα ότι υπάρχει ένας μεγάλος αριθμός από instances διαθέσιμος στο training set. Αρχικοποίηση στο 1, θα σήμαινε ανά attribute i ότι μ=ki και μ*pki = 1 .

Όσον αφορά τις τιμές που λείπουν, εάν έρχεται νέα προς ταξινόμηση instance και του λείπει κάποια τιμή σε κάποιο attribute, τότε οι πιθανότητες ταξινόμησης για τις κλάσεις θα υπολογιστούν μόνο βάσει των άλλων attributes. Εάν στο training dataset υπάρχει instance με τιμή που λείπει, τότε απλώς η τιμή αυτή αγνοείται,ενώ για το συγκεκριμένο attribute οι πιθανότητες Pr[Ei|Hj] δε θα συμπεριλάβουν στο συνολικό πλήθος instances για την κλάση Hj αυτή την τιμή.

Τα παραπάνω εφαρμόζονται ως έχουν για ονομαστικές τιμές. Για αριθμητικές θεωρούμε συνήθως ότι υπάρχει στα attributes “κανονική” ή “Gaussian” κατανομή πιθανοτήτων. Απαραίτητες ενέργειες έπειτα , είναι η εύρεση του μέσου όρου και της τυπικής απόκλισης των τιμών ανά κλάση ανά attribute. Ο μέσος όρος ως γνωστόν είναι ο λόγος του αθροίσματος των τιμών προς το πλήθος τους, ενώ τυπική απόκλιση είναι η τετραγωνική ρίζα της διακύμανσης των τιμών. Η διακύμανση ανά attribute ανά κλάση βρίσκεται ως εξής: αφαιρούμε τον μέσο όρο από κάθε τιμή, τετραγωνίζουμε την κάθε αφαίρεση,προσθέτουμε όλες τις τετραγωνίσεις, και τέλος διαιρούμε το άθροισμα με το πλήθος των τιμών μειωμένο κατά ένα.

Έχοντας τα παραπάνω, μπορούμε τώρα ανά attribute ανά κλάση να βρούμε την συνάρτηση πυκνότητας πιθανότητας για κανονική κατανομή. Πρόκειται για τη συνεχή συνάρτηση: f(x) = [e^[- (x-μ)^2 / 2*σ^2]] / [π*σ*2^(-1)] , όπου μ = μέσος όρος, σ= τυπική απόκλιση, x η τιμή του νέου instance για το συγκεκριμένο attribute. Η συνάρτηση f(x) χρησιμοποιείται στις αριθμητικές τιμές αντί για το P[Ei|Hj], υπακούοντας όμως στην παραπάνω μεθοδολογία. Συνεπώς μπορούν τα γινόμενα Pr[E|Hj]*Pr[Hj] να εμπεριέχουν και P[Ei|Hj] και f(x) όρους, παρόλο που τυπικά για συγκεκριμένη τιμή x έχουμε Pr(x)=0, ενώ είναι διάφορο του μηδενός για μία μικρή περιοχή γύρω από το x. Δηλαδή εξ ορισμού Pr(x) = ε*f(x) για z-ε<x<z+ε, όπου z μια σταθερή τιμή και ε μικρή σταθερά.

Ομοίως με πριν,εάν λείπουν αριθμητικές τιμές, ο μέσος όρος και η τυπική απόκλιση υπολογίζονται μόνο με τις τιμές που δεν λείπουν.

Ένα σημαντικό πεδίο της μηχανικής μάθησης είναι η ταξινόμηση εγγράφων, στην οποία κάθε instance είναι ένα έγγραφο και η κλάση του instance είναι το θέμα του εγγράφου. Τα έγγραφα χαρακτηρίζονται από τις λέξεις που εμφανίζονται σε αυτά, και ο δρόμος προς την ταξινόμηση ξεκινά θεωρώντας την παρουσία ή απουσία μίας λέξης ως μία boolean μεταβλητή. Ο NaiveBayes αλγόριθμος είναι κατάλληλος για αυτές τις ταξινομήσεις, ως γρήγορος και ακριβής., όμως δε λαμβάνει υπόψιν τις επανεμφανίσεις των λέξεων. Μια τροποποίησή του τις λαμβάνει, θεωρώντας ένα έγγραφο ως μία συλλογή(και όχι σύνολο) από λέξεις, συνοδευόμενες από τη συχνότητα εμφάνισής τους, και λειτουργεί καλύτερα για λεξικογραφικού

μεγέθους datasets.

Τελειώνοντας με τον NaiveBayes, η κανονική κατανομή βάσει της οποίας αναπτύχθηκε η παραπάνω μεθοδολογία, δεν είναι πάντα παρούσα. Είτε τροποποιούμε τη μεθοδολογία σύμφωνα με την διαφορετική κατανομή, είτε εάν δεν υπάρχει κατανομή χρησιμοποιούμε διαδικασίες για KDE (kernel density estimation). Το KDE είναι ένας μη παραμετρικός τρόπος για την εκτίμηση της συχνότητας πυκνότητας πιθανότητας για μία τυχαία μεταβλητή.

4.3 Διαίρει και Βασίλευε : Κατασκευή Δέντρων Αποφάσεων – ID3 => C4.5

Κάποια datasets έχουν απλοϊκή λογική δομή, περιλαμβάνοντας μερικά μόνο attributes, και αυτή η δομή μπορεί να συλληφθεί με ένα δέντρο αποφάσεων. Το πρόβλημα της κατασκευής ενός δέντρου αποφάσεων μπορεί να εκφραστεί αναδρομικά, ως μία διαδικασία "διαίρει και βασίλευε", ή αλλιώς "από πάνω προς τα κάτω επαγωγή". Αρχικά επιλέγεται ένα attribute προς τοποθέτηση στον κόμβο-κορυφή,και από εκεί ξεκινούν παρακλάδια για κάθε πιθανή τιμή. Έτσι διασπάται η αρχική είσοδος σε υποσύνολα, ένα για κάθε τιμή του attribute. Η διαδικασία επαναλαμβάνεται για κάθε παρακλάδι, λαμβάνοντας υπόψιν μόνο τα instances που καταφθάνουν στο παρακλάδι αυτό. Εάν σε οποιονδήποτε κόμβο όλα τα instances έχουν ίδια κλάση, ο κόμβος αυτός γίνεται φύλλο και δεν αναπτύσσεται παραπέρα. Το μόνο που χρειάζεται, είναι ο τρόπος να επιλεχθεί το κατάλληλο attribute ανά κόμβο. Η διαδικασία αυτή αντιστοιχεί στις ονομαστικές τιμές. Η τροποποίηση του αλγορίθμου για να ισχύει και για αριθμητικές τιμές είναι αρκετά σύνθετη, και δε θα παρατεθεί εδώ. Ο ολοκληρωμένος αλγόριθμος ονομάζεται C4.5 και προέρχεται από τον ID3 .

Επειδή επιθυμούμε μικρά δέντρα, επιθυμούμε να δημιουργούνται κόμβοι-φύλλα όσο νωρίτερα γίνεται. Συνεπώς αναζητούμε τα attribute αυτά που θα δώσουν έπειτα θυγατρικούς κόμβους με την όσο δυνατόν ισχυρότερη πλειοψηφία υπέρ μιας κλάσης ανά κόμβο. Η μετρική της "αγνότητας" ενός θυγατρικού κόμβου λέγεται "πληροφορία",και μετριέται σε "bits", δίχως να είναι απαραίτητο να συσχετιστούν με τους αντίστοιχους όρους της πληροφορικής. Η ιδέα είναι, ότι στον αρχικό κόμβο-κορυφή υπάρχει το σύνολο της πληροφορίας, την οποία δημιουργώντας παρακλάδια για ένα attribute, την εξάγουμε(αφαιρούμε) από την υπολειπόμενη. Αυτή η εξαγόμενη πληροφορία λέγεται "κέρδος". Πέρα από την πληροφορία-κέρδος που αφαιρείται κατά την πράξη της διάσπασης, υπάρχει πληροφορία και στους κόμβους. Η πληροφορία ενός κόμβου είναι όση χρειάζομαι ώστε να ταξινομήσω ένα instance, δεδομένου ότι το instance αυτό έχει καταφθάσει σε αυτόν τον κόμβο. Τα δε bits μπορούν να είναι και κλάσματα, ακόμη και μικρότερα της μονάδος.

Υποθέτοντας 2 κλάσεις κ1,κ2 ,χωρίς βλάβη της γενικότητας, και σε έναν κόμβο χ εμφανίσεις της

κλάσης κ1 και ψ της κ2, τότε η πληροφορία στον κόμβο είναι info[χ,ψ]. Αν κατά τη διάσπαση ενός κόμβου ζ1, δημιουργούνται οι κόμβοι ζ2,ζ3,ζ4, τότε κέρδος διάσπασης = *info(ζ1) – info(ζ2,ζ3,ζ4)*, όπου info(ζ1) εννοούμε info[χ1,ψ1] και info(ζ2,ζ3,ζ4) = η μέση πληροφορία των κόμβων ζ2,ζ3 και ζ4. Αν όλα τα instances εισόδου είναι s σε πλήθος , και ρ είναι το πλήθος των instances σε έναν θυγατρικό κόμβο, τότε *info(ζ2,ζ3,ζ4) = (ρ2/s)*info(ζ2) + (ρ3/s)*info(ζ3) + (ρ4/s)*info(ζ4).* Όμοια με το info ανά κόμβο, η μέση πληροφορία θυγατρικών κόμβων είναι όση χρειάζομαι ώστε να ταξινομήσω ένα instance, δεδομένου ότι το instance αυτό έχει καταφθάσει σε αυτό το υποδέντρο και επίπεδο, δηλαδή στο σημείο των θυγατρικών κόμβων ζ2,ζ3,ζ4.

Η απόφαση συνεπώς για το ποίο attribute σε κάποιο σημείο του δέντρου , πρέπει να διασπαστεί σε θυγατρικούς κόμβους λαμβάνεται επιλέγοντας τη διάσπαση που προσφέρει το μεγαλύτερο κέρδος. Θυγατρικοί κόμβοι με μόνο μία κλάση συνεισφέρουν αποφασιστικά στο κέρδος διάσπασης. Δεύτεροι σε ισχύ έρχονται οι κόμβοι με μεγάλο πλήθος instances , και πολύ ισχυρή πλειοψηφία υπέρ μίας κλάσης. Η διαδικασία συνεχίζει αναδρομικά και σταματά στα φύλλα, δηλαδή όταν έχουν όλα τα instances ίδια κλάση, ή όταν τελειώσουν τα attributes για περαιτέρω διασπάσεις. Στη δεύτερη περίπτωση, συχνά τα φύλλα δεν έχουν απαραίτητα μόνο μία κλάση. Ίσως με ακριβώς τα ίδια attributes, να υπάρχουν instances με διαφορετική κλάση. Εναλλακτικά, θα μπορούσαμε να σταματήσουμε τις διασπάσεις όταν το κέρδος είναι μηδενικό, αλλά αυτή είναι πιο συντηρητική προσέγγιση του προβλήματος.

Ας επισημανθεί εδώ, ότι η πληροφορία ενός κόμβου με μία μόνο κλάση, είναι 0 bits, ενώ όταν όλες οι κλάσεις εμφανίζονται με ισόποσο τρόπο, η πληροφορία είναι στο μέγιστό της. Η πληροφορία υπολογίζεται από την συνάρτηση εντροπίας : *entropy(x1,...,xN) = - (x1/s)*log(x1/s) - ... - (xN/s)*log(xN/s)* (bits) ,όπου x είναι οι εμφανίσεις της αντίστοιχης κλάσης στον κόμβο , s είναι το πλήθος όλων των instances στον κόμβο, και log έχουν βάση το 2. Παραδείγματος χάριν, αν έχω 3 κλάσεις και info=info[x1,x2,x3], τότε ισχύει *info[x1,x2,x3] = entropy(x1,x2,x3) = – (x1/s)*log(x1/s) – (x2/s)*log(x2/s) – (x3/s)*log(x3/s).* Για p1 = (x1/s) ,κ.ό.κ., προκύπτει *info[x1,x2,x3]= – p1*logp1 – p2*logp2 – p3*logp3.* Επιπλέον ισχύει ότι p1+p2+p3 = 1 και για την εντροπία: *entropy(x1,x2,x3)= entropy(p1 , p2+p3) + (p2+p3)* entropy(p2/(p2+p3), p3/(p2+p3)).*

Όσες περισσότερες τιμές μπορεί να πάρει ένα attribute, τόσο τείνει κάθε θυγατρικός κόμβος από τη διάσπαση αυτού του attribute να έχει ισχυρότερη πλειοψηφία υπέρ μίας κλάσης, με αποτέλεσμα το info του κόμβου να μειώνονται, συνεπώς το κέρδος διάσπασης να αυξάνεται. Άρα σύμφωνα με όσα είπαμε, ο αλγόριθμος θα προτιμήσει τη διάσπαση ενός τέτοιου attribute. Στην ακραία περίπτωση ένα attribute αποτελεί αναγνωριστικό, με μία τιμή ανά instance. Μετά από μία τέτοια διάσπαση κάθε θυγατρικός κόμβος θα έχει info=info[1,0,...,0]=0. Άρα το κέρδος θα είναι μέγιστο, ίσο με την πληροφορία του κόμβου-

γονέα. Προφανώς μία τέτοια διάσπαση δε προσφέρει αληθινή γνώση για τη δομή των δεδομένων εισόδου.

Ως αντιστάθμισμα σε αυτό το πρόβλημα, χρησιμοποιείται η μετρική "αναλογία κέρδους" αντί για το ίδιο το κέρδος διάσπασης. Χρησιμοποιεί μία τροποποιημένη μορφή της "πληροφορίας",δηλαδή αλλάζει τον ορισμό του info ανά κόμβο, με τρόπο τέτοιο ώστε να ασχολείται μόνο με το πλήθος και το μέγεθος των θυγατρικών κόμβων,και όχι με τις κλάσεις. Αν πούμε το νέο info ως info2, τότε υποθέτοντας 2 πιθανές τιμές για το attribute του κόμβου-γονέα τ1,τ2 ,χωρίς βλάβη της γενικότητας, και στον θυγατρικό κόμβο που αντιστοιχεί στην τ1 υπάρχουν σ1 σε πλήθος instances και στον θυγατρικό της τ2 υπάρχουν σ2, τότε η τροποποιημένη πληροφορία στον κόμβο-γονέα είναι info2[σ1,σ2]. Οι τιμές των info2 είναι πάντα μεγαλύτερες της μονάδος. Ισχύει συνεπώς για έναν κόμβο-γονέα ζ: *info2(ζ) = info2[σ1,σ2] = – (σ1/s)*log(σ1/s) – (σ2/s)*log(σ2/s)*, όπου s είναι το πλήθος όλων των instances στον κόμβο-γονέα ζ. Όμοια ισχύει η σχέση και για περισσότερες τιμές ανά attribute. Ορίζουμε: *"κέρδος αναλογίας" κόμβου ζ = "κέρδος διάσπασης κόμβου ζ" / info2(ζ) .*

Με το κέρδος αναλογίας περιορίζεται σημαντικά η ισχύς ενός παραπλανητικού, για τον αλγόριθμο, attribute. Έπειτα με μερικούς ελέγχους μπορεί να απαλειφθεί κάθε περιττό attribute.

Τέλος γενικότερα ισχύει ως προϋπόθεση, ότι είναι απαραίτητο για κάθε attribute που επιλέγεται να διασπαστεί, να έχει κέρδος διάσπασης μεγαλύτερο ή ίσο με τον μέσο όρο κέρδους διάσπασης από όλα τα attributes που εξετάζονται.

4.4 Αλγόριθμοι Κάλυψης : Κατασκευάζοντας Κανόνες – PRISM

Υπάρχουν datasets στα οποία λίγοι ανεξάρτητοι κανόνες κυριαρχούν στην ανάθεση των instances σε διαφορετικές κλάσεις. Μία προσέγγιση λοιπόν, είναι να δρούμε σε μία κλάση κάθε φορά, και να ψάχνουμε έναν τρόπο να καλυφθούν όλα τα instances με αυτήν την κλάση, εξαιρώντας όσα δεν έχουν την κλάση αυτή (εν αντιθέσει με την "από πάνω προς τα κάτω" στρατηγική των δέντρων αποφάσεων , η οποία σε κάθε στάδιο ψάχνει ο διαχωρισμός ποίου attribute χωρίζει καλύτερα τις κλάσεις). Αυτή η προσέγγιση με κάλυψη από την ίδια της τη φύση οδηγεί σε ένα σύνολο κανόνων, παρά σε ένα δέντρο αποφάσεων. Και στην προσέγγιση μέσω δέντρων αποφάσεων το δέντρο μπορεί να μετατραπεί σε ένα σύνολο κανόνων ταξινόμησης, αλλά το να γίνουν και αποτελεσματικοί δεν είναι μία τετριμμένη διαδικασία.

Η σαφήνεια στην αναπαράσταση της δομής των δεδομένων διαφέρει μεταξύ δέντρων και κανόνων. Οι αλγόριθμοι top-down, σε αντίθεση με τους covering που ασχολούνται με μία κλάση κάθε φορά, δημιουργούν μία εννοιολογική περιγραφή για όλες τις κλάσεις σε κάθε διάσπαση, προκειμένου να είναι πιο αποδοτική αυτή η διάσπαση σε κλάσεις. Συνεπώς, όπως έχουμε δει παραπάνω, οι κανόνες μπορούν

να είναι συμμετρικοί, ενώ τα δέντρα καταλήγουν να είναι πολύ μεγαλύτερα σε μέγεθος αναπαράστασης από ότι οι κανόνες, γιατί ξεκινούν με τη διάσπαση ενός attribute.

Οι αλγόριθμοι κάλυψης λειτουργούν προσθέτοντας ελέγχους στον υπό κατασκευή κανόνα, προσπαθώντας πάντα αυτός να αποκτήσει μέγιστη ακρίβεια. Από την άλλη μεριά, οι αλγόριθμοι διαίρει και βασίλευε λειτουργούν προσθέτοντας ελέγχους στο υπό κατασκευή δέντρο, προσπαθώντας πάντα να μεγιστοποιήσουν το διαχωρισμό μεταξύ των κλάσεων. Και οι δύο προσεγγίσεις χρειάζονται κάποιο attribute να διασπάσουν: τα κριτήρια όμως για τη βέλτιστη διάσπαση είναι διαφορετικά σε κάθε περίπτωση. Ενώ ΔκΒ αλγόριθμοι όπως ο ID3 επιλέγουν σε κάθε βήμα ένα attribute αποσκοπώντας σε μεγιστοποίηση του κέρδους διάσπασης, οι αλγόριθμοι κάλυψης επιλέγουν ένα ζεύγος attribute και τιμής του ώστε να αυξηθεί η πιθανότητα της επιθυμητής ταξινόμησης σε συγκεκριμένη κλάση.

Κατά την κατασκευή ενός κανόνα,κάθε νέος όρος-συνθήκη που του προστίθεται περιορίζει την κάλυψη του κανόνα επί των instances εισόδου. Σκοπός είναι η κάλυψη όσων περισσότερων instances της επιθυμητής κλάσης και μόνο. Αν ένας κανόνας με προσθήκη νέας συνθήκης καλύπτει t σε πλήθος instances, και p εξ αυτών έχουν την επιθυμητή κλάση, τότε επιλέγω τη νέα συνθήκη έτσι ώστε ο λόγος (p / t) να μεγιστοποιείται όσο είναι δυνατόν.

Ακολουθεί ψευδοκώδικας που συνοψίζει τον PRISM, έναν βασικό ταξινομητή με κανόνες:

o Για κάθε κλάση C
o Αρχικοποίησε το E στο σύνολο των instances εισόδου
o Όσο το E εμπεριέχει instances με κλάση C
o Φτιάξε έναν κανόνα R με άδεια μία νέα θέση συνθήκης και με συμπέρασμα την C
o Μέχρι να είναι ο R τέλειος (ή να μην έχουν μείνει άλλα attributes προς χρήση) κάνε:
o Για κάθε attribute A που δεν αναφέρεται στον R, και για κάθε τιμή v,
o Εξέτασε την πρόσθεση της συνθήκης A=v στην νέα άδεια θέση του R
o Επέλεξε τα A και v που μεγιστοποιούν την ακρίβεια p/t
o (αναίρεσε ισοπαλίες διαλέγοντας τη συνθήκη με το μεγαλύτερο p)
o Πρόσθεσε το A=v στον R
o Αφαίρεσε τα instances του E που καλύπτονται από τον R

Μελετώντας τον παραπάνω ψευδοκώδικα αντιλαμβανόμαστε την γενική ιδέα της λειτουργίας του. Ο PRISM θα φτιάξει όσους κανόνες χρειάζονται για να καλυφθεί πλήρως μία κλάση,προσέχοντας πάντα δύο σημεία: Πρώτον κανένας κανόνας ανά κλάση να μην καλύπτει instances που ανήκουν σε άλλη κλάση, και δεύτερον από τον ορισμό του PRISM φροντίζεται ότι αν οι κανόνες μίας κλάσης επικαλύπτονται δεν δημιουργούν πρόβλημα, αφού ανήκουν στην ίδια κλάση,και συνεπώς μία σύγχρονη πυροδότησή

τους από κάποιο instance θα δώσει ως πρόβλεψη την ίδια κλάση.

Βάσει της δεύτερης σημείωσης, ενώ για τους κανόνες ανά κλάση δείχνει να έχει σημασία η σειρά εκτέλεσης, αφού κάθε επιπλέον κανόνας που κτίζεται για μία κλάση πυροδοτείται μόνο από όσα αποδεκτά instances δεν κάλυψαν οι προηγούμενοι κανόνες της κλάσης αυτής,στην ουσία δεν έχει σημασία η σειρά,αφού και να επικαλύπτονται οι κανόνες,ίδιο συμπέρασμα δίνουν.

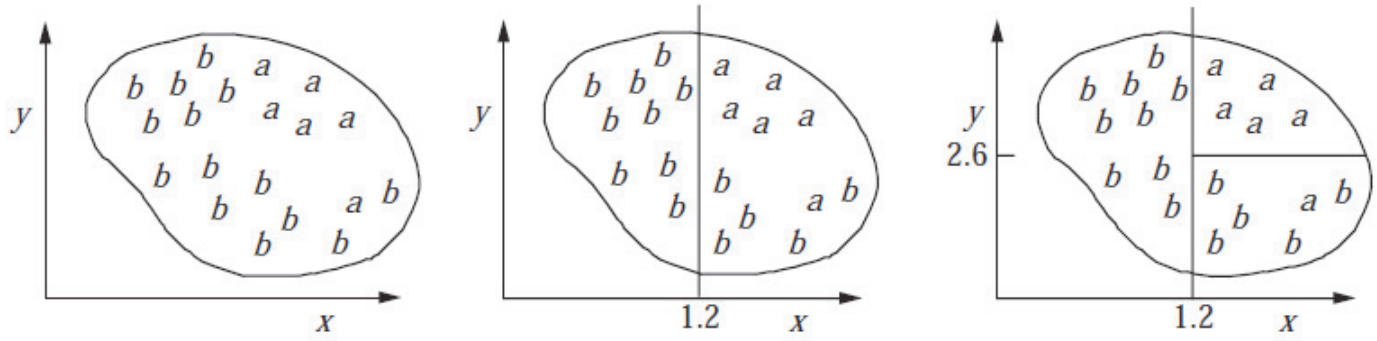
Σύμφωνα με τον ψευδοκώδικα ο PRISM δε σταματά εάν δεν ξεφορτωθεί όλα τα λάθος instances για έναν κανόνα. Βέβαια αυτό θα μπορούσε να δεχθεί τροποποίηση, ώστε οι κανόνες να μην πέφτουν στην παγίδα του overfitting.

Αφού ολοκληρωθεί η κατασκευή των κανόνων για τις κλάσεις, πρέπει να κτιστούν κανόνες και για την περίπτωση κάποιο instance να μην ανήκει σε καμία κλάση, εκτός και αν υπάρχει ένας αρχικός κανόνας για τέτοιες περιπτώσεις.

Προεκτείνοντας στον ολικό αλγόριθμο, δεν υπάρχει κάποια χρονική διάταξη εκτέλεσης των κανόνων μεταξύ των διαφόρων κλάσεων, συνεπώς όλοι οι κανόνες που παράγονται από τον PRISM μπορούν να εκτελεστούν εκτός σειράς (με οποιαδήποτε σειρά δηλαδή), γεγονός που αποτελεί και το βασικό ελάττωμα του PRISM, ελάττωμα κάθε αλγορίθμου που παράγει κανόνες ανεξάρτητους της σειράς εκτέλεσης. Όπως έχουμε ήδη αναλύσει, κανόνες ανεξάρτητοι της σειράς εκτέλεσης δίνουν σαφή ανεξάρτητα "κομμάτια" γνώσης των δεδομένων εισόδου, αλλά δε μπορούν να επιλέξουν κλάση όταν πυροδοτούνται συγχρόνως κανόνες με διαφορετικό συμπέρασμα/κλάση. Είδαμε ήδη ότι οι στρατηγικές αντιμετώπισης του συγκεκριμένου προβλήματος επηρεάζουν αισθητά την απόδοση του αλγορίθμου παραγωγής ανεξάρτητων κανόνων.

Είναι εφικτό να φτιαχτούν λίστες αποφάσεων που ξεπερνούν το παραπάνω πρόβλημα, όπως με χρήση της μεθόδου incremental reduced-error pruning και του κανόνα RIPPER(repeated incremental pruning to produce error reduction),αλλά δε θα τα αναλύσουμε παραπέρα.

Σχήμα 4: Στάδια παραγωγής κανόνα:
Εάν x > 1.2 και y > 2.6 , τότε κλάση = α.



4.5 Γραμμικά Μοντέλα

Άλλα datasets περιλαμβάνουν αριθμητικές τιμές με μία γραμμική εξάρτηση μεταξύ τους,και όπως ήδη ξέρουμε για τέτοιες περιπτώσεις, σημασία δίνεται σε ένα ζυγισμένο άθροισμα τιμών με κατάλληλα επιλεγμένα βάρη. Ενώ το φυσικό πεδίο λειτουργίας των αλγορίθμων παραγωγής δέντρων και κανόνων είναι οι ονομαστικές τιμές , και λειτουργούν σε αριθμητικές με τροποποιήσεις στους ιδίους ή στο dataset, για τα γραμμικά μοντέλα είναι οι αριθμητικές. Και να επαναλάβουμε εδώ, ότι τα γραμμικά μοντέλα λειτουργούν και ως συστατικά πιο σύνθετων μεθόδων μάθησης.

4.5.1 Αριθμητική Πρόβλεψη: Γραμμική Παλινδρόμηση

Όταν η κλάσεις και όλα τα υπόλοιπα attributes είναι αριθμητικά, η πρώτη φυσική επιλογή είναι η μέθοδος της γραμμικής παλινδρόμησης (Linear Regression). Είδαμε ήδη ότι η ιδέα της είναι η έκφραση της κάθε κλάσης ως έναν γραμμικό συνδυασμό από attributes, με συντελεστές προαποφασισμένα βάρη. Συνεπώς όλος ο κόπος βρίσκεται στην εύρεση των βαρών. Αν i ένα από τα n instances του dataset, xi η κλάση του και w τα βάρη των k σε πλήθος attributes του instance αυτού, τότε στη μέθοδο αυτή η πρόβλεψη της κλάσης είναι:

*wo+w1*a1+...+wk*ak = Σk wi*ai , όπου ao=1.*

Σκοπός είναι η εύρεση των (k+1) βαρών wo,w1,...,wk, έτσι ώστε να ελαχιστοποιηθεί το άθροισμα:

$$\sum_{i=1}^{n}\left(x^{(i)}-\sum_{j=0}^{k}w_j a_j^{(i)}\right)^2$$

, όπου η ποσότητα που τετραγωνίζεται είναι η διαφορά της πραγματικής από την προβλεπόμενη τιμή του instance i. Τα βάρη βρίσκονται με τη μέθοδο ελαχίστων τετραγώνων.

Λόγω της γραμμικότητάς της που μπορεί να φανεί ακατάλληλη για δεδομένα που εμφανίζουν μη-γραμμική εξάρτηση, η γραμμική παλινδρόμηση χρησιμοποιείται κυρίως ως μέρος πιο σύνθετων μεθόδων.

4.5.2 Γραμμική Ταξινόμηση: Λογιστική Παλινδρόμηση

Όπως και η γραμμική παλινδρόμηση, έτσι και οποιαδήποτε τεχνική παλινδρόμησης μπορεί να χρησιμοποιηθεί για ταξινόμηση, είτε είναι γραμμική είτε όχι. Η μέθοδος που ακολουθείται είναι να ορίζουμε το xi ενός instance του dataset που ανήκει στην κλάση που εξετάζουμε ίσο με 1, και 0 για τα άλλα, και έπειτα εφαρμόζουμε την επιθυμητή τεχνική παλινδρόμησης,δημιουργώντας (k+1) βάρη ανά κλάση,δηλαδή μία

(γραμμική) σχέση ανά κλάση. Εφαρμόζοντας αυτά τα βήματα ανά κλάση προκύπτουν τόσα σύνολα (k+1) βαρών (τόσες σχέσεις) όσες και οι κλάσεις.

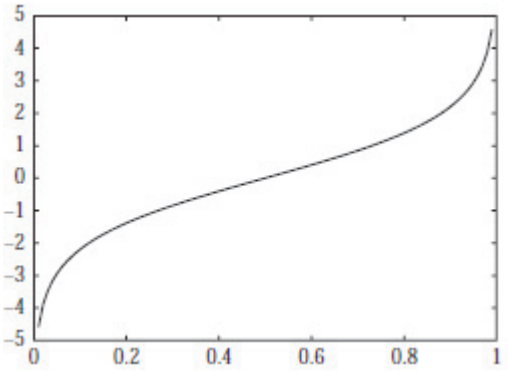
Από τα νέα instances, αυτά προς ταξινόμηση, παίρνουμε τις τιμές των attributes και τα θέτουμε στις σχέσεις των κλάσεων που δημιουργήσαμε. Όποια σχέση δώσει μεγαλύτερο αποτέλεσμα κερδίζει το instance, δηλαδή αυτό ταξινομείται στην κλάση της σχέσης που επιλέχθηκε. Αυτή η μέθοδος λέγεται συχνά multiresponse linear regression. Να σημειώσουμε, ότι εφόσον τα training instances έχουν ως νέες τιμές κλάσης 0 ή 1, συνεπώς και οι σχέσεις για κάθε νέο instance τιμές 0 ή 1 προσπαθούν να προσεγγίσουν.

Στα μειονεκτήματα της μεθόδου αυτής συγκαταλέγεται το γεγονός ότι τα αποτελέσματα που δίνουν ανά instance οι σχέσεις των κλάσεων δεν είναι αληθινές πιθανότητες. Όχι μόνο δεν αθροίζονται με αποτέλεσμα τη μονάδα, αλλά ίσως κάποιο αποτέλεσμα είναι μεγαλύτερό της. Επίσης η μέθοδος των ελαχίστων τετραγώνων υποθέτει ότι τα λάθη είναι και στατιστικά ανεξάρτητα, καθώς και κανονικά κατανεμημένα με την ίδια τυπική απόκλιση, μία υπόθεση που παραβιάζεται κατάφωρα όταν θέτουμε 0 ή 1 ως αληθινή κλάση xi στο κάθε i instance.

Μία συναφής τεχνική είναι η λογιστική παλινδρόμηση (Logistic Regression) , η οποία δεν υποφέρει από τα παραπάνω μειονεκτήματα. Αντί του να προσεγγίζει τα 0 ή 1 κατευθείαν, κτίζει ένα γραμμικό μοντέλο βασισμένο σε μία μετασχηματισμένη μεταβλητή. Ας υποθέσουμε χωρίς βλάβη της γενικότητας ότι υπάρχουν μόνο δύο κλάσεις. Η λογιστική παλινδρόμηση θα αντικαταστήσει την μεταβλητή *Pr[1|a1,...,ak]*, που δεν μπορεί να προσεγγιστεί με ακρίβεια με την γραμμική παλινδρόμηση, με την *log[Pr[1|a1,...,ak] / (1 – Pr[1|a1,...,ak])]*. Η τελευταία μεταβλητή λέγεται λογιστικός μετασχηματισμός.

Οι τιμές του λογιστικού ΜΣ δεν περιορίζονται μεταξύ 0 και 1 , όπως με τις πιθανότητες. Η μετασχηματισμένη μεταβλητή δίνεται ως: *Pr[1|a1,...,ak] = 1/ (1+E)*, όπου
*E = exp (- wo – w1*a1 - ... - wk*ak)*. Προφανώς και εδώ το ζητούμενο είναι να βρεθούν βάρη κατάλληλα, που αν ταιριάζουν στην δομή των δεδομένων εισόδου.

Σχήμα 5: Λογιστικός Μετασχηματισμός



Εδώ η ποσότητα που πρέπει να μεγιστοποιηθεί λέγεται log-likelihood, και δίνεται ως:

$$\sum_{i=1}^{n}(1-x^{(i)})\log(1-\Pr[1\mid a_1^{(1)}, a_2^{(2)}, \ldots, a_k^{(k)}]) + x^{(i)}\log(\Pr[1\mid a_1^{(1)}, a_2^{(2)}, \ldots, a_k^{(k)}])$$

, όπου τα xi είναι είτε 0 είτε 1. Ένας απλός τρόπος για την επίλυση αυτού του προβλήματος μεγιστοποίησης είναι να λύσουμε επαναληπτικά μία ακολουθία ζυγισμένων προβλημάτων ελαχίστων τετραγώνων μέχρις ότου η παραπάνω ποσότητα να συγκλίνει στο μέγιστό της, το οποίο συνήθως συμβαίνει σε μερικές επαναλήψεις.

Για να γενικευτεί η λογιστική παλινδρόμηση σε πολλές κλάσεις, μία δυνατότητα είναι να εφαρμοστεί ανεξάρτητα ανά κλάση, εφαρμόζοντας multiresponse linear regression, η οποία δυστυχώς συνοδεύεται από το πρόβλημα που ήδη αναφέραμε , των ανακριβών πιθανοτικών προσεγγίσεων που δεν αθροίζονται στη μονάδα. Για ακριβείς πιθανότητες, είναι απαραίτητη η κατά ζεύγη αντιμετώπιση των κλάσεων , έχοντας έτσι να επιλύσουμε ένα κοινό πρόβλημα βελτιστοποίησης.

Επιστρέφοντας στις δύο κλάσεις, υπάρχει ένα σύνορο μεταξύ των κλάσεων, το οποίο συσχετίζει όλα τα attributes, και δίνεται για *Pr[1|a1,...,ak] = 1/ (1+E) = 0.5*, το οποίο ισχύει για :

*- wo − w1*a1 - ... - wk*ak = 0 => για wo + w1*a1 + ... + wk*ak = 0.*

Το σύνορο *wo + w1*a1 + ... + wk*ak = 0* αποτελεί μία γραμμική σχέση των attributes, και συνεπώς αποτελεί ένα υπερεπίπεδο. Ως μειονέκτημα λοιπόν της λογιστικής παλινδρόμησης πρέπει να αναφερθεί ότι πάντα μπορούν να βρεθούν instances που δεν γίνεται να διαχωριστούν με χρήση ενός επιπέδου. Το ίδιο πρόβλημα αντιμετωπίζει και η multiresponse linear regression με γραμμική παλινδρόμηση. Για δύο κλάσεις ένα νέο instance θα ανήκει στην πρώτη μόνο εάν:

*w1o + w11*a1 + ... + w1k*ak > w2o + w21*a1 + ... + w2k*ak =>*

*=> για (w1o-w2o) + (w11-w21)*a1 + ... + (w1k-w2k)*ak > 0* , όπου πάλι προκύπτει ότι το σύνορο *(w1o-w2o) + (w11-w21)*a1 + ... + (w1k-w2k)*ak = 0* μεταξύ των δύο κλάσεων είναι υπερεπίπεδο.

4.5.3 Γραμμική Ταξινόμηση με Χρήση του Perceptron

Ενώ τα παραπάνω επαρκούν για τη σωστή ταξινόμηση ενός νέου instance, δεν είναι απαραίτητο να προβούμε σε υπολογισμό πιθανοτήτων όταν το ζητούμενο είναι ένα σύνορο που διαχωρίζει τις κλάσεις. Εάν τα δεδομένα είναι γραμμικώς διαχωρίσιμα, δηλαδή εάν μπορούν να διαχωριστούν τέλεια σε δύο ομάδες με χρήση ενός υπερεπιπέδου, τότε υπάρχει ένας απλός αλγόριθμος για την εύρεση αυτού του υπερεπιπέεδου-συνόρου, ο perceptron κανόνας μάθησης.

Αν για ένα νέο instance το άθροισμα *wo + w1*a1 + ... + wk*ak* ισούται με μηδέν, τότε ανήκει στην δεύτερη ομάδα, αλλιώς αν είναι μεγαλύτερο του μηδενός ανήκει στην 1η ομάδα,δηλαδή στην 1η κλάση. Παρατίθεται ο ψευδοκώδικας του αλγορίθμου:

- *Θέσε όλα τα βάρη ίσα με μηδέν*
- *Μέχρις ότου όλα τα training instances να είναι σωστά ταξινομημένα*
- *Για κάθε training instance I*
- *Εάν το I δεν είναι σωστά ταξινομημένο από το perceptron*
- *Εάν το I ανήκει στην 1η κλάση πρόσθεσέ το στο διάνυσμα βαρών*
- *αλλιώς αφαίρεσέ το από το διάνυσμα βαρών*

Ο αλγόριθμος κάνει επαναλήψεις μέχρι να βρεθεί μία τέλεια λύση, αλλά θα δουλέψει σωστά μόνο εάν τα δεδομένα είναι γραμμικώς διαχωρίσιμα. Σε κάθε επανάληψη εξετάζονται όλα τα training instances. Εάν κάποιο εξ αυτών βρίσκεται από την λάθος μεριά του συνόρου,δηλαδή η περιοχή στην οποία βρίσκεται αντιστοιχεί στην "αντίπαλη" κλάση από αυτήν στην οποία όντως ανήκει το training instance, τότε τα βάρη του συνόρου αλλάζουν καταλλήλως ώστε το instance αυτό να προσεγγίσει το σύνορο,ή και να το διασχίσει προς τη σωστή μεριά.

Η τροποποίηση των βαρών γίνεται ως εξής: στο σύνορο προστίθενται οι τιμές του υπό εξέταση instance, όταν βρίσκεται στη μεριά της 2ης ομάδας αλλά ανήκει και πρέπει να πάει στη μεριά της 1ης κλάσης. Άρα για το instance i το σύνορο γίνεται:
*(wo+aio) + (w1+ai1)*a1 + ... + (wk+aik)*ak.* Όταν όμως εξετάζω το instance i ισχύει *ai1=ai, ai2=a2,...,aik=ak* *=>* το αποτέλεσμα της γραμμικής ποσότητας *(wo + w1*a1 + ... + wk*ak)* έχει αυξηθεί κατά *aio*aio + ai1*ai1 + ... + aik*aik > 0* , άρα το σύνορο έχει μετακινηθεί προς την σωστή κατεύθυνση. Αντιθέτως, για instance που πρέπει να πάει από την 1η στη 2η ομάδα τροποποιώ τα βάρη αφαιρώντας, και προκύπτει μείωση στο αποτέλεσμα της γραμμικής ποσότητας κατά

*- aio*aio - ai1*ai1 - ... - aik*aik < 0* , άρα και πάλι το σύνορο έχει μετακινηθεί προς την σωστή κατεύθυνση.

Αυτές οι διορθώσεις γίνονται στα όρια, και μπορεί να παρεμβαίνουν σε προηγούμενες διορθώσεις, ωστόσο μπορεί να αποδειχθεί ότι ο αλγόριθμος συγκλίνει μετά από πεπερασμένο πλήθος επαναλήψεων εάν τα δεδομένα είναι γραμμικώς διαχωρίσιμα. Εάν δεν είναι, ο αλγόριθμος δεν τερματίζει, συνεπώς πρέπει να τεθεί ένα άνω όριο πλήθους επαναλήψεων.

4.5.4 Γραμμική Ταξινόμηση με Χρήση του Winnow

Για datasets με δυαδικά attributes (να παίρνουν τιμές 0 ή 1) υπάρχει μία εναλλακτική μέθοδος γνωστή ως Winnow. Παρατίθεται ο ψευδοκώδικας του αλγορίθμου, μία φορά στην απλή εκδοχή και έπειτα στην ισορροπημένη:

o Όσο κάποια instances είναι λάθος ταξινομημένα
o για κάθε instance E
o ταξινόμησε το E με τα παρόντα βάρη
o εάν η κλάση πρόβλεψης είναι λάθος
o εάν το E ανήκει στην 1η κλάση
o για κάθε ai attribute που είναι 1,
o πολλαπλασίασε το wi με Z
o (αν ai == 0, wi δεν αλλάζει)
o αλλιώς
o για κάθε ai attribute που είναι 1,
o διαίρεσε το wi με Z
o (αν ai == 0, wi δεν αλλάζει)
o (Απλή μορφή)

Όπως και ο perceptron κανόνας , ο Winnow ενημερώνει τα βάρη, όταν εντοπίζεται λανθασμένα ταξινομημένο instance, καθοδηγείται δηλαδή από τα λάθη. Η διαφορά τους έγκειται στο πώς αλλάζει τα βάρη ο καθένας. Ο perceptron κάνει προσθαφαιρέσεις. Ο Winnow πολλαπλασιασμούς και διαιρέσεις. Ο συντελεστής Z ορίζεται από τον χρήστη και πρέπει Z > 1. Τα βάρη αρχικοποιούνται την πρώτη φορά σε κάποιες σταθερές που ορίζονται πάλι από τον χρήστη. Το όριο θ του συνόρου δεν είναι το μηδέν,αλλά ορίζεται από τον χρήστη: Ένα instance ταξινομείται στην 1η κλάση όταν *wo + w1*a1 + ... + wk*ak > θ* για αυτό το instance. Προφανώς τα βάρη δεν επηρεάζονται από τα attributes με μηδενική τιμή.

o *Όσο κάποια instances είναι λάθος ταξινομημένα*

o *για κάθε instance E*

o *ταξινόμησε το E με τα παρόντα βάρη*

o *εάν η κλάση πρόβλεψης είναι λάθος*

o *εάν το E ανήκει στην 1η κλάση*

o *για κάθε ai attribute που είναι 1,*

o *πολλαπλασίασε το (wi+) με Z*

o *διαίρεσε το (wi-) με Z*

o *(αν ai == 0, wi δεν αλλάζει)*

o *αλλιώς*

o *για κάθε ai attribute που είναι 1,*

o *πολλαπλασίασε το (wi-) με Z*

o *διαίρεσε το (wi+) με Z*

o *(αν ai == 0, wi δεν αλλάζει)*

o *(Ισορροπημένη μορφή)*

Στην απλή μορφή, ο Winnow δεν επιτρέπει αρνητικά βάρη, που ίσως σε κάποιες περιπτώσεις να είναι μειονέκτημα. Η ισορροπημένη μορφή του επιτρέπει αρνητικούς συντελεστές στη γραμμική σχέση, με χρήση των βαρών (wi+),(wi-). Ένα instance ταξινομείται στην 1η κλάση όταν
*((wo+) - (wo-)) + ((w1+) - (w1-))*a1 + … + ((wk+) - (wk-))*ak > θ* για αυτό το instance.

Ο Winnow είναι πολύ αποτελεσματικός στο να στοχεύει στα σχετικά χαρακτηριστικά σε ένα dataset. Είναι χρήσιμος π.χ. όταν ένα dataset έχει πολλά (binary) attributes και τα περισσότερα εξ αυτών είναι αδιάφορα. Πάντως και ο Winnow και ο Perceptron μπορούν να χρησιμοποιηθούν για ταξινόμηση νέων instances σε ζωντανό χρόνο, επειδή μπορούν προσθετικά και οριακά να ενημερώνουν το υπερεπίπεδο που χωρίζει τις κλάσεις.

4.6 Αναπαράσταση με Βάση τα Instances

Υπάρχουν datasets που όταν οι ταξινομήσεις ενδείκνυνται να γίνονται σε συγκεκριμένες περιοχές τιμών των attributes των instances, οι ταξινομήσεις αυτές εξαρτώνται από τις αποστάσεις μεταξύ των ίδιων των instances. Μιλάμε για instance-based learning, και όπως ήδη έχουμε αναφέρει, σε αυτό το είδος μάθησης τα training instances αποθηκεύονται αυτολεξεί, και μία συνάρτηση υπολογισμού απόστασης αποφασίζει ποιο μέλος του training συνόλου βρίσκεται πιο κοντά στο νέο test instance. Μόλις βρεθεί, η κλάση του δίνεται ως πρόβλεψη για το test instance.

4.6.1 Συνάρτηση Απόστασης

Για την απόσταση αριθμητικών τιμών , ήδη μιλήσαμε για την ευκλείδεια απόσταση στο ρόλο συνάρτησης απόστασης (Distance Function): δηλαδή την τετραγωνική ρίζα του αθροίσματος των τετραγώνων των διαφορών των τιμών των attributes από δύο instances. Η σύγκριση των ευκλείδειων αποστάσεων μπορεί να γίνει κατευθείαν στα αθροίσματα χωρίς να υπολογιστεί η ρίζα. Εναλλακτικά,αντί για δύναμη του 2 , μπορούμε να έχουμε μονάδα, καταλήγοντας στην απόσταση Manhattan (αλλά με απόλυτες τιμές των διαφορών), ή και μεγαλύτερη του 2. Υψηλότερες δυνάμεις ενισχύουν την επίδραση των μεγαλύτερων διαφορών στο άθροισμα σε σχέση με τις μικρότερες. Γενικά η ευκλείδεια απόσταση είναι ένας καλός συμβιβασμός.

Τα διάφορα attributes συνήθως βρίσκονται σε διαφορετική κλίμακα. Προαπαιτείται λοιπόν κανονικοποίηση στις τιμές τους, και συνηθίζεται οι νέες τιμές να βρίσκονται μεταξύ 0 και 1.
Συνεπώς vi_nor = [vi - min{vi}] / [max{vi} - min{vi}] , όπου vi είναι η κανονική τιμή του i attribute και vi_nor η κανονικοποιημένη. Τα max και min βρίσκονται μεταξύ όλων των εμφανιζόμενων τιμών στο training dataset για αυτό το attribute.

Για τις ονομαστικές τιμές, οι διαφορές μεταξύ των τιμών των attributes είναι συνήθως 1 για διαφορετικές τιμές σε ένα attribute και 0 για ίδιες. Έχοντας έτσι τις διαφορές ως ποσότητες 0 και 1, μπορώ να τις θέσω στον τύπο της ευκλείδειας απόστασης. Προφανώς στις ονομαστικές τιμές δε χρειάζεται κανονικοποίηση.

Στις ονομαστικές τιμές έστω και μία τιμή να λείπει σε μία διαφορά, η διαφορά τίθεται ίση με 1. Για αριθμητικές τιμές , αν λείπουν και οι δύο η διαφορά τίθεται 1. Αν λείπει η μία μόνο, η διαφορά τίθεται είτε ίση με την κανονικοποιημένη τιμή της τιμής που δε λείπει, είτε ως 1 μείον την κανονικοποιημένη τιμή της τιμής που δε λείπει, διαλέγοντας το μεγαλύτερο εκ των δύο. Γενικότερα δηλαδή, όταν λείπει μία τιμή, η διαφορά των τιμών τίθεται όσο μεγαλύτερη γίνεται να τεθεί.

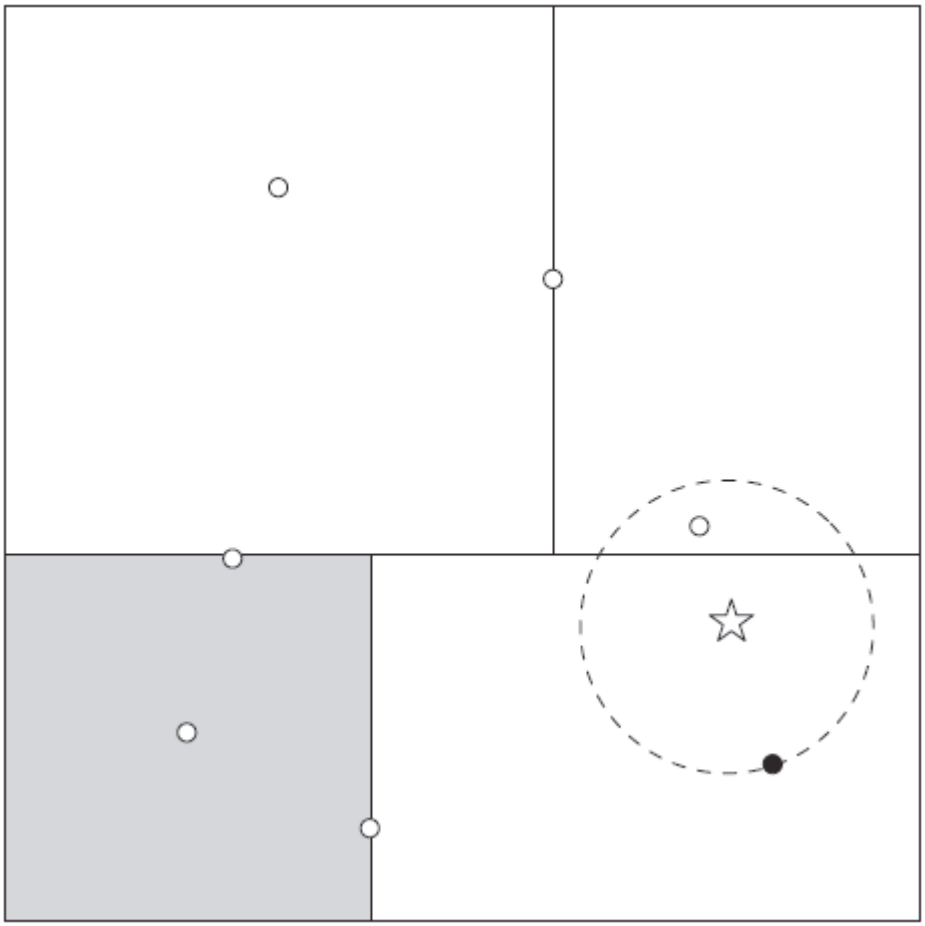
4.6.2 Εύρεση Κοντινότερου Γείτονα Αποδοτικά

Τα υπερεπίπεδα δεν αποτελούν σύνορα αποφάσεων. Οι αποφάσεις ταξινόμησης λαμβάνονται σύμφωνα με τον κανόνα του κοντινότερου γείτονα. Υποθέτοντας για αρχή k=2 χωρίς βλάβη της γενικότητας, δηλαδή ότι έχουμε μόνο 2 attributes, και ότι ήδη ξέρουμε πώς θα αναπτυχθεί το δέντρο, η δομή του μπορεί να μεταφερθεί σε ένα επίπεδο 2 διαστάσεων με χ,ψ άξονες. Καθώς το δέντρο είναι δυαδικό, οι διαχωρισμοί των attributes αντιστοιχούν σε κάθετες στους άξονες τομές. Κάθε τομή θα διασχίζει ένα training instance. Training instances που δεν έχουν κάποιο attribute με τιμή που αποτελεί σημείο διαχωρισμού του αντίστοιχου άξονα αποτελούν φύλλα του δέντρου. Η τελική εικόνα του επιπέδου είναι ένα σύνολο από

εφαπτόμενα ορθογώνια, με τα φύλλα στο εσωτερικό τους και τα instances διαχωρισμού στις πλευρές. Λόγω της φύσεως των δυαδικών δέντρων, έως και τα μισά φύλλα μπορούν να είναι φύλλα.

Προτού δοθεί η μέθοδος παραγωγής ενός kD- δέντρου, παρατίθεται η χρησιμότητά τους στην βελτίωση των υπολογισμών του nearest-neighbor κανόνα. Ας υποθέσουμε την περίπτωση με training instances με k=2 όπως στο σχήμα:

Σχήμα 6: Περιοχές kD-tree με k=2



Το νέο instance προς ταξινόμηση σημειώνεται με αστέρι. Ο κόμβος-φύλλο στου οποίου την περιοχή ανήκει το αστέρι σημειώνεται με μαύρο χρώμα.

Γενικότερα η τακτική εύρεσης του κοντινότερου γείτονα απαιτεί πρώτα την πορεία στο δέντρο από την ρίζα προς το φύλλο που εμπεριέχει στην περιοχή του το αστέρι. Το μαύρο δεν είναι ο πιο κοντινός γείτονας του αστεριού, αλλά είναι μία καλή πρώτη προσέγγιση.

Δεν ξέρουμε εκ των προτέρων εάν υπάρχει κοντινότερος γείτονας από το μαύρο, συνεπώς θα ψάξουμε το δέντρο ξεκινώντας από την περιοχή του άλλου παιδιού του κόμβου γονέα του φύλλου στην περιοχή του οποίου βρίσκεται το αστέρι. Στο σχήμα ο κόμβος γονέας φαίνεται γιατί ακουμπάει πάνω στην πλευρά

του τετραγώνου του μαύρου. Η δε θέση του μαρτυρά το άλλο παιδί του γονέα, του οποίου η περιοχή στο σχήμα έχει γκρίζο χρώμα. Εφόσον εδώ δεν βρέθηκε τέτοιος γείτονας στο άλλο παιδί, εξετάζουμε το υποδέντρο κάτω από τον αδελφικό κόμβο του τωρινού γονέα , κ.ό.κ. .

Γενικά ο αλγόριθμος αυτός είναι γρηγορότερος από το να ελεγχθούν σειριακά όλα τα training instances σε σχέση με το αστέρι. Η εύρεση της πρώτης εκτίμησης (μαύρο σημείο σχήματος) εξαρτάται από το ύψος του δέντρου, και για ένα ισορροπημένο δέντρο χρειάζονται log(ύψος) βήματα (βάση 2). Το κόστος υπολογισμού του κοντινότερου γείτονα εξαρτάται από την δομή του δέντρου και από το πόσο καλή ήταν η πρώτη εκτίμηση. Για ένα καλώς κατασκευασμένο δέντρο, δηλαδή για ένα δέντρο που οι περιοχές τείνουν να είναι τετράγωνα παρά μακρόστενα ορθογώνια, και για dataset που δεν έχει πάρα πολλά attributes, τότε το κόστος υπολογισμού του κοντινότερου γείτονα είναι λογαριθμικό ως προς το πλήθος των κόμβων του δέντρου.

Όσον αφορά την κατασκευή του δέντρου, το πρόβλημα συνοψίζεται στην εύρεση του πρώτου κοψίματος σε κάποιο training instance. Έπειτα εφαρμόζεται η ίδια διαδικασία αναδρομικά σε κάθε επιμέρους κομμάτι μετά την διάσπαση. Η κατεύθυνση ενός κοψίματος μπορεί να επιλεχθεί βάσει του κατά μήκους ποίου άξονα(εξετάζοντάς τον ανεξάρτητα) η διακύμανση των τιμών είναι μεγαλύτερη. Έπειτα τον διαχωρίζεις κάθετα με ένα υπερεπίπεδο. Το σημείο τομής του υπερεπιπέδου στον άξονα βρίσκεται ως εξής: είτε επιλέγεις το ενδιάμεσο σημείο κατά μήκος του άξονα ,δηλαδή το σημείο που εκατέρωθέν του θα έχει ίσες ποσότητες από άλλα σημεία, είτε βρίσκεις τον μέσο όρο των τιμών των σημείων κατά μήκους του άξονα και επιλέγεις το σημείο με τιμή πιο κοντά στον μέσο όρο.

Εάν η κατανομή των σημείων στο χώρο είναι ομοιόμορφη, τότε η χρήση ενδιάμεσου σημείου για το κόψιμο ενδείκνυται, καθώς και ισομοιράζονται τα σημεία από κάθε μεριά του υπερεπιπέδου λόγω επιλογής ενδιάμεσου σημείου και αλλάζει η κατεύθυνση (άξονα) κοπής πολύ συχνά λόγω της καλής κατανομής, παράγοντας έτσι ισορροπημένα δέντρα. Αν όμως η κατανομή είναι στρεβλή, τότε με το ενδιάμεσο σημείο ίσως δεν αλλάζει συχνά ο άξονας κοπής, με αποτέλεσμα μακρόστενες περιοχές (αλλά παραμένουν ισορροπημένα τα δέντρα λόγω επιλογής ενδιάμεσου σημείου). Αντιθέτως με την πιο ευφυή στρατηγική της χρήσης μέσου όρου επιτυγχάνεται και καλός διαμοιρασμός σημείων εκατέρωθεν των τομών, και συχνή εναλλαγή του άξονα κοπής ,δηλαδή δέντρα όχι εντελώς ισορροπημένα, αλλά αποδεκτά.

Ένα πλεονέκτημα της instance-based μηχανικής μάθησης σε σύγκριση με τους περισσότερους άλλους αλγορίθμους, είναι ότι νέα training instances μπορούν να προστεθούν οποιαδήποτε στιγμή. Για να διατηρήσουμε αυτό το πλεονέκτημα στα kD-trees, πρέπει για κάθε νέο instance να κοιτούμε σε ποία περιοχή βρίσκεται. Αν εκείνη είναι κενή, απλώς πρόσθεσέ το. Αν δεν είναι κενή, κόψε την περιοχή στα δύο, τέμνοντας την μεγάλη της πλευρά. Έτσι φυσικά δεν διατηρείται η ισορροπία του δέντρου, ούτε ότι τα

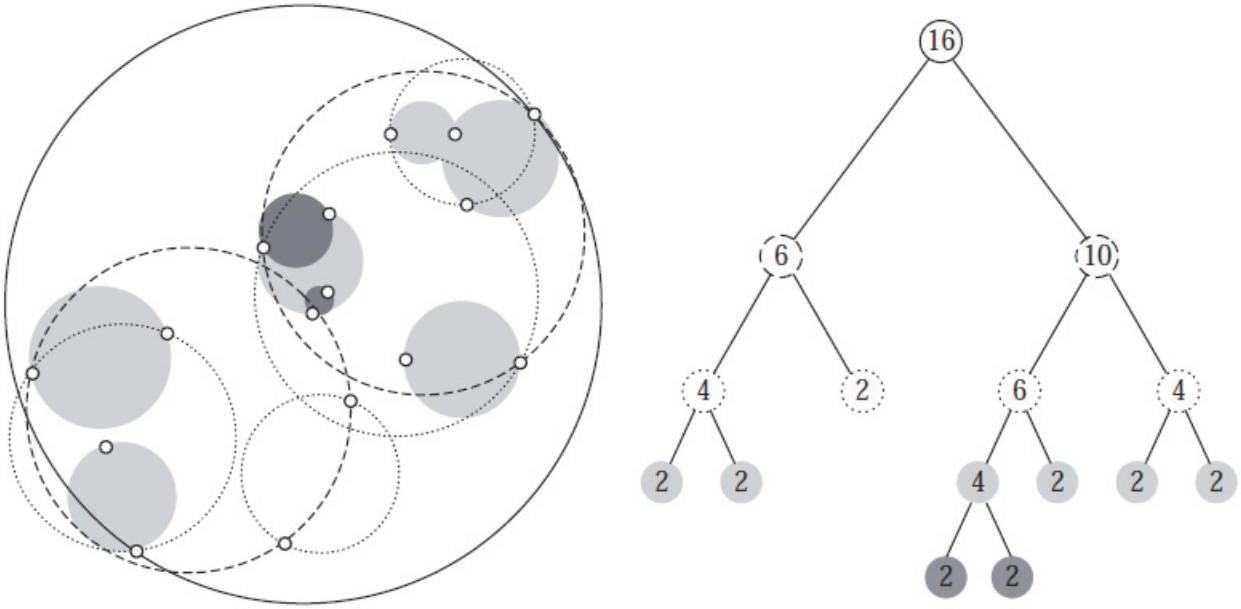
υπερεπίπεδα θα είναι κατάλληλα σχηματισμένα για να εφαρμόζεται αναζήτηση κοντινότερου γείτονα. Είναι σωστή τακτική, όταν το δέντρο ξεπερνά το διπλάσιο του ύψους που θα είχε εάν ήταν ισορροπημένο, να δημιουργείται από την αρχή.

Βασικό μειονέκτημα των kD-δέντρων είναι όπως είδαμε ο συμβιβασμός που πρέπει να γίνει μεταξύ ισορροπημένου δέντρου και τετράγωνων περιοχών όταν έχουμε στρεβλή κατανομή σημείων στον υπερχώρο k διαστάσεων. Εξίσου σημαντικό είναι το γεγονός ότι ως σχήμα τα ορθογώνια δημιουργούν προβλήματα με τις εφαπτόμενες γωνίες τους. Κοντά σε μία γωνία εφάπτονται και άλλες, και αυξάνονται έτσι οι περιοχές αναζήτησης κοντινότερου γείτονα.

Το τελευταίο πρόβλημα επιλύεται με τη δημιουργία περιοχών υπερσφαιρών αντί για υπερορθογωνίων. Μια δομή “ball” tree ορίζει k-διάστατες υπερσφαίρες(balls) , οι οποίες καλύπτουν τα δεδομένα, και τις οργανώνει σε ένα δέντρο. Κάθε σφαίρα μπορεί να εμπεριέχει πολλά σημεία-instances και να επικαλύπτεται με άλλες σφαίρες. Όσο βρισκόμαστε σε μεγαλύτερο βάθος στο δέντρο, οι σφαίρες-κόμβοι έχουν μικρότερη ακτίνα και περιέχουν λιγότερα σημεία από τους κόβους-προγόνους τους. Κάθε κόμβος-σφαίρα διατηρεί το κέντρο του και την ακτίνα του αποθηκευμένα. Κόμβοι-φύλλα διατηρούν και τα σημεία-instances που εμπεριέχουν. Σε κάθε επίπεδο του δέντρου, κάθε σημείο μπορεί αντιστοιχίζεται σε μία μόνο σφαίρα από τις επικαλυπτόμενες στις οποίες το εμπεριέχουν.

Η αναζήτηση κοντινότερου γείτονα στα ball-trees γίνεται ακριβώς όπως και στα kD-trees με ορθογώνια. Το μόνο που απομένει είναι η μέθοδος με την οποία κανείς διαχωρίζει μία σφαίρα σε άλλες. Καταρχάς όπως και στα kD-trees με ορθογώνια, έτσι και στα kD-ball trees, δεν είναι απαραίτητο να αφήνεται ο διαχωρισμός να φτάνει έως και τη σύγκριση δύο τελευταίων στοιχείων. Μπορεί να επιλεχθεί εξ αρχής ένα όριο πλήθους σημείων που δεν συγκρίνονται περεταίρω. Ο διαχωρισμός γίνεται ως εξής: Διαλέγεται το πιο απομακρυσμένο σημείο από το κέντρο της σφαίρας, και έπειτα το πιο απομακρυσμένο από αυτό. Τα υπόλοιπα αντιστοιχίζονται στο κοντινότερο εκ των δύο. Έπειτα υπολογίζεται το κέντρο βάρους κάθε ομάδας και εβρίσκεται μία ελάχιστη ακτίνα ανά ομάδα που να περικυκλώνει όλα τα σημεία κάθε ομάδας. Το κόστος της μεθόδου σε υπολογισμούς είναι γραμμικό του πλήθους των σημείων προς διαχωρισμό σε ομάδες. Υπάρχουν και πιο σύνθετες μέθοδοι διαχωρισμού των σφαιρών που δε θα αναλυθούν.

Σχήμα 7: Ball-tree για k=2 και 16 training instances



5 Αξιολόγηση των Αλγορίθμων

Οι αλγόριθμοι χρησιμοποιούνται για την πρόβλεψη των κλάσεων των instances. Κατά την αξιολόγησή τους όμως τα instances έχουν ήδη την αληθινή τιμή της κλάσης, ώστε να μπορούμε να ελέγξουμε την απόδοσή τους. Ο κλασσικότερος τρόπος είναι η δημιουργία ενός απλού ποσοστού: πόσες σωστές προβλέψεις έχει εκτελέσει ο αλγόριθμος προς τις όλες. Στην παρούσα διπλωματική εργασία, θα περιοριστούμε σε αυτή τη μορφή αξιολόγησης, αν και δεν είναι σε όλες τις εφαρμογές η σημαντικότερη, ή η μόνη σημαντική.

Παρουσιάστηκε παραπάνω ο διαχωρισμός σε training set και testing set, και ποία είναι η λειτουργία τους. Συχνά όμως, όπως και εδώ, αυτά δεν δίνονται χώρια, αλλά ως ένα ενιαίο dataset. Όταν αυτό συμβαίνει , με έναν ποσοστιαίο διαχωρισμό, το λεγόμενο percentage split, μπορεί κανείς να διαχωρίσει το ενιαίο dataset στα επιμέρους training και test. Τυπικές τιμές είναι τα 66% και 70% , όπου υποδηλώνουν το ποσοστό του αρχικού dataset που γίνεται training set.

Ένας ευφυέστερος τρόπος αξιολόγησης, είναι το cross-validation, η σταυρωτή αξιολόγηση , όπου δίνει κανείς ένα ποσοστό το οποίο αντιπροσωπεύει το κάθε φορά ποσοστό επί του όλου, του testing set. Από πληθώρα data mining εφαρμογών προκύπτει ιδανική τιμή όσον αφορά την ποιότητα της εκτίμησης λάθους, άρα και της απόδοσης του αλγορίθμου, ένα ποσοστό 10% για το training set. Συνεπώς μπορούν να προκύψουν 10 μη-επικαλυπτόμενα testing sets από το αρχικό dataset, και κάθε φορά ο αλγόριθμος προπονείται στο 90%. Ως τελική απόδοση του αλγορίθμου δίνεται ο μέσος όρος του ποσοστού επιτυχίας των 10 ταξινομήσεων.

6 Ανάλυση Συναισθήματος στα Κοινωνικά Δίκτυα

Η ανάλυση συναισθήματος στα κοινωνικά δίκτυα (Social Media) φιλοξενεί την εξαγωγή χρήσιμων συμπερασμάτων σχετικά με τη μέση κοινή γνώμη σε μια ποικιλία θεμάτων, αλλά παραβάλει σοβαρές τεχνικές προκλήσεις. Αυτό συμβαίνει εξαιτίας του αραιού,θορυβώδους, πολυγλωσσικού περιεχομένου που αναρτάται on-line από τους χρήστες των κοινωνικών δικτύων.

Οι γραπτές εκφράσεις συναισθημάτων δηλώνουν είτε τη διάθεση των συντακτών τους, είτε την άποψή τους σε σχέση με μια συγκεκριμένη οντότητα. Το διαδίκτυο αφθονεί σε τέτοιες εκφράσεις , καθώς οι τεχνολογίες του Web 2.0 επιτρέπουν σε κοινούς χρήστες να σχολιάζουν και να αναρτούν διαδικτυακά τις σκέψεις τους για οτιδήποτε.

Πλέον συνηθισμένη , ωστόσο, είναι η έκφραση προσωπικών συναισθημάτων μέσω Υπηρεσιών Κοινωνικής Δικτύωσης (Social Networking Services (SNSs)). Δίκτυα όπως το Facebook και το Twitter έχουν αποκτήσει πρόσφατα ένα τεράστιο μερίδιο της συνολικής δραστηριότητας του διαδικτύου, επιτρέποντας στους χρήστες τους να συζητούν καθημερινά θέματα, να ανταλλάζουν πολιτικές απόψεις , και να αξιολογούν υπηρεσίες και προϊόντα.

Η εκτενής χρήση συναισθηματικών εκφράσεων οδήγησε στην ανάπτυξη εξειδικευμένων συμβολισμών που σηματοδοτούν ένα συναίσθημα, ονόματι emoticons (π.χ., “:)” και “:(“ για θετικά και αρνητικά συναισθήματα αντίστοιχα. Στο ίδιο πνεύμα, οι πλατφόρμες κοινωνικής δικτύωσης ανέπτυξαν ειδικές λειτουργίες για να υποστηρίξουν την έκφραση απόψεων• το κουμπί “Like” του Facebook αποτελεί το πιο χαρακτηριστικό παράδειγμα τέτοιων λειτουργιών.

Οι υπάρχουσες τεχνικές συνήθως βασίζονται στην αναγνώριση μοτίβων σε ελεύθερο κείμενο που εκφράζει ένα συναίσθημα. Αυτά τα πρότυπα αφορούν είτε διακριτές (σειρές από) λέξεις , είτε ν-γράμματα χαρακτήρων. Οι προηγούμενες μέθοδοι βασίζονται στο μοντέλο διανυσμάτων από όρους (term vectors), ενώ οι τελευταίες στο μοντέλο ν-γραμμάτων .

6.1 Εγγενή Χαρακτηριστικά των Κοινωνικών Δικτύων

Παρόλο που επιτυγχάνουν υψηλές επιδόσεις στο περιεχόμενο συγκεκριμένων ρυθμίσεων, οι τωρινές μέθοδοι είναι ανεπαρκείς στο να διαχειρίζονται τα δεδομένα που παράγονται από χρήστες των Social Media. Ο λόγος είναι ότι οι υπάρχουσες τεχνικές είναι ανά γλώσσα συγκεκριμένες: είναι κατασκευασμένες για μία συγκεκριμένη γλώσσα, συνήθως εξαρτώμενες από ένα λεξικό (όπως το WordNet) για την εκτίμηση του νοήματος ή της λεξικολογικής κατηγορίας συγκεκριμένων λέξεων ή φράσεων. Το περιεχόμενο των κειμένων των Social Media, ωστόσο, καταρρίπτει τη θεμελιώδη υπόθεση αυτών των μεθόδων,

εξαιτίας των εγγενών του χαρακτηριστικών:

1. Πολυγλωσσία:

Παρόλο που η πλειοψηφία των χρηστών τους προέρχεται από αγγλόφωνες χώρες, τα SNSs δημοφιλή παγκοσμίως. Η βάση χρηστών τους περιλαμβάνει,ως εκ τούτου, οι άνθρωποι να μιλούν σε πολλές γλώσσες και διαλέκτους.

2. Αργκό και Νεολογισμοί:

Το περιεχόμενο των κειμένων των Social Media είναι μάλλον ανεπίσημο, καθώς περιλαμβάνει κυρίως επικοινωνία μεταξύ φίλων. Ως εκ τούτου οι χρήστες χρησιμοποιούν λέξεις και εκφράσεις που δε θεωρούνται πρότυπες σε καμία διάλεκτο ή γλώσσα. (π.χ. "koo" αντί για "cool"). Επιπροσθέτως , το περιορισμένο μέγεθος των μηνυμάτων τους (ο Twitter π.χ. επιτρέπει μηνύματα έως και 140 χαρακτήρες) τους παροτρύνει να συντομεύουν τις λέξεις σε νέες μορφές, οι οποίες δεν φέρουν μεγάλες ομοιότητες με τις αρχικές. Παραδείγματος χάριν, το "gr8" χρησιμοποιείται συχνά αντί του "great", και το "congratz" αντί του "congratulations".

3. Θόρυβος:

Η πραγματικού χρόνου φύση των Social Media ενθαρρύνει τους χρήστες να αναρτούν τα μηνύματά τους γρήγορα, χωρίς να επαληθεύουν την ορθότητά τους σε σχέση με το νόημα, καθώς και με τη γραμματική και τους συντακτικούς κανόνες. Σε περίπτωση που ένα μήνυμα (ή ένα μέρος του) δεν είναι κατανοητό, ο συγγραφέας του το αντικαθιστά με ένα νέο. Αυτή είναι η περίπτωση, για παράδειγμα, των διαδικτυακών συνομιλιών. Ως αποτέλεσμα , το περιεχόμενο που προέρχεται από τον χρήστη αφθονεί σε λέξεις με ορθογραφικά λάθη και εσφαλμένες χρήσεις φράσεων.

Όλες αυτές οι πρακτικές αναπόφευκτα κάνουν το έργο της ανάλυσης προτύπων πιο σύνθετο, και να καλεί για μια ουδέτερη γλωσσών μέθοδο που είναι ανεκτική στο θόρυβο.

Στην [1] δημοσίευση, προτείνεται η χρήση νέου μοντέλου παρουσίασης εγγράφων για το έργο της ανάλυσης συναισθήματος βασισμένης στο περιεχόμενο, ονόματι n-gram graphs (γράφοι ν – γραμμάτων). Βελτιώνει το μοντέλο ν – γραμμάτων χαρακτήρων προσθέτοντας εννοιολογική πληροφορία: αντί να παράγει ένα απλό σύνολο από ν-γράμματα , λαμβάνει υπόψιν τη σειρά εμφάνισης για να διαμορφώσει ένα έγγραφο ως γράφο. Οι κόμβοι του αντιστοιχούν σε συγκεκριμένα ν-γράμματα, και οι ακμές που τους συνδέουν σηματοδοτούν πόσο κοντά βρίσκονται κατά μέσο όρο στο δοσμένο έγγραφο.

Συνεπώς οι n-gram γράφοι συλλαμβάνουν περισσότερη πληροφορία από τα ν-γράμματα, δίχως να κάνουν κάποια υπόθεση σχετικά με τη γλώσσα των δοσμένων εγγράφων. Αυτό οδηγεί σε μεγαλύτερη αποτελεσματικότητα,σε συνδυασμό με μεγαλύτερη απόδοση: το πλήθος των χαρακτηριστικών που φέρουν δε βασίζεται στην ποικιλομορφία του λεξιλογίου των δοσμένων κειμένων. Απεναντίας, τα χαρα-

κτηριστικά τους βασίζονται αποκλειστικά στο πλήθος των κατηγοριών πολικότητας, με κάθε κατηγορία (κλάση,class) να εισάγει τρεις διαφορετικές μετρικές ομοιότητας. Έτσι,σε αντίθεση με τα άλλα μοντέλα αναπαράστασης, το μοντέλο των n-gram γράφων δεν υποφέρει από το πρόβλημα της διάστασης.

6.2 Χαρακτηριστικά του Κοινωνικού Δικτύου Twitter

Η δημοσίευση [1] επικεντρώνεται στην micro-blogging υπηρεσία του Twitter ως πεδίο δοκιμών για την αξιολόγηση της προσέγγισής. Υπάρχουν αρκετοί λόγοι για την επιλογή αυτή: Πρωτίστως συνεπάγεται από αυστηρούς κανόνες για κοινωνική αλληλεπίδραση, καθώς περιλαμβάνει έναν περιορισμένο δε, αλλά εξαιρετικά ευέλικτο και εκφραστικό τρόπο επικοινωνίας απόψεων και συναισθημάτων. Οι χρήστες επιτρέπεται να αναρτούν μόνο σύντομα μηνύματα ελεύθερου κειμένου έως και 140 χαρακτήρες, τα οποία ονομάζονται tweets. Αντιθέτως, άλλα Social Media προσφέρουν ένα πιο ποικιλόμορφο σύνολο αλληλεπιδράσεων μεταξύ των χρηστών, κάνοντας έτσι πιο πολύπλοκη τη μελέτη της συναισθηματικής ανάλυσης βασισμένης στο περιεχόμενο.

Δεύτερον, το Twitter προσφέρει εύκολους τρόπους πρόσβασης σε σημαντικό όγκο πραγματικών, παραγόμενων από χρήστες, δεδομένων, μέσω της εύχρηστης διεπαφής του (API). Υπάρχει επίσης ένα πρότυπο και καλώς εδραιωμένο μοντέλο αναπαράστασης (π.χ. εικονίδια συναισθημάτων – emoticons) για τα δεδομένα,το οποίο επιτρέπει την αποτελεσματική εξαγωγή των χαρακτηριστικών σημείων από μια δημόσια συζήτηση επί ενός θέματος. Τελευταίο ,αλλά εξίσου σημαντικό , το Twitter είναι από τα πιο δημοφιλή Social Media, με μια βάση χρηστών περίπου 200 εκατομμυρίων χρηστών, οι οποίοι αναρτούν ένα δις σύντομα μηνύματα τη βδομάδα.

Την επιτυχία του πιστοποιούν και οι πολλές εξειδικευμένες υπηρεσίες που έχουν αναπτυχθεί επάνω σε αυτό,όπως το twitrratr, μια εφαρμογή που παρακολουθεί απόψεις στο Twitter. Τα παραπάνω χαρακτηριστικά εξηγούν γιατί το Twitter βρίσκεται στο επίκεντρο εντατικής έρευνας.

Η προσέγγιση της [1] δημοσίευσης ασχολείται κυρίως με τα πολωμένα tweets. Δηλαδή τα tweets που εκφράζουν,είτε θετικό, είτε αρνητικό συναίσθημα, όπως υποδηλώνεται από το κατάλληλο emoticon. Πιο λεπτομερώς, θεωρούμε θετικό tweet αυτό που περιέχει οποιοδήποτε από τα ακόλουθα χαμογελαστά εικονίδια (smilies):

“:)” “(:” “:-)” “(-:” “:)” “(:” “:D” ή “=)” .

Από την άλλη πλευρά , ταξινομούμε ως αρνητικά τα tweets που είναι σημειωμένα με:

“:(“ “):” “:-(“ “)-:” “: (“ ή “) :” .

Tweets χωρίς καθόλου ένδειξη πολικότητας θεωρείται ότι δεν εκφράζουν αρνητικό ή θετικό συναίσθημα (ουδέτερα tweets). Ας σημειωθεί ότι στην ανάλυσή μας αγνοούμε εντελώς tweets που περιέχουν και

αρνητικό και θετικό smilie (δε θεωρούνται ούτε πολωμένα ούτε ουδέτερα). Αυτές οι υποθέσεις αποτελούν κοινή πρακτική στη σχετική βιβλιογραφία.

6.3 Διατύπωση του Προβλήματος

Η ανάλυση συναισθημάτων μοντελοποιείται συνήθως στη βιβλιογραφία ως ένα δυαδικό πρόβλημα. Στην πραγματικότητα αντιμετωπίζεται ως πρόβλημα ταξινόμησης μονής σήμανσης: κάθε κείμενο (π.χ. tweet) ανήκει σε μια κατηγορία μονής πολικότητας. Έτσι ο σκοπός είναι συνήθως να αναγνωρίσεις εάν ένα tweet είναι θετικό ή αρνητικό. Με βάση τους παραπάνω ορισμούς, αυτό το πρόβλημα μπορεί οριστεί τυπικά στο πλαίσιο του Twitter ως εξής:

Πρόβλημα 1 (Δυαδική Ταξινόμηση Πολικότητας) :

Δοσμένης μιας συλλογής από tweets T και του συνόλου των κλάσεων δυαδικής πολικότητας Pb = negative, positive, το ζητούμενο είναι να προσεγγιστεί η άγνωστη επιθυμητή συνάρτηση F : T ⟶ Pb, η οποία περιγράφει την πόλωση των tweets σύμφωνα με ένα αλάνθαστο πρότυπο, το οποίο είναι μια συνάρτηση F' : T ⟶ Pb , η οποία λέγεται binary polarity classifier (δυαδικός ταξινομητής πολικότητας).

Αυτές οι ρυθμίσεις απλοποιούν το πρόβλημα της συναισθηματικής ανάλυσης, με αποτέλεσμα υψηλότερη ακρίβεια ταξινόμησης. Εντούτοις, η είσοδος εφαρμογών συναισθηματικής ανάλυσης πραγματικών δεδομένων σπάνια αποτελείται από απλά πολωμένα tweets. Πρακτικές μέθοδοι πρέπει συνεπώς να λάβουν υπόψιν την επιπρόσθετη κλάση των ουδετέρων tweets. Το παρακάτω πιο γενικευμένο πρόβλημα ταξινόμησης πολικότητας πρέπει να ληφθεί επίσης υπόψιν:

Πρόβλημα 2 (Γενικευμένη Ταξινόμηση Πολικότητας) :

Δοσμένης μιας συλλογής από tweets T και του συνόλου των κλάσεων γενικευμένης πολικότητας Pg = negative, positive, neutral, το ζητούμενο είναι να προσεγγιστεί η άγνωστη επιθυμητή συνάρτηση F : T ⟶ Pg, η οποία περιγράφει την πόλωση των tweets σύμφωνα με ένα αλάνθαστο πρότυπο, το οποίο είναι μια συνάρτηση F' : T ⟶ Pg , η οποία λέγεται general polarity classifier (γενικευμένος ταξινομητής πολικότητας).

Υφιστάμενες δουλειές που αντιμετωπίζουν το πρόβλημα 2 πάντα το χωρίζουν σε δύο στάδια: το πρώτο στοχεύει στην κατηγοριοποίηση των κειμένων σε υποκειμενικά και αντικειμενικά(π.χ. ουδέτερα) , ενώ το δεύτερο περαιτέρω τα διαχωρίζει τα υποκειμενικά σε θετικά και αρνητικά. Στην δημοσίευση [1] τα δύο στάδια συγχωνεύτηκαν στο πρόβλημα 2 για δύο λόγους: Πρώτον, έτσι παρέχεται μια συνολική επισκόπηση της απόδοσης της συναισθηματικής ανάλυσης, και δεύτερον, παρουσιάζεται η επίδραση του να λαμβάνεται υπόψιν μια επιπρόσθετη κλάση στο πρόβλημα 1.

7 Μοντέλα Αναπαράστασης Εγγράφων για Ανάλυση Συναισθήματος

Παρακάτω παρουσιάζονται αρχικώς τα μοντέλα αναπαράστασης κειμένων που χρησιμοποιούνται συνήθως στο πλαίσιο της συναισθηματικής ανάλυσης βασισμένης στο περιεχόμενο. Αναλύονται συνοπτικά οι διαφορές τους και οι αδυναμίες τους όταν εφαρμόζονται στο πολυγλωσσικό , θορυβώδες περιεχόμενο του Twitter και των άλλων Social Media. Στην συνέχεια εισάγεται η καινούργια τεχνική για τη σύλληψη μοτίβων κειμένου της δημοσίευσης [1], και αναλύονται τα τεχνικά της χαρακτηριστικά που την καθιστούν κατάλληλη για ταξινόμηση πολικότητας.

7.1 Μοντέλο Διανυσμάτων Όρων

Η μέθοδος αυτή του Μοντέλου Διανυσμάτων Όρων (Term Vector Model) αποτελεί τον ακρογωνιαίο λίθο της Ανάκτησης Δεδομένων ως κυρίαρχη τεχνική για τον εντοπισμό των κειμένων που είναι πιο σχετικές με ένα ερώτημα-κλειδί. Στο πλαίσιο της ταξινόμησης κειμένου,και συνεπώς της συναισθηματικής ανάλυσης, χρησιμοποιείται ως εξής: Δοσμένης μια συλλογής από tweets T, αθροίζει το σύνολο των διακριτών λέξεων W (π.χ. tokens) που ανήκουν στην T. Κάθε tweet ti που ανήκει στην T αναπαρίσταται σαν ένα διάνυσμα vti, όπου vti = (v1,v2,...,v|w|) μεγέθους |W|, με την j-οστή του διάσταση vj να ποσοτικοποιεί την “βαρύτητα” του j-οστού token (όρου) wj για το ti (όπου wj ανήκει στο W).

Αυτή η μέτρηση συνήθως εκφράζεται μέσω των TF-IDF βαρών: η τιμή κάθε όρου wj ορίζεται ως το γινόμενο του Term frequency (TFj) του, και του Inverse Document Frequency (IDFj) του. Το TF υποδηλώνει το πλήθος των φορών που ο αντίστοιχος όρος εμφανίζεται σε ένα συγκεκριμένο έγγραφο (π.χ. tweet). Το IDF συμπυκνώνει την συχνότητα μεταξύ εγγράφων των όρων, με σκοπό να υποβαθμίσει το βάρος των tokens που εμφανίζονται σε πολλά tweets (π.χ. λέξεις τερματισμού). Πιο αναλυτικά :

$$DF_i = \log \frac{|\mathcal{T}|}{|\{t : w_i \in t \wedge t \in \mathcal{T}\}|}$$

,όπου ο αριθμητής συμβολίζει το μέγεθος της συλλογής εισόδου, και ο παρονομαστής το πλήθος των tweets που περιέχουν την λέξη wi.

Παρόμοια με τα μεμονωμένα έγγραφα (π.χ. tweets), οι κλάσεις πολικότητας μοντελοποιούνται ως διανύσματα όρων, τα οποία περιλαμβάνουν τους όρους που έχουν συγκεντρωθεί από τα tweets τα οποία χαρακτηρίζουν.

7.2 Ν-Γράμματα Χαρακτήρων

Το σύνολο των ν-γραμμάτων χαρακτήρων (Character N-Grams) ενός κειμένου περιλαμβάνει τις υπο-συμβολοσειρές μήκους ν του αρχικού κειμένου. Τα πιο συνήθη μεγέθη για το ν είναι τα 2 (bigrams), 3 (trigrams), 4 (four-grams). Για παράδειγμα, η φράση “home_phone” αποτελείται από τα ακόλουθα τρι-γράμματα: hom , ome , me_ , _ph , pho , hon , one .

Σύμφωνα με το μοντέλο των character n-grams, κάθε tweet αναπαρίσταται από ένα διάνυσμα,του οποίου η i-οστή διάσταση συμπυκνώνει τη βαρύτητα του i-οστού n-gram. Αντί των TF-IDF βαρών, τα n-grams χρησιμοποιούνται συνήθως σε συνδυασμό με την TF των αντίστοιχων n-grams. Παρομοίως, μια κλάση πολικότητας μοντελοποιείται ως ένα διάνυσμα που περιέχει όλα τα διαφορετικά n-grams που περιέχονται στα tweets τα οποία χαρακτηρίζει.

Το κύριο πλεονέκτημα του μοντέλου αυτού έναντι του term vector model είναι η ανοχή του στον θόρυβο και τα ορθογραφικά λάθη: με το να ασχολείται με υπο-συμβολοσειρές αντί ολόκληρων λέξεων, η πιθανότητα σοβαρών ορθογραφικών λαθών μειώνεται σημαντικά.

7.3 Γράφοι Ν - Γραμμάτων

Στη δημοσίευση [1] εισάγεται μια νέα μέθοδος για τη σύλληψη κειμενικών προτύπων, η οποία υποστηρίζει εγγενώς το προαναφερθέν απαιτητικό είδος περιεχομένου των Social Media. Στην ουσία, δημιουργεί έναν γράφο του οποίου οι κόμβοι αντιστοιχούν στους ν-γράφους των χαρακτήρων του εγγράφου, ενώ οι σταθμισμένες ακμές του υποδηλώνουν τη μέση απόσταση μεταξύ τους. Διαφορετικά έγγραφα της αυτής πολικότητας μπορούν να συγκεντρωθούν σε έναν γράφο κλάσης πολικότητας, ο οποίος μπορεί να συγκριθεί με εξατομικευμένα έγγραφα προκειμένου να αναγνωριστεί η κατηγορία του συναισθήματός τους.

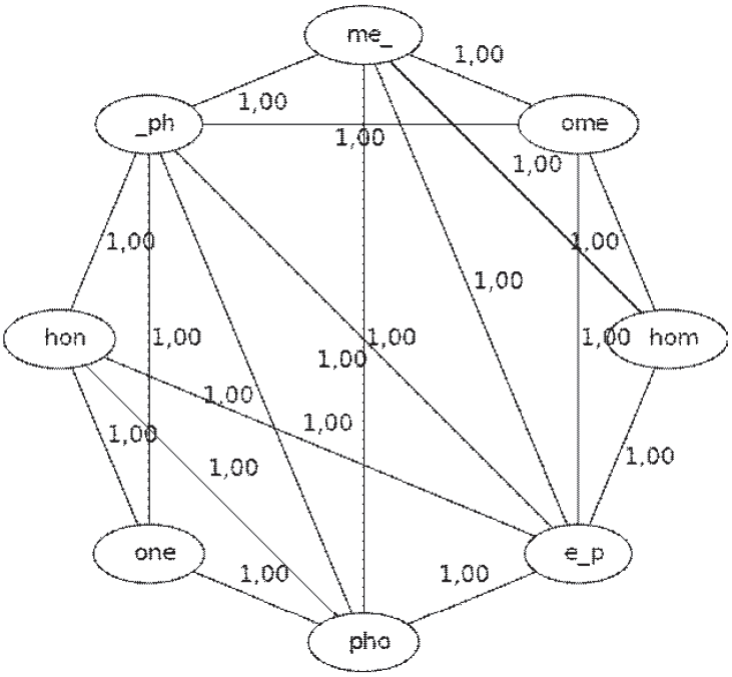
Προς αξιολόγηση της προσέγγισης , οι συγγραφείς της [2-1] διεξήγαγαν πειράματα μεγάλης κλίμακας σε ρεαλιστικά δεδομένα που απορρέουν από ένα στιγμιότυπο της δραστηριότητας του Twitter. Τα αποτελέσματα της προσέγγισης υποδεικνύουν σημαντικές βελτιώσεις συγκριτικά με άλλες μεθόδους που χρησιμοποιούνται συνήθως σε αυτό το περιεχόμενο, όχι μόνο σε σχέση με την αποδοτικότητα,αλλά και την απόδοση.

Το κύριο μειονέκτημα του προηγούμενου μοντέλου αποτελεί το γεγονός ότι μετατρέπει ένα tweet σε μια συλλογή από n-grams, αγνοώντας έτσι πολύτιμη πληροφορία που είναι συμπυκνωμένη στην πραγματική θέση των n-grams στο αρχικό κείμενο. Για παράδειγμα, οι λέξεις “wiki” και “kiwi” έχουν την ίδια αναπαράσταση σε bigrams , παρόλο που η σημασία τους είναι τελείως διαφορετική.

Για να ξεπεραστεί αυτό το πρόβλημα , η μέθοδος των Γράφων Ν - Γραμμάτων (n-gram graphs) , η οποία αρχικά επινοήθηκε ως μέθοδος περιληπτικής παρουσίασης , συσχετίζει όλα τα δυνατά ζευγάρια από n-grams με ακμές που υποδηλώνουν πόσο κοντά βρίσκονται κατά μέσο όρο στο(α) δοσμένο(α) tweet(s). Δηλαδή , σχηματίζει ένα γράφημα του οποίου οι κόμβοι αντιστοιχούν σε διακριτά n-grams, ενώ οι ακμές του σταθμίζονται αναλογικά με την μέση απόσταση – σε ορολογία n-gram – μεταξύ των αντίστοιχων κόμβων.

Προς διευκρίνηση αυτής της δομής, η παρακάτω εικόνα απεικονίζει τον n-gram γράφο που πηγάζει από την φράση “home_phone”. Προφανώς προσφέρει περισσότερη πληροφορία από ότι η αναπαράσταση της ίδιας φράσης με trigrams μέσω του μοντέλου των ν-γραμμάτων χαρακτήρων.

Σχήμα 8: *Δείγμα tri-gram γράφου, το οποίο αντιπροσωπεύει τη συμβολοσειρά home_phone*



Κάθε τριπλέτα γραμμάτων αυτής της συμβολοσειράς αντιστοιχεί σε έναν κόμβο στο γράφο, ενώ οι ακμές συνδέουν τριπλέτες, των οποιών η απόσταση είναι μικρότερη των τριών γραμμάτων , ανεξαρτήτου της σχετικής τους θέσης.

Τυπικότερα, ένας n-gram γράφος ορίζεται ως εξής:

Ορισμός 1 (N-Gram Graph): Ένας n-gram graph είναι ένας γράφος G = Vg,Eg,W, όπου Vg είναι το σύνολο των κορυφών (επισημασμένων από το αντίστοιχο n-gram) , Eg είναι το σύνολο των προσανατολισμένων ακμών (επισημασμένων από την συνένωση των ετικετών των κορυφών στην κατεύθυνση της σύνδεσης) , και W είναι μια συνάρτηση που αντιστοιχίζει ένα βάρος σε κάθε ακμή.

Σημειώστε ότι ένας n-gram graph χαρακτηρίζεται από τρεις παραμέτρους: (i) τον ελάχιστο n-gram βαθμό Lmin , (ii) τον μέγιστο n-gram βαθμό Lmax και (iii) την μέγιστη απόσταση γειτνίασης Dwin. Στα επόμενα , θεωρούμε αποκλειστικά τη ρύθμιση Lmin = Lmax = Dwin = n, η οποία επαληθεύτηκε πειραματικά ότι προσφέρει απόδοση κοντά στην βέλτιστη, όπως προέκυψε μετά από λεπτομερή διαδικασία ρύθμισης.

Για την κατασκευή ενός n-gram γράφου , ένα παράθυρο μεγέθους Dwin τρέχει πάνω σε ένα δοσμένο tweet, αναλύοντάς το σε επικαλυπτόμενα ν-γράμματα χαρακτήρων και καταγράφοντας πληροφορία σχετικά με τα γειτονικά ν-γράμματα (εντός του παραθύρου). Έτσι, μια ακμή eg που ανήκει στο Eg και συνδέει ένα ζεύγος ν-γραμμάτων υποδεικνύει την εγγύτητα μεταξύ αυτών των ακολουθιών χαρακτήρων

(των n-grams δηλαδή) στο αρχικό κείμενο, εντός του προκαθορισμένου παραθύρου μεγέθους Dwin. Το πραγματικό βάρος των ακμών εκτιμάται μετρώντας το εκατοστιαίο ποσοστό των κοινών εμφανίσεων των κορυφών-n-grams εντός τους παραθύρου.

Το μοντέλο των n-gram graphs μπορεί να χρησιμοποιηθεί για να αναπαριστά ομοιόμορφα ένα σύνολο από tweets (π.χ. μια κλάση πολικότητας) μέσω ενός μόνο γράφου. Αυτό μπορεί να υλοποιηθεί πολύ απλά με τη βοήθεια της λειτουργίας αναβάθμισης: δοσμένου ενός συνόλου από tweets T, χτίζει έναν αρχικά άδειο γράφο Gt. Έπειτα το i-οστό tweet ti του T μετατρέπεται σε έναν n-gram γράφο Gti, ο οποίος κατόπιν συγχωνεύεται με τον Gt προς σχηματισμό ενός νέου γράφου Gu με τις ακόλουθες ιδιότητες: Gu=(Eu,Vu,Wui), όπου *Eu = Egt και Egti* , *Vu = Vgt και Vgti* και *Wui(e) = Wgt(e) + (Wgti(e) − Wgt(e))*(1/i)* (η διαίρεση με το i στον υπολογισμό των νέων βαρών εξασφαλίζει ότι το συνολικό βάρος συγκλίνει στην μέση τιμή των ατομικών τιμών βάρους). Το e είναι μία ακμή του n-gram γράφου.

7.3.1 Είδη Ομοιότητας Μεταξύ των N-Gram Γράφων

Με τη βοήθεια της λειτουργίας αναβάθμισης, μπορούμε να συνδυάσουμε όλα τα tweets (του training set) για κάθε κλάση πολικότητας σε έναν κοινό γράφο (π.χ. μία κλάση για τα αρνητικά , μία για τα θετικά και μία για τα ουδέτερα tweets). Οι τελικοί γράφοι συλλαμβάνουν μοτίβα κοινά στα περιεχόμενα κάθε κλάσης, όπως επαναλαμβανόμενες και γειτονικές ακολουθίες χαρακτήρων , ειδικούς χαρακτήρες και ψηφία.

Έτσι μπορούν να χρησιμοποιηθούν για να μετρήσουν την ομοιότητα ενός μεμονωμένου tweet (από το testing set) με κάθε κλάση πολικότητας. Η ομοιότητα υπολογίζεται μεταξύ του αντίστοιχου γράφου αναπαράστασης Gti του tweet και του γράφου αναπαράστασης της κλάσης, Gj. Στο πλαίσιο της [1] δημοσίευσης χρησιμοποιούνται τα ακόλουθα τρία είδη ομοιότητας μεταξύ των n-gram γράφων:

(i) Containment Similarity (CS) : Εκφράζει το ποσοστό των ακμών ενός γράφου Gi που είναι κοινές με έναν δεύτερο γράφο Gj. Υποθέτοντας ότι G είναι ένας n-gram γράφος, e είναι μία ακμή του και ότι για μία συνάρτηση μ(e,G) ισχύει ότι μ=1 αν και μόνο αν η e ανήκει στο G, αλλιώς μ=0, τότε :

$$\mathrm{CS}(G^i, G^j) = \frac{\sum_{e \in G^i} \mu(e, G^j)}{\min(|G^i|, |G^j|)}$$

, όπου |G| υποδηλώνει το πλήθος των ακμών του G (π.χ. το μέγεθος του n-gram γράφου).

(ii) Size Similarity (SS) : Υποδηλώνει την αναλογία των μεγεθών δύο γράφων :

$$\mathrm{SS}(G^i, G^j) = \frac{\min(|G^i|,|G^j|)}{\max(|G^i|,|G^j|)}$$

(iii) Value Similarity (VS) : Δείχνει πόσες από τις ακμές που περιέχονται στον γράφο Gi περιέχονται και στον γράφο Gj, λαμβάνοντας υπόψιν επίσης και τα βάρη των ακμών που ταιριάζουν στο ζητούμενο αυτό. Σε αυτήν τη μέτρηση , κάθε ταιριαστή ακμή e με βάρος Wi(e) στον γράφο Gi συνεισφέρει [VR(e) / max{|Gi|,|Gj|}] στο σύνολο, ενώ μη ταιριαστές ακμές δεν συνεισφέρουν καθόλου (για ακμή e που δεν ανήκει στον Gi ορίζουμε wi(e) = 0).

Το ValueRatio (VR) είναι ένας βαθμωτός παράγοντας που ορίζεται ως :
VR(e) = [min{wi(e) , wj(e)} / max{wi(e) , wj(e)}]. Η εξίσωση δείχνει ότι το VR είναι συμμετρικό, παίρνοντας τιμές στο διάστημα [0,1]. Έτσι η πλήρης εξίσωση για το VS είναι:

$$\mathrm{VS}(G^i, G^j) = \frac{\sum_{e \in G^i} \frac{\min(w_e^i, w_e^j)}{\max(w_e^i, w_e^j)}}{\max(|G^i|,|G^j|)}$$

Αυτή η μετρική συγκλίνει στο 1 για γράφους που μοιράζονται κοινές ακμές και παρόμοια βάρη , με την τιμή VS = 1 να υποδεικνύει τέλειο ταίριασμα μεταξύ των υπό σύγκριση γράφων.

(iv) Μια σημαντική παράγωγη μετρική είναι η Normalized Value Similarity (NVS), η οποία υπολογίζεται ως : NVS(Gi,Gj) = [VS(Gi,Gj) / SS(Gi,Gj)]. Η NVS είναι μια μετρική ομοιότητας τιμής όταν η αναλογία των μεγεθών των υπό σύγκριση γράφων δεν παίζει κάποιον ρόλο.

Συνολικά, σύμφωνα με το μοντέλο των n-gram γράφων , για την ταξινόμηση ενός tweet απλώς λαμβάνουμε υπόψιν τρεις μετρικές για κάθε μία από τις εμπλεκόμενες κλάσεις. Αυτές είναι οι CS, VS και η NVS για την αρνητική και θετική κλάση του δυαδικού προβλήματος πόλωσης, και ακριβώς οι ίδιες για την ουδέτερη κλάση του γενικευμένου προβλήματος ταξινόμησης.

7.3.2 Διακριτοποιώντας τις Ομοιότητες των N-Gram Γράφων

Οι παραπάνω ομοιότητες μπορούν να λάβουν πολύ χαμηλές τιμές. Για παράδειγμα, τα Containment Similarities (CS) ενός συγκεκριμένου tweet με τα αρνητικά και με τα θετικά μπορεί να διαφέρουν μόνο στο 8ο δεκαδικό ψηφίο. Για μερικούς ταξινομητές, αυτή η διαφορά είναι πολύ λεπτή για να μπορεί να χρησιμοποιηθεί στην αναγνώριση αξιόπιστων μοτίβων. Σε αυτές τις περιπτώσεις, μια διακριτή τιμή μπορεί να χρησιμοποιηθεί για να αποσαφηνιστεί η σωστή κλάση για ένα tweet. Συνεπώς, μπορούμε να διακριτοποιήσουμε τις ομοιότητες ενός συγκεκριμένου tweet (Discretizing N-Gram Graph Similarities) σε σχέση με δύο κλάσεις πολικότητας του προβλήματος 1 ως εξής:

$$dsim(sim_{neg}, sim_{pos}) = \begin{cases} negative, & \text{if } sim_{pos} < sim_{neg} \\ equal, & \text{if } sim_{neg} = sim_{pos} \\ positive, & \text{if } sim_{neg} < sim_{pos} \end{cases}$$

Παρατηρήστε ότι αυτή η τεχνική διακριτοποίησης εφαρμόζεται σε ομοιότητες του ίδιου είδους (π.χ. συγκρίνοντας το CS της αρνητικής τάξης με το CS της θετικής). Ως αποτέλεσμα, ένα tweet ταξινομείται στο δυαδικό πρόβλημα πολικότητας σύμφωνα με τρία ονομαστικά χαρακτηριστικά: dsim(CSneg , CSpos) , dsim(NVSneg, NVSpos) , dsim(VSneg,VSpos).

Στην περίπτωση του προβλήματος 2, έξι περισσότερα ονομαστικά χαρακτηριστικά προστίθενται, συγκρίνοντας τις ομοιότητες της ουδέτερης κλάσης με τις αντίστοιχες της αρνητικής - dsim(CSneg , CSneu) , dsim(NVSneg, NVSneu) , dsim(VSneg,VSneu) – και της θετικής - dsim(CSpos ,CSneu) , dsim(NVSpos, NVSneu) , dsim(VSpos,VSneu) .

8 Υλοποίηση

Έχοντας πια γνωρίσει το μοντέλο αναπαράστασης εγγράφων για συναισθηματική ανάλυση " Γράφοι Ν – Γραμμάτων" , θα παρουσιαστεί τώρα ο τρόπος αξιοποίησής του:

Τα datasets που χρησιμοποιήθηκαν ως είσοδος τροποποιήθηκαν ανεξάρτητα το καθένα ώστε να απομακρυνθούν σοβαρά λάθη στη δομή τους , και κατόπιν εντατική επεξεργασία ακολούθησε ώστε να παρουσιάζουν τιμές ακριβώς για τις ίδιες πληροφορίες , καθώς και για να τις παρουσιάζουν με ακριβώς τον ίδιο τρόπο. Τα επεξεργασμένα datasets ενώθηκαν σε ένα των 1,586,980 instances. 'Επειτα φιλτραρίστηκαν ώστε να εξαλειφθούν δεδομένα κειμένου περιττά για την ταξινόμηση, δηλαδή θόρυβος, και κατόπιν ταξινομήθηκαν λεξικογραφικά ως το περιεχόμενο των tweets,εντός του ολικού dataset.

Το ολικό dataset χωρίστηκε σε τρία, όπου κάθε ένα εκ των τριών είχε instances της ίδιας μόνο κλάσης πολικότητας , όπου τις κλάσεις τις έχουν δώσει από πριν οι πηγές των datasets. Πάρθηκε από το κάθε dataset εκ των τριών τελευταίων το 70% , αφού πρώτα ανακατεύτηκαν τα instances με τυχαίο τρόπο εντός του κάθε αρχείου εκ των τριών. Από κάθε τμήμα 70%, δημιουργήθηκαν γράφοι ν-γραμμάτων, δηλαδή γράφοι ανά κλάση πολικότητας. Ως μέγεθος ν των ν-γραμμάτων επιλέχθηκαν οι τιμές 1,2,3 και 4 , και ως αποτέλεσμα, η συνέχεια της υλοποίησης γίνεται 4 φορές, μία για κάθε μέγεθος. Συνολικά λοιπόν δημιουργήθηκαν 3*4 = 12 γράφοι για τις κλάσεις πολικότητας. Αυτοί όπως είδαμε παραπάνω, αντιπροσωπεύουν τις κλάσεις, και η ταξινόμηση κάθε νέου instance γίνεται σε σύγκριση με αυτούς. Ανά ν-μέγεθος , τα 30% κομμάτια ενώθηκαν σε ένα ενιαίο.

Σε αυτό το σημείο πρέπει να σημειωθεί το εξής: Η τυπική διαδικασία έχει το training set να είναι σε μορφή κειμένου, πάνω στο οποίο άμεσα γίνεται η εκπαίδευση του αλγορίθμου μηχανικής μάθησης και έπειτα το testing set ακολουθεί, περιέχοντας μικρά κείμενα, τα "τιτιβίσματα" (tweets) του Twitter, για τα οποία θα προβλεφθεί ένα συναίσθημα (θετικό, αρνητικό, ουδέτερο) , από τον "μορφωμένο" πια αλγόριθμο. Ίσως επιπροσθέτως να μην είναι εύκαιρο κάποιο testing set, και συνεπώς με ευφυείς τρόπους προκύπτουν συνολικά αποτελέσματα από την χρήση πολλών μικρών διαφορετικών κομματιών του αρχικού διαθέσιμου dataset ως testing set, και του υπόλοιπου dataset ως training set. Στην μέθοδο με γράφους ν-γραμμάτων αποφεύγεται η άμεση επιμόρφωση των αλγορίθμων πάνω σε κείμενο, και όσα ελαττώματα συνεπάγεται αυτή. Στην παρούσα διπλωματική χρησιμοποιήσαμε το 70% των 1,586,980 instances, δηλαδή μία μεγάλη ποσότητα, για την παραγωγή γράφων κλάσεων πολικότητας. Οι γράφοι αποτελούν στην ουσία ένα βήμα πριν τα training sets: Η πολύ μεγάλη ποσότητα από instances που χρησιμοποιήθηκε (θυσιάστηκε) για τη δημιουργία τους, εξασφαλίζει ότι αυτοί εμπεριέχουν μία αντιπροσωπευτικότατη περιγραφή των κλάσεων. Το ενιαίο σύνολο με τα τρία 30% κομμάτια (συνολικά 475,383 instances) θα μετατραπεί στο τελικό αριθμητικό training set.

Ο σαφής διαχωρισμός των 70% και 30% κομματιών είναι επιτακτικός. Έχουμε ήδη εξηγήσει αυτήν την ανάγκη όταν αναφερόμαστε σε training και testing sets. Ο έλεγχος πρέπει να γίνεται μόνο από καινούργια instances, και δεν πρέπει να χρησιμοποιείται ένα training instance ως testing set, γιατί ο αλγόριθμος θα είναι προκατειλημμένος απέναντί του, και κατά πάσα πιθανότητα θα το ταξινομήσει σωστά. Παρόμοια και στους γράφους πολικότητας, τα instances που έδωσαν τους γράφους θα έχουν ουτοπικά ιδανικές τιμές μετρικών ομοιότητας με τον γράφο στην κλάση του οποίου ανήκουν, αν χρησιμοποιηθούν ως test instances , και δώσουν μία εικονική βελτίωση στην απόδοση των αλγορίθμων.

Η μετατροπή γίνεται ως εξής: Κάθε instance με κείμενο δίνει το κείμενό του (tweet) να συγκριθεί με τους γράφους πολικότητας. Για κάθε tweet που συγκρίνεται προκύπτουν 9 τιμές μετρικών, οι Containment Similarity (CS), Value Similarity (VS) και Normalized Value Similarity (NVS) ανά κλάση (θετικό, αρνητικό, ουδέτερο). Άρα το νέο αριθμητικό tweet που προκύπτει είναι το CS_pos, VS_pos,NVS_pos, CS_neut, VS_neut, NVS_neut, CS_neg, VS_neg, NVS_neg, class , όπου class είναι η δοσμένη από τους ειδικούς, αληθινή κλάση του tweet. 475,383 τέτοιας δομής instances προκύψαν, και αποτελούν το τελικό αριθμητικό dataset. Στην πραγματικότητα έχει attributes με αριθμητικές τιμές, ενώ οι κλάσεις είναι 3 ονομαστικές τιμές.

Συνεχίζει σε αυτό το σημείο η υλοποίηση, και εκτελούνται αλγόριθμοι ταξινόμησης σε αυτό το τελευταίο dataset. Στην παρούσα διπλωματική χρησιμοποίησα και percentage-split και cross-validation για να φανούν οι διαφορές που προκαλούν στην απόδοση του αλγορίθμου. Πριν την εκτέλεση μέσω αυτών των τακτικών αξιολόγησης αλγορίθμων, είναι απαραίτητο να ανακατεύονται με τυχαίο τρόπο τα instances εντός του dataset. Ο τυχαίος αυτός τρόπος βασίζεται σε έναν ψευδο-τυχαίο αριθμό. Η παραγωγή του απαιτεί έναν αριθμό τροφοδοσίας (seed) (όπως χρειάστηκε και στην java γλώσσα που χρησιμοποιήθηκε). Τα percentage-split και cross-validation δέχονται ως επιλογή τον αριθμό τροφοδοσίας. Βασιζόμενος σε αυτό, εκτέλεσα τα percentage-split και cross-validation για αριθμούς τροφοδοσίας από 1 έως και 10 (ακέραιες τιμές) και βρήκα τους μέσους όρους ως τελικά αποτελέσματα των percentage-split και cross-validation ανά αλγόριθμο. Αυτή η διαδικασία έγινε στην πλειοψηφία των αλγορίθμων που μελετήθηκαν, με απώτερο σκοπό εγγυρότερα αποτελέσματα.

8.1 Συγγραφή Εργαλείου σε Java

Ο κώδικας που δημιουργήθηκε για την υλοποίηση της παρούσας διπλωματικής είναι γραμμένος στη γλώσσα προγραμματισμού Java. Η συγγραφή του πραγματοποιήθηκε μέσω του περιβάλλοντος προγραμματισμού Eclipse. Η αξιοποίηση των ιδιοτήτων του αντικειμενοστραφούς προγραμματισμού όχι μόνο προτιμήθηκε κατά τη συγγραφή του κώδικα, αλλά ήταν και απολύτως απαραίτητη για την ορθή οργάνωση και εκπόνηση των πολλών ζητούμενων της διπλωματικής, καθώς και των ποικίλλων ενδιάμεσων επιπέδων τους.

Ενώ ο τρόπος οργάνωσης των δομών δεδομένων δεν είναι ποτέ μοναδικός, η τελική κατηγοριοποίηση που ακολούθησα δείχνει πολύ ακέραια και όχι εύκολα αντικαταστάσιμη. Οι μεγάλες διεργασίες χωρίστηκαν σε εργασίες(projects) με επιμέρους ζητούμενα. Κάθε ουσιώδες ζητούμενο υλοποιήθηκε από μια δομή δεδομένων (κλάση). Κάθε κλάση που χρησίμευε επαναλαμβανόμενα σε άλλες, ομαδοποιήθηκε με τις όμοιές της σε μια βασική ομάδα(πακέτο). Κάθε διεργασία που χρησίμεψε τοπικά από κλάση-ζητούμενο, τοποθετήθηκε στο πακέτο της κλάσης αυτής,συνεπώς κάθε ζητούμενο όφειλε να έχει το δικό του πακέτο. Λοιπές κλάσεις υπεύθυνες για ξεπερασμένες εργασίες ή για διαδικασίες αποσφαλμάτωσης διατηρήθηκαν προς αποφυγή συγχύσεως σε ένα τελευταίο ξεχωριστό πακέτο του project.

Για την εκπόνηση της παρούσας διπλωματικής εργασίας έγινε χρήση του λογισμικού Weka . Το Weka είναι λογισμικό εξόρυξης δεδομένων (data mining) γραμμένο στη γλώσσα προγραμματισμού Java. Πιο συγκεκριμένα, είναι μια συλλογή από machine learning αλγορίθμους για data mining σκοπούς. Οι αλγόριθμοι μπορούν να εφαρμοσθούν απευθείας πάνω στα datasets ή να κληθούν από Java πηγαίο κώδικα. Κατά τη συγγραφή του κώδικα χρησιμοποίησα τον πηγαίο κώδικα για την εφαρμογή που εκτελεί τους αλγόριθμους, και την διαπροσωπία του εργαλείου Weka για γρήγορες επαληθεύσεις πάνω σε δείγματα.Μερικοί από του αλγορίθμους εγκαθίστανται επιπρόσθετα μέσω του διαχειριστή πακέτων αλγορίθμων του Weka.

Το Weka εμπεριέχει εργαλεία για την προ-επεξεργασία (pre-processing), ταξινόμηση (classification), στατιστική παλινδρόμηση (regression), συσταδοποίηση (clustering) και απεικόνιση (visualization) δεδομένων, καθώς και για την εύρεση κανόνων συσχέτισης(association rules) επί των δεδομένων. Είναι κατάλληλο και για την ανάπτυξη νέων machine learning σχεδίων. Είναι ανοικτό λογισμικό υπό την Γενική Άδεια Δημόσιας Χρήσης GNU και βρίσκεται στην 3η έκδοσή του.

Για την παραγωγή των γράφων πολικότητας ν-γραμμάτων χρησιμοποιήθηκε ο πηγαίος κώδικας του JInsect project του Γιώργου Γιαννακόπουλου.

8.2 Υπολογιστικοί Πόροι

Για την υλοποίηση χρησιμοποιήθηκαν δύο υπολογιστικά συστήματα, ένα για την συγγραφή και την δοκιμή του κώδικα, και ένα για την εκτέλεση των πειραμάτων. Ο server για την εκτέλεση των πειραμάτων είχε τις παρακάτω δυνατότητες:

Manufacturer Alienware – Model Aurora-R4
Windows 7 Professional – 64-bit operating system
Processor Intel(R) Core(TM) i7-3820 CPU @ 3.60GHz – Number of processor cores 4
Total amount of system memory 32,0 GB RAM

Η μέγιστη χρήση μνήμης RAM κατά την εκτέλεση των αλγορίθμων ταξινόμησης στα ολικά , τελικά datasets των 475,383 instances βρισκόταν κοντά στα 14 GB ανά αλγόριθμο για τον πιο απαιτητικό , ενώ 5 GB κατά μέσο όρο στους περισσότερους. Οι instance-based αλγόριθμοι απαιτούσαν κοντά στα 8 GB. Το όριο κατά την δημιουργία των γράφων και των ομοιοτήτων ήταν μικρότερο,αλλά κοντινό. Η χρήση του επεξεργαστή ήταν χαμηλή, καθώς τα έτοιμα υποπρογράμματα που χρησιμοποιήθηκαν για την παραγωγή των γράφων, των μετρικών ομοιότητας και της ταξινόμησης δεν δρούσαν με πολυνηματικό τρόπο.

Το σύστημα συγγραφής κώδικα , κατά πολύ απλούστερο , αλλά και επαρκές:

ASUSTeK Computer INC. P5K Deluxe (LGA775)
Windows 7 Professional 64-bit SP1
CPU Intel Core 2 Duo E6420 @ 2.13GHz
RAM 4,00GB Dual-Channel DDR2 @ 400MHz (6-6-6-18)

8.3 Δομή Δεδομένων Εισόδου

Όπως προείπαμε,τα δεδομένα εισόδου αποτελούνται από ανεξάρτητες μεταξύ τους σειρές κειμένου , δηλαδή από instances. Για την παρούσα διπλωματική απαραίτητο ήταν μόνο το κείμενο των χρηστών της υπηρεσίας κοινωνικής δικτύωσης Twitter και η αξιολόγηση των κειμένων αυτών. Πληθώρα άλλων στοιχείων όμως ήταν διαθέσιμη (π.χ. ημερομηνία, χρήστης του Twitter,κά.).

Η σημασία της δοσμένης αξιολόγησης του κειμένου είναι μεγάλη. Το κείμενο από μόνο του δεν μπορεί να προσφέρει μια βαθμολογία της απόδοσης των αλγορίθμων που θα δράσουν στο κείμενο. Μόνο εάν είναι δοσμένο το συναισθηματικό φορτίο των instances μπορεί να αξιολογηθεί σωστά η απόδοση του αλγορίθμου, συγκρίνοντας το τι αποφαίνεται με το ποιο είναι το πραγματικό συναίσθημα των δεδομένων εισόδου.

Οι συλλογές των δεδομένων εισόδου (datasets) που χρησιμοποιούνται για classification, καθώς και αυτές που χρησιμοποιήθηκαν στην παρούσα διπλωματική εργασία είναι προσεκτικά επιλεγμένες είτε από την ίδια την υπηρεσία του Twitter , είτε από ερευνητές , ώστε να είναι καλώς ορισμένο το συναισθηματικό φορτίο των tweets.

Οι συλλογές των tweets που χρησιμοποιήθηκαν είναι σε πλήθος τέσσερις:
(1): Dataset for Characterizing Debate Performance via Aggregated Twitter Sentiment by Nicholas Diakopoulos and David A. Shamma . Βλέπε δημοσιεύσεις [9] και [10] και dataset [1].
(2): Updown package , paper: Twitter Polarity Classification with Label Propagation over Lexical Links and the Follower Graph by Michael Speriosu, Nikita Sudan, Sid Upadhyay, and Jason Baldridge, from The EMNLP 2011 Workshop on Unsupervised Learning in NLP. Βλέπε δημοσίευση [11] και dataset [2].
(3): Twitter Sentiment Analysis Training Corpus (Dataset). Βλέπε dataset [3].
(4): Ένα πολύ μικρό ανεπίσημο,αλλά ελεγμένο και λειτουργικό dataset του εργαστηρίου στο οποίο υπάγεται η παρούσα διπλωματική.

Η αξιολόγηση γίνεται χειροκίνητα από ανθρώπους, και συχνά από παραπάνω του ενός για κάθε tweet. Το (1) παρακάτω έχει 8 διαφορετικές αξιολογήσεις ανά instance , ενώ το (4) δύο. Το συνολικό συναίσθημα στις περιπτώσεις πολλών αξιολογήσεων θεωρήθηκε ο μέσος όρος των αξιολογήσεων, αντιστοιχίζοντας (-1) για αρνητικό, 0 για ουδέτερο και (+1) για αρνητικό βαθμό.

Εδώ υπήρξε η ιδιαιτερότητα του διαφορετικού τρόπου συμβολισμού ενός συναισθήματος μεταξύ των datasets, η οποία επιλύθηκε στο πλαίσιο της γενικότερης επεξεργασίας των δεδομένων εισόδου όπως εξηγώ παρακάτω.

8.4 Τροποποίηση Εισόδου και Δυσκολίες

Η τροποποίηση των δεδομένων εισόδου αποτέλεσε μια επίπονη διαδικασία για πολλούς λόγους,όπως λόγω των διαφορετικών πηγών προέλευσης των δεδομένων,των διαφορετικών κανόνων οργάνωσής τους, της χρήσεως διαφορετικών συμβόλων για ταυτόσημες έννοιες. Η δε υπερβολικά μεγάλη τους έκταση δυσκόλεψε την μελέτη των δεδομένων μέσω της ανάγνωσής τους, η οποία είναι και το απαραίτητο πρώτο βήμα της μελέτης τους.

Επιπλέον, τα δεδομένα εισόδου εφόσον αποτελούν έγγραφες εκφράσεις ανθρώπινων συναισθημάτων κρύβουν πολλές ανομοιότητες, έλλειψη κανόνων ή επανάληψης, αλόγιστη χρήση πληθώρας συμβόλων με συνέπεια να μην περισσεύουν τα βολικότερα για προγραμματιστικές προσεγγίσεις ή επισημάνσεις.

Επιπροσθέτως, οι πηγαίοι συλλέκτες των δεδομένων αυτών δεν παρείχαν πάντα τιμές για κάθε στήλη ενός tweet, και δυστυχώς σε μερικά tweet δεν τήρησαν και τους κανόνες τους. Ιδιαιτερότητα υπήρξε και στην εκμάθηση του υποπρογράμματος που επεξεργαζόταν την είσοδο να αγνοεί αλλαγή σειράς στο κείμενο εισόδου εφόσον ένα δεδομένο εισόδου (instance) δεν έχει διαβαστεί πλήρως.

Παρόλο που η μετατροπή των δεδομένων σε αποδεκτή μορφή αποτελεί δευτερευούσης σημασίας σκέλος της διπλωματικής, ήταν δυστυχώς από τα πιο χρονοβόρα. Κώδικας έπρεπε να γραφεί επακριβώς πάνω στα μέτρα του κάθε αρχείου εισόδου, και συχνά παραπάνω της μία φοράς, ώστε να επιτευχθεί το σωστό φιλτράρισμα σε όσο το δυνατόν περισσότερες περιπτώσεις ακανόνιστων ,με τα υπόλοιπα του ίδιου αρχείου, tweets.

Οι αλγόριθμοι πριν την εκτέλεσή τους επί των ολικών datasets επαληθεύτηκαν ως προς το ορθό της λειτουργίας τους με απευθείας χρήση στην διαπροσωπία του Weka, σε μικρά δείγματα των datasets. Δημιουργήθηκαν συνεπώς ποικιλία δειγμάτων του τελικού αριθμητικού dataset.

Τα datasets μετετράπησαν σε μορφή αρχείου .arff πριν τη χρήση τους από το λογισμικό Weka, με ενσωματωμένες και τις απαραίτητες επικεφαλίδες.

8.5 Φιλτράρισμα Εισόδου

Η σημασία των ορθώς δομημένων δεδομένων εισόδου έχει γίνει έως τώρα σαφής. Οι γράφοι πολικότητας που αντιπροσωπεύουν την κλάση τους δηλητηριάζονται με άχρηστη πληροφορία όταν υπάρχει θόρυβος στα δεδομένα εισόδου, επηρεάζονται δηλαδή από περιττή πληροφορία που δεν πηγάζει από το αληθινό περιεχόμενο των όσων έχει ο χρήστης του Twitter να εκφράσει γραπτώς.

Λεπτομερές και σχετικά στοχευμένο φιλτράρισμα, δηλαδή εξαρτώμενο από τα συγκεκριμένα datasets, έπρεπε να λάβει χώρα. Εφαρμόστηκε λοιπόν στο πρώτο ολικό σχηματισμένο dataset, αυτό που προέκυψε από την ένωση όλων των άλλων, αφού πρώτα ομοιογενοποιήθηκε η δομή τους και του διορθώθηκαν σοβαρά λάθη (π.χ. πρέπει ένα instance να ανήκει σε μία σειρά του αρχείου όχι σε παραπάνω).
Προς εξάλειψη του θορύβου έγιναν τα παρακάτω ουσιαστικά βήματα στο διορθωμένο αυτό dataset , στο περιεχόμενο των tweets των instances:

– Αφαιρέθηκαν τα εισαγωγικά ’ “ ’ και ’ ” ’ όπου υπήρχαν σε ζεύγη, ακόμα και τα ζεύγη μέσα σε ζεύγη, αναδρομικά .
– Σε κάθε στάδιο του φιλτραρίσματος αφαιρούταν τα κενά στην αρχή και το τέλος .
– Σβήστηκαν οι σύνδεσμοι ιστοσελίδων (url links) .
– Σβήστηκαν τα emails .
– Σβήστηκαν οι αναφορές (mentions) που υπάρχουν στο Twitter (π.χ. Καλημέρα @Νίκο) .
– Σβήστηκαν οι απαντήσεις (responses) σε mentions ή γενικότερα (π.χ. Retweet @Νίκο σχετικά με την δήλωσή του για τον καιρό) .
– Πολλά άλλα λιγότερο σημαντικά .

Φροντίδα υπήρξε ώστε η απαλοιφή ενός είδους θορύβου να μην αφαιρεί κομμάτι από κείμενο άλλου είδους θορύβου, αφήνοντας έτσι υπολείμματα. Π.χ. Το https://d@john.com θεωρείται link και όχι αναφορά ή email. Επίσης στα datasets που χρησιμοποιήθηκαν υπήρχαν οι εξής εκφράσεις πριν από κάθε απάντηση: ”re”, ”rt”, ”rb”, ”re-pinging”, ”retweet” ,”retweeting”, ”Re”, ”Rt” ,”Rb”, ”Re-Pinging” ,”Re-pinging”, ”Retweet”, ”Retweeting”, ”RE”, ”RT”, ”RB”, ”RE-Pinging”, ”RE-pinging”, ”REtweet”, ”REtweeting”, ”RE-PINGING”, ”RETWEET”, ”RETWEETING”.
Πρώτα ταξινομήθηκαν οι εκφράσεις και αφαιρέθηκαν με φθίνουσα σειρά μήκους, καθώς κάποιες μικρότερες εμπεριέχονται σε μεγαλύτερες.

8.6 Εκτέλεση Αλγορίθμων και Αποτελέσματα

Ακολουθούν 4 πίνακες, ένας ανά μέγεθος ν των γράφων ν-γραμμάτων, με τους αλγόριθμους που εξετάστηκαν και τα αποτελέσματά τους . Αναλόγως την πολυπλοκότητα του κάθε αλγορίθμου, άρα και τον χρόνο που απαιτεί για την εκτέλεσή του, καθώς και την μνήμη που απαιτεί, επιλέχθηκε και το πλήθος των επαναλήψεών του. Πιο συγκεκριμένα, για τους αλγόριθμους χρησιμοποιήθηκαν οι μέθοδοι αξιολόγησης cross-validation και percentage split, όπως είδαμε παραπάνω. Στους ταχείς αλγορίθμους εφαρμόστηκαν και το cross-validation και το Percentage Split 10 φορές με τροφοδοσία τυχαίου αριθμού από 1 έως και 10. Τα 10 διαφορετικά αποτελέσματα ανά μέθοδο αξιολόγησης έδωσαν έναν αξιόπιστο μέσο όρο, συνεπώς στο τέλος προκύπτουν 2 αντιπροσωπευτικοί μέσοι όροι απόδοσης.

Εδώ αξίζει να σημειωθεί ότι ο μέσος όρος από αποτελέσματα 10 τροφοδοσιών προσφέρει ουσιαστική βελτίωση στα αποτελέσματα της μεθόδου Percentage Split. Στο cross-validation ήδη εξ ορισμού εξάγεται ένας μέσος όρος μεταξύ των αποτελεσμάτων κάθε επανάληψης. Με σύνηθες πλήθος επαναλήψεων (folds) 10 για το cross-validation, και με 10 διαφορετικές τροφοδοσίες, καταλήγουμε ανά αρχείο εισόδου να εκτελείται ταξινόμηση 10*10 = 100 φορές. Το αποτέλεσμα είναι πολύ αξιόπιστο, αλλά παρατηρήθηκε ότι και 10 φορές εξ ορισμού είναι αρκετές, και το αποτέλεσμά τους απέχει από αυτό των 100 φορών στα δεκαδικά ψηφία.

Βάσει του παραπάνω σκεπτικού, στους αργούς αλγορίθμους εκτελέστηκε cross-validation μόνο με τροφοδοσία 1, και γενικότερα επιλέχθηκαν συνδυασμοί στο πόσες φορές θα επιλεχθεί ο κάθε αλγόριθμος, λόγω περιορισμών υπολογιστικής ταχύτητας, μνήμης και χρόνου. Πάντα όμως επιλέγονται οι εκτελέσεις επί των αρχείων εισόδου που σχετίζονται με το μέγεθος (ν = 4) των γράφων ν-γραμμάτων, καθώς όπως φαίνεται και στους πίνακες παρακάτω, το μέγεθος (ν = 4) δίνει τα καλύτερα αποτελέσματα. Συνεπώς αν επιλεχθούν ν=4,3,2,1 , CVτροφοδοσία = 1-10 και PSτροφοδοσία = 1-10 , τότε πραγματοποιώ 4*[(10*10)+(10*1)] = 440 ταξινομήσεις, ενώ για ν=4,3,2,1 , CVτροφοδοσία = 1 και PSτροφοδοσία = 1 , 4*[(10*1)+1] = 44 ταξινομήσεις.

Ως εξαίρεση, οι KStar και LWL αλγόριθμοι εκτελέστηκαν πάνω σε δείγμα 50,000 Instances εκ των αρχικών 475,383 , λόγω περιορισμένης υπολογιστικής μνήμης. Επιπλέον, το Percentage Split εφαρμόστηκε παντού με ποσοστό 66%. Τέλος, κάποιοι εκ των αλγορίθμων επιχειρούσαν κατά την ταξινόμηση να διακριτοποιήσουν τα datasets εισόδου, όμως οι μετρικές που αυτά εμπεριέχουν απέχουν συχνά στο 6ο δεκαδικό ψηφίο τουλάχιστον, με αποτέλεσμα το συχνό φαινόμενο 2 ετικέτες κατά την διακριτοποίηση να προκύπτουν με το ίδιο όνομα, και η εκτέλεση να διακόπτεται. Ως λύση κανονικοποιήθηκαν τα δεδομένα εισόδου με φίλτρα που παρέχει το εργαλείο Weka, σε μία τάξη μεγέθους 1000000 φορές μεγαλύτερη των αρχικών τιμών, και η διακριτοποίηση μπόρεσε να κάνει σαφείς διαχωρισμούς στις ετικέτες.

Στους παρακάτω πίνακες, CV_folds είναι οι επαναλήψεις της μεθόδου Cross-Validation και CV_seeds οι τροφοδοσίες της. Ομοίως για τα PS_seeds της μεθόδου Percentage Split. CV_Score και PS_Score είναι οι επί τοις εκατό αποδόσεις για τις επιτυχημένες ταξινομήσεις των αλγορίθμων μέσω των CV και PS μεθόδων αξιολόγησης.

Σχήμα 9: Αποτελέσματα για n=4

		Μέγεθος n=4, PS= 66% , Dataset = 475,383 Instances Εξαίρεση τα Kstar,LWL σε δείγμα 50,000 Instances				
	weka.classifiers	CV_folds	CV_seeds	PS_seeds	**CV_Score**	**PS_Score**
1	rules.ZeroR	10	1-10	1-10	**49.943**	**49.951**
2	rules.OneR	10	1-10	1-10	**52.493**	**52.527**
3	trees.J48	10	1-10	1-10	**63.343**	**63.355**
4	bayes.NaiveBayes	10	1-10	1-10	**53.958**	**53.838**
5	misc.HyperPipes	10	1-10	1-10	**49.945**	**49.952**
6	rules.DecisionTable	10	1-10	1-10	**62.565**	**62.355**
7	bayes.BayesNet	10	1-10	1-10	**56.143**	**56.040**
8	rules.JRip	10	1-10	1-10	**61.237**	**61.220**
9	rules.PART	10	1	1	**62.816**	**62.706**
10	lazy.IBk	10	1	1	**56.493**	**56.324**
11	rules.ConjunctiveRule	10	1-3	1	**56.309**	**57.039**
12	lazy.KStar	-	-	1	**-**	**58.788**
13	lazy.LWL	-	-	1	**-**	**54.812**
14	rules.DTNB	10	1-10	1-10	**62.433**	**62.349**
15	rules.Ridor	10	1	1	**59.000**	**57.693**
16	bayes.AveragedNDependenceEstimators.A1DE	10	1-10	1-10	**61.881**	**61.885**
17	bayes.AveragedNDependenceEstimators.A2DE	10	1-10	1-10	**62.639**	**62.362**
18	trees.LMT	10	1-2	1	**64.816**	**64.937**
19	trees.NBTree	10	1-9	1	**62.651**	**62.893**
20	trees.RandomForest	10	1-10	1-10	**60.637**	**60.573**
21	trees.RandomTree	10	1-10	1-10	**56.906**	**57.031**
22	trees.FT	10	1-10	1-10	**64.686**	**64.241**
23	trees.LADTree	10	1-10	1-10	**60.400**	**60.376**
24	trees.REPTree	10	1-10	1-10	**63.015**	**62.973**
25	trees.SimpleCart	10	1	-	**64.487**	**-**
26	functions.SMO	10	1	1	**63.261**	**62.897**
27	functions.Logistic	10	1-10	1-10	**64.720**	**64.477**
28	functions.MLPClassifier	10	1-10	1-10	**63.743**	**63.706**

Σχήμα 10: Αποτελέσματα για n=3

	weka.classifiers	Μέγεθος n=3, PS= 66% , Dataset = 475,383 Instances Εξαίρεση τα Kstar,LWL σε δείγμα 50,000 Instances CV_folds	CV_seeds	PS_seeds	CV_Score	PS_Score
1	rules.ZeroR	10	1-10	1-10	**49.943**	**49.951**
2	rules.OneR	10	1-10	1-10	**51.476**	**51.259**
3	trees.J48	10	1-10	1-10	**58.485**	**58.258**
4	bayes.NaiveBayes	10	1-10	1-10	**53.253**	**53.150**
5	misc.HyperPipes	10	1-10	1-10	**49.943**	**49.951**
6	rules.DecisionTable	10	1-10	1-10	**58.462**	**58.508**
7	bayes.BayesNet	10	1-10	1-10	**54.932**	**54.632**
8	rules.JRip	10	1-10	1-10	**54.435**	**54.558**
9	rules.PART	10	1	1	**58.021**	**57.896**
10	lazy.IBk	10	1	1	**53.555**	**53.638**
11	rules.ConjunctiveRule	10	1	1	**54.888**	**54.947**
12	lazy.KStar	-	-	-	**-**	**-**
13	lazy.LWL	-	-	-	**-**	**-**
14	rules.DTNB	10	1-10	1-10	**58.080**	**58.543**
15	rules.Ridor	-	-	-	**-**	**-**
16	bayes.AveragedNDependenceEstimators.A1DE	10	1-10	1-10	**57.101**	**56.880**
17	bayes.AveragedNDependenceEstimators.A2DE	10	1-10	1-10	**58.259**	**58.029**
18	trees.LMT	10	1	1	**62.431**	**62.461**
19	trees.NBTree	10	1	1	**54.944**	**54.627**
20	trees.RandomForest	10	1-10	1-10	**56.970**	**56.908**
21	trees.RandomTree	10	1-10	1-10	**54.429**	**54.405**
22	trees.FT	-	-	-	**-**	**-**
23	trees.LADTree	10	1-10	1-10	**57.378**	**57.297**
24	trees.REPTree	10	1-10	1-10	**58.989**	**58.757**
25	trees.SimpleCart	-	-	-	**-**	**-**
26	functions.SMO	-	-	-	**-**	**-**
27	functions.Logistic	10	1-10	1-10	**62.299**	**62.301**
28	functions.MLPClassifier	10	1-10	1-10	**62.573**	**62.699**

Σχήμα 11: Αποτελέσματα για n=2

	weka.classifiers	Μέγεθος n=2, PS= 66% , Dataset = 475,383 Instances Εξαίρεση τα Kstar,LWL σε δείγμα 50,000 Instances CV_folds	CV_seeds	PS_seeds	CV_Score	PS_Score
1	rules.ZeroR	10	1-10	1-10	**49.943**	**49.951**
2	rules.OneR	10	1-10	1-10	**51.052**	**50.998**
3	trees.J48	10	1-10	1-10	**57.638**	**57.781**
4	bayes.NaiveBayes	10	1-10	1-10	**53.393**	**53.377**
5	misc.HyperPipes	10	1-10	1-10	**49.943**	**49.951**
6	rules.DecisionTable	10	1-10	1-10	**54.962**	**54.949**
7	bayes.BayesNet	10	1-10	1-10	**54.272**	**54.260**
8	rules.JRip	10	1-10	1-10	**53.806**	**53.947**
9	rules.PART	10	1	1	**55.970**	**56.528**
10	lazy.IBk	10	1	1	**52.841**	**52.814**
11	rules.ConjunctiveRule	10	1	1	**53.411**	**53.326**
12	lazy.KStar	-	-	-	**-**	**-**
13	lazy.LWL	-	-	-	**-**	**-**
14	rules.DTNB	10	1-10	1-10	**55.042**	**55.058**
15	rules.Ridor	-	-	-	**-**	**-**
16	bayes.AveragedNDependenceEstimators.A1DE	10	1-10	1-10	**54.735**	**54.729**
17	bayes.AveragedNDependenceEstimators.A2DE	10	1-10	1-10	**55.093**	**55.031**
18	trees.LMT	10	1	1	**61.353**	**61.277**
19	trees.NBTree	10	1	1	**54.278**	**54.256**
20	trees.RandomForest	10	1-10	1-10	**55.937**	**55.922**
21	trees.RandomTree	10	1-10	1-10	**53.751**	**53.657**
22	trees.FT	-	-	-	**-**	**-**
23	trees.LADTree	10	1-10	1-10	**54.877**	**55.040**
24	trees.REPTree	10	1-10	1-10	**57.833**	**57.873**
25	trees.SimpleCart	-	-	-	**-**	**-**
26	functions.SMO	-	-	-	**-**	**-**
27	functions.Logistic	10	1-10	1-10	**61.167**	**61.090**
28	functions.MLPClassifier	10	1-10	1-10	**60.350**	**61.224**

Σχήμα 12: Αποτελέσματα για n=1

	weka.classifiers	Μέγεθος n=1, PS= 66% , Dataset = 475,383 Instances Εξαίρεση τα Kstar,LWL σε δείγμα 50,000 Instances CV_folds	CV_seeds	PS_seeds	CV_Score	PS_Score
1	rules.ZeroR	10	1-10	1-10	49.943	49.951
2	rules.OneR	10	1-10	1-10	51.010	50.775
3	trees.J48	10	1-10	1-10	53.259	53.500
4	bayes.NaiveBayes	10	1-10	1-10	33.979	36.928
5	misc.HyperPipes	10	1-10	1-10	49.943	49.951
6	rules.DecisionTable	10	1-10	1-10	54.608	54.644
7	bayes.BayesNet	10	1-10	1-10	53.087	53.088
8	rules.JRip	10	1-10	1-10	52.802	53.145
9	rules.PART	10	1	1	52.981	53.177
10	lazy.IBk	10	1	1	51.959	51.895
11	rules.ConjunctiveRule	10	1	1	52.563	52.656
12	lazy.KStar	-	-	-	-	-
13	lazy.LWL	-	-	-	-	-
14	rules.DTNB	10	1-10	1-10	54.558	54.675
15	rules.Ridor	-	-	-	-	-
16	bayes.AveragedNDependenceEstimators.A1DE	10	1-10	1-10	53.532	53.417
17	bayes.AveragedNDependenceEstimators.A2DE	10	1-10	1-10	54.329	54.344
18	trees.LMT	-	-	-	-	-
19	trees.NBTree	10	1	1	53.104	53.082
20	trees.RandomForest	10	1-10	1-10	53.121	53.015
21	trees.RandomTree	10	1-10	1-10	52.026	52.005
22	trees.FT	-	-	-	-	-
23	trees.LADTree	10	1-10	1-10	54.359	53.931
24	trees.REPTree	10	1-10	1-10	54.514	54.485
25	trees.SimpleCart	-	-	-	-	-
26	functions.SMO	-	-	-	-	-
27	functions.Logistic	10	1-10	1-10	56.037	55.986
28	functions.MLPClassifier	10	1-10	1-10	56.328	56.460

Παρατηρούμε ότι οι αλγόριθμοι αποδίδουν καλύτερα για μέγεθος ν=4 στους γράφους ν-γραμμάτων. Επιπλέον ότι όλες οι κατηγορίες (trees, bayes, functions, κτλ) φιλοξενούν αλγορίθμους με υψηλή απόδοση, αλλά κυρίως οι functions και trees. Τέλος, εντοπίζουμε ότι κανένας αλγόριθμος δεν ξεπέρασε το κατώφλι της 65% επιτυχίας.

Ως ο πιο αξιόπιστος , αλλά με μικρές διαφορές από αρκετούς άλλους αλγορίθμους, προέκυψε στην παρούσα διπλωματική εργασία ο weka.classifiers.trees.LMT .

8.7 Συνοπτικές Τεκμηριώσεις των Αλγορίθμων

Παρατίθενται μερικές πληροφορίες για τους αλγορίθμους, παρμένες από τις τεκμηριώσεις στους πηγαίους κώδικες :

1. Class for building and using a 0-R classifier. Predicts the mean (for a numeric class) or the mode (for a nominal class). Eibe Frank (eibe@cs.waikato.ac.nz) $Revision: 10153 $

2. Class for building and using a 1R classifier; in other words, uses the minimum-error attribute for prediction, discretizing numeric attributes. Για περαιτέρω πληροφορίες βλέπε δημοσίευση [3]. Ian H. Witten (ihw@cs.waikato.ac.nz) $Revision: 10153 $

3. Class for generating a pruned or unpruned C4.5 decision tree. For more information, see Ross Quinlan (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA. Eibe Frank (eibe@cs.waikato.ac.nz) $Revision: 10531 $

4. Class for a Naive Bayes classifier using estimator classes. Numeric estimator precision values are chosen based on analysis of the training data. For more information on Naive Bayes classifiers, see George H. John, Pat Langley: Estimating Continuous Distributions in Bayesian Classifiers. In: Eleventh Conference on Uncertainty in Artificial Intelligence, San Mateo, 338-345, 1995. Len Trigg (trigg@cs.waikato.ac.nz) Eibe Frank (eibe@cs.waikato.ac.nz) $Revision: 10203 $

5. Class implementing a HyperPipe classifier. For each category a HyperPipe is constructed that contains all points of that category (essentially records the attribute bounds observed for each category). Test instances are classified according to the category that ”most contains the instance”. Does not handle numeric class, or missing values in test cases. Extremely simple algorithm, but has the advantage of being extremely fast, and works quite well when you have ”smegloads” of attributes. Lucio de Souza Coelho (lucio@intelligenesis.net) Len Trigg (len@reeltwo.com) $Revision: 8109 $

6. Class for building and using a simple decision table majority classifier. For more information see: Ron Kohavi: The Power of Decision Tables. In: 8th European Conference on Machine Learning, 174-189, 1995. author Mark Hall (mhall@cs.waikato.ac.nz) $Revision: 10153 $

7. Bayes Network learning using various search algorithms and quality measures. Base class for a Bayes Network classifier. Provides datastructures (network structure, conditional probability distributions, etc.) and facilities common to Bayes Network learning algorithms like K2 and B. For more information see: http://sourceforge.net/projects/weka/files/documentation/WekaManual-3-7-0.pdf . Remco Bouckaert (rrb@xm.co.r $Revision: 10386 $

8. This class implements a propositional rule learner, Repeated Incremental Pruning to Produce Error Reduction (RIPPER), which was proposed by William W. Cohen as an optimized version of IREP. Details please see: William W. Cohen: Fast Effective Rule Induction. In: Twelfth International Conference on Machine Learning, 115-123,1995. Xin Xu (xx5@cs.waikato.ac.nz) Eibe Frank (eibe@cs.waikato.ac.nz) $Revision: 101

9. Class for generating a PART decision list. Uses separate-and-conquer. Builds a partial C4.5 decision tree in each iteration and makes the ”best” leaf into a rule. For more information, see: Eibe Frank, Ian H. Witten: Generating Accurate Rule Sets Without Global Optimization. In: Fifteenth International Conference on Machine Learning, 144-151, 1998. Eibe Frank (eibe@cs.waikato.ac.nz) $Revision: 10153 $

10. K-nearest neighbours classifier. Can select appropriate value of K based on cross-validation. Can also do distance weighting. For more information, see D. Aha, D. Kibler (1991). Instance-based learning algorithms. Machine Learning. 6:37-66. Stuart Inglis (singlis@cs.waikato.ac.nz) Len Trigg (trigg@cs.waikato.ac.nz) Eibe Frank (eibe@cs.waikato.ac.nz) $Revision: 10141 $

11. This class implements a single conjunctive rule learner that can predict for numeric and nominal class labels. A rule consists of antecedents ”AND”ed together and the consequent (class value) for the classification/regression. In this case, the consequent is the distribution of the available classes (or mean for a numeric value) in the dataset. If the test instance is not covered by this rule, then it’s predicted using the default class distributions/value of the data not covered by the rule in the training data.This learner selects an antecedent by computing the Information Gain of each antecendent and prunes the generated rule using Reduced Error Prunning (REP) or simple pre-pruning based on the number of antecedents. For classification, the Information of one antecedent is the weighted average of the entropies of both the data covered and not covered by the rule. For regression, the Information is the weighted average of the mean-squared errors of both the data covered and not covered by the rule. In pruning, weighted average of the accuracy rates on the pruning data is used for classification while the weighted average

of the mean-squared errors on the pruning data is used for regression. Xin XU (xx5@cs.waikato.ac.nz) $Revision: 10335 $

12. K* is an instance-based classifier, that is the class of a test instance is based upon the class of those training instances similar to it, as determined by some similarity function. It differs from other instance-based learners in that it uses an entropy-based distance function. For more information on K*, see John G. Cleary, Leonard E. Trigg: K*: An Instance-based Learner Using an Entropic Distance Measure. In: 12th International Conference on Machine Learning, 108-114, 1995. Len Trigg (len@reeltwo.com) Abdelaziz Mahoui (am14@cs.waikato.ac.nz) $Revision: 10141 $

13. Locally weighted learning. Uses an instance-based algorithm to assign instance weights which are then used by a specified WeightedInstancesHandler. Can do classification (e.g. using naive Bayes) or regression (e.g. using linear regression). For more info, see Eibe Frank, Mark Hall, Bernhard Pfahringer: Locally Weighted Naive Bayes. In: 19th Conference in Uncertainty in Artificial Intelligence, 249-256, 2003. C. Atkeson, A. Moore, S. Schaal (1996). Locally weighted learning. AI Review.. Len Trigg (trigg@cs.waikato.ac.nz) Eibe Frank (eibe@cs.waikato.ac.nz) Ashraf M. Kibriya (amk14[at-the-rate]cs[dot]waikato[dot]ac[dot]nz) $Revision: 10141 $

14. Class for building and using a decision table/naive bayes hybrid classifier. At each point in the search, the algorithm evaluates the merit of dividing the attributes into two disjoint subsets: one for the decision table, the other for naive Bayes. A forward selection search is used, where at each step, selected attributes are modeled by naive Bayes and the remainder by the decision table, and all attributes are modelled by the decision table initially. At each step, the algorithm also considers dropping an attribute entirely from the model. For more information, see: Mark Hall, Eibe Frank: Combining Naive Bayes and Decision Tables. In: Proceedings of the 21st Florida Artificial Intelligence Society Conference (FLAIRS), ???-???, 2008. Mark Hall (mhall[at]pentaho[dot]org) Eibe Frank (eibe[at]cs[dot]waikato[dot]ac[dot]nz) $Revision: 10341 $

15. An implementation of a RIpple-DOwn Rule learner. It generates a default rule first and then the exceptions for the default rule with the least (weighted) error rate. Then it generates the ”best” exceptions for each exception and iterates until pure. Thus it performs a tree-like expansion of exceptions.The exceptions are a set of rules that predict classes other than the default. IREP is used to generate the exceptions. For more information about Ripple-Down Rules, see: Brian R. Gaines, Paul Compton (1995). Induction

of Ripple-Down Rules Applied to Modeling Large Databases. J. Intell. Inf. Syst.. 5(3):211-228. Xin XU (xx5@cs.waikato.ac.nz) $Revision: 8109 $

16. AODE achieves highly accurate classification by averaging over all of a small space of alternative naive-Bayes-like models that have weaker (and hence less detrimental) independence assumptions than naive Bayes. The resulting algorithm is computationally efficient while delivering highly accurate classification on many learning tasks. For more information, see G. Webb, J. Boughton, Z. Wang (2005). Not So Naive Bayes: Aggregating One-Dependence Estimators. Machine Learning. 58(1):5-24. Further papers are available at http://www.csse.monash.edu.au/ webb/. Default frequency limit set to 1. Janice Boughton (jrbought@csse.monas
Zhihai Wang (zhw@csse.monash.edu.au) Nayyar Zaidi (nayyar.zaidi@monash.edu) $Revision: 2 $

17. A2DE achieves highly accurate classification by averaging over all of a small space of alternative naive-Bayes-like models that have weaker (and hence less detrimental) independence assumptions than naive Bayes. The resulting algorithm is computationally efficient while delivering highly accurate classification on many learning tasks. For more information, see G.I. Webb, J. Boughton, F. Zheng, K.M. Ting and H. Salem (2012). Learning by extrapolation from marginal to full-multivariate probability distributions: decreasingly naive Bayesian classification. Machine Learning. 86(2):233-272. Further papers are available at http://www.csse.monash.edu.au/ webb/. Default frequency limit set to 1. Nayyar Zaidi (nayyar.zaidi@monash.edu
Janice Boughton (jrbought@csse.monash.edu.au) @version $Revision: 2 $

18. Classifier for building 'logistic model trees', which are classification trees with logistic regression functions at the leaves. The algorithm can deal with binary and multi-class target variables, numeric and nominal attributes and missing values. For more information see: Niels Landwehr, Mark Hall, Eibe Frank (2005). Logistic Model Trees. Machine Learning. 95(1-2):161-205. Marc Sumner, Eibe Frank, Mark Hall: Speeding up Logistic Model Tree Induction. In: 9th European Conference on Principles and Practice of Knowledge Discovery in Databases, 675-683, 2005. Niels Landwehr Marc Sumner $Revision: 10153 $

19. Class for generating a decision tree with naive Bayes classifiers at the leaves. For more information, see Ron Kohavi: Scaling Up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid. In: Second International Conference on Knoledge Discovery and Data Mining, 202-207, 1996. Mark Hall $Revision: 8109 $

20. Class for constructing a forest of random trees. For more information see: Leo Breiman (2001). Random

Forests. Machine Learning. 45(1):5-32. Richard Kirkby (rkirkby@cs.waikato.ac.nz) $Revision: 10476 $

21. Class for constructing a tree that considers K randomly chosen attributes at each node. Performs no pruning. Also has an option to allow estimation of class probabilities (or target mean in the regression case) based on a hold-out set (backfitting). Eibe Frank (eibe@cs.waikato.ac.nz) Richard Kirkby (rkirkby@cs.waikato.ac.nz) $Revision: 10471 $

22. Classifier for building 'Functional trees', which are classification trees that could have logistic regression functions at the inner nodes and/or leaves. The algorithm can deal with binary and multi-class target variables, numeric and nominal attributes and missing values. For more information see: Joao Gama (2004). Functional Trees. Niels Landwehr, Mark Hall, Eibe Frank (2005). Logistic Model Trees. João Gama Carlos Ferreira $Revision: 8108 $

23. Class for generating a multi-class alternating decision tree using the LogitBoost strategy. For more info, see Geoffrey Holmes, Bernhard Pfahringer, Richard Kirkby, Eibe Frank, Mark Hall: Multiclass alternating decision trees. In: ECML, 161-172, 2001. Richard Kirkby $Revision: 10324$

24. Fast decision tree learner. Builds a decision/regression tree using information gain/variance and prunes it using reduced-error pruning (with backfitting). Only sorts values for numeric attributes once. Missing values are dealt with by splitting the corresponding instances into pieces (i.e. as in C4.5). Eibe Frank (eibe@cs.waikato.ac.nz) $Revision: 10274 $

25. Class implementing minimal cost-complexity pruning. Note when dealing with missing values, use "fractional instances" method instead of surrogate split method. For more information, see: Leo Breiman, Jerome H. Friedman, Richard A. Olshen, Charles J. Stone (1984). Classification and Regression Trees. Wadsworth International Group, Belmont, California. Haijian Shi (hs69@cs.waikato.ac.nz) $Revision: 10490 $

26. Implements John Platt's sequential minimal optimization algorithm for training a support vector classifier.This implementation globally replaces all missing values and transforms nominal attributes into binary ones. It also normalizes all attributes by default. (In that case the coefficients in the output are based on the normalized data, not the original data — this is important for interpreting the classifier.) Multi-class problems are solved using pairwise classification (1-vs-1 and if logistic models are built pairwise

coupling according to Hastie and Tibshirani, 1998). To obtain proper probability estimates, use the option that fits logistic regression models to the outputs of the support vector machine. In the multi-class case the predicted probabilities are coupled using Hastie and Tibshirani's pairwise coupling method. Note: for improved speed normalization should be turned off when operating on SparseInstances. For more information on the SMO algorithm, see J. Platt: Fast Training of Support Vector Machines using Sequential Minimal Optimization. In B. Schoelkopf and C. Burges and A. Smola, editors, Advances in Kernel Methods - Support Vector Learning, 1998. S.S. Keerthi, S.K. Shevade, C. Bhattacharyya, K.R.K. Murthy (2001). Improvements to Platt's SMO Algorithm for SVM Classifier Design. Neural Computation. 13(3):637-649. Trevor Hastie, Robert Tibshirani: Classification by Pairwise Coupling. In: Advances in Neural Information Processing Systems, 1998. Eibe Frank (eibe@cs.waikato.ac.nz) Shane Legg (shane@intelligenesis.net) (sparse vector code) Stuart Inglis (stuart@reeltwo.com) (sparse vector code) $Revision: 10141 $

27. Class for building and using a multinomial logistic regression model with a ridge estimator. There are some modifications, however, compared to the paper of leCessie and van Houwelingen (1992) : If there are k classes for n instances with m attributes, the parameter matrix B to be calculated will be an m*(k-1) matrix. The probability for class j with the exception of the last class is Pj(Xi) = exp(XiBj)/((sum[j=1..(k-1)]exp(Xi*Bj))+1) The last class has probability 1-(sum[j=1..(k-1)]Pj(Xi))= 1/((sum[j=1..(k-1)]exp(Xi*Bj))+1) The (negative) multinomial log-likelihood is thus: L = -sum[i=1..n]{sum[j=1..(k-1)](Yij * ln(Pj(Xi))) +(1 - (sum[j=1..(k-1)]Yij))* ln(1 - sum[j=1..(k-1)]Pj(Xi)) } + ridge * (B^2). In order to find the matrix B for which L is minimised, a Quasi-Newton Method is used to search for the optimized values of the m*(k-1) variables. Note that before we use the optimization procedure, we 'squeeze' the matrix B into a m*(k-1) vector. For details of the optimization procedure, please check weka.core.Optimization class. Although original Logistic Regression does not deal with instance weights, wemodify the algorithm a little bit to handle the instance weights. For more information see: le Cessie, S., van Houwelingen, J.C. (1992). Ridge Estimators in Logistic Regression. Applied Statistics. 41(1):191-201. Xin Xu (xx5@cs.waikato.ac.nz) $Revision: 10540 $

28. Trains a multilayer perceptron with one hidden layer using WEKA's Optimization class by minimizing the squared error plus a quadratic penalty with the BFGS method. Note that all attributes are standardized. There are several parameters. The ridge parameter is used to determine the penalty on the size of the weights. The number of hidden units can also be specified. Note that large numbers produce long training times. Finally, it is possible to use conjugate gradient descent rather than BFGS updates, which may be faster for cases with many parameters. To improve speed, an approximate version of the logistic function

is used as the activation function. Also, if delta values in the backpropagation step are within the user-specified tolerance, the gradient is not updated for that particular instance, which saves some additional time. Paralled calculation of squared error and gradient is possible when multiple CPU cores are present. Data is split into batches and processed in separate threads in this case. Note that this only improves runtime for larger datasets. Nominal attributes are processed using the unsupervised NominalToBinary filter and missing values are replaced globally using ReplaceMissingValues. Eibe Frank (eibe@cs.waikato.ac.nz) $Revision: 9345 $

9 Μελλοντικές Επεκτάσεις

Στο μέλλον θα μπορούσαν να μελετηθούν οι ίδιοι ή και περισσότεροι αλγόριθμοι με ποικίλες τροποποιήσεις στη μεθοδολογία. Θα μπορούσε να χρησιμοποιηθεί και το μέγεθος (n = 5) στους n-gram graphs, να δοκιμαστεί Percentage Split με 70%-30% διαχωρισμό, να τροποποιηθούν οι τιμές των default παραμέτρων στους αλγορίθμους και να πειραματιστεί κανείς σε ένα λογικό εύρος τιμών γύρω από αυτές. Φυσικά αυτές οι τροποποιήσεις στις τιμές θα απαιτούσαν αποκλειστικότητα σε έναν εργαστηριακό server επί σειρά εβδομάδων, κάτι δύσκολα εφικτό, ενώ πειράματα σε εύρος τιμών αλγόριθμων όπως οι instance-based αποτελούν μία πολύ επίπονη σε χρόνο και υπολογιστικούς πόρους διαδικασία, κάτι που φαίνεται και από τους πιθανούς συνδυασμούς τιμών παραμέτρων που μπορούν να προκύψουν ανά αλγόριθμο.

Σχήμα 13: Παράδειγμα .arff αρχείου με δεδομένα καιρού στα attributes και κλάση το αν μπορεί να παίξει κανείς έξω

```
% ARFF file for the weather data with some numeric features
%
@relation weather

@attribute outlook { sunny, overcast, rainy }
@attribute temperature numeric
@attribute humidity numeric
@attribute windy { true, false }
@attribute play? { yes, no }

@data
%
% 14 instances
%
sunny, 85, 85, false, no
sunny, 80, 90, true, no
overcast, 83, 86, false, yes
rainy, 70, 96, false, yes
rainy, 68, 80, false, yes
rainy, 65, 70, true, no
overcast, 64, 65, true, yes
sunny, 72, 95, false, no
sunny, 69, 70, false, yes
rainy, 75, 80, false, yes
sunny, 75, 70, true, yes
overcast, 72, 90, true, yes
overcast, 81, 75, false, yes
rainy, 71, 91, true, no
```

10 Αναφορές

10.1 Βιβλία

[1] *Data Mining – Practical Machine Learning Tools and Techniques , Third Edition.*
Ian H. Witten Eibe Frank , Mark A. Hall.
Copyright © 2011 Elsevier Inc.
url: http://www.cs.waikato.ac.nz/ml/weka/book.html

[2] *Εισαγωγή Στη Θεωρία Πιθανοτήτων Και Στατιστική , Τέταρτη Έκδοση.*
Γ.Κοκολάκης (Καθηγητής) , Ι.Σπηλιώτης (Επ.Καθηγητής).
Copyright © 1999 Εκδ. Συμεών.

10.2 Δημοσιεύσεις

[1] *Sentiment analysis of social media content using N-Gram graphs.*
Fotis Aisopos, George Papadakis, Theodora Varvarigou.
Copyright 2011 ACM WSM'11,November 30, 2011, Scottsdale, Arizona, USA.
url: http://dl.acm.org/citation.cfm?id=2072614
url: http://scholar.google.gr/citations?view_op=view_citation&hl=el&user=g5_q1R4AAAAJ
&sortby=pubdate&citation_for_view=g5_q1R4AAAAJ:d1gkVwhDpl0C

[2] *Content vs. context for sentiment analysis: a comparative analysis over microblogs.*
Fotis Aisopos, George Papadakis, Konstantinos Tserpes, Theodora Varvarigou.
Copyright 2012 ACM HT'12, June 25–28, 2012, Milwaukee, Wisconsin, USA.
url: https://dl.acm.org/citation.cfm?id=2310028

[3] *Very Simple Classification Rules Perform Well on Most Commonly Used Datasets.*
Robert C. Holte.
Published in: Journal Machine Learning Volume 11 Issue 1, April 1993 , Pages 63-90 ,
Kluwer Academic Publishers Hingham, MA, USA.
url: http://dl.acm.org/citation.cfm?id=173574

[4] *Twitter Sentiment Analysis.*
Thomas Lake, Western Michigan University, Kalamazoo, MI.
Client: William Fitzgerald , April 30, 2011

[5] *Twitter Sentiment Classification using Distant Supervision.*
Alec Go , Richa Bhayani , Lei Huang .
Technical report,2009 Stanford.
Kluwer Academic Publishers Hingham, MA, USA.
url: http://cs.stanford.edu/people/alecmgo/papers/TwitterDistantSupervision09.pdf
url: http://help.sentiment140.com/

[6] *Twitter Sentiment Analysis: The Good the Bad and the OMG!.*
Efthymios Kouloumpis , TheresaWilson, Johanna Moore .
Work performed while at the University of Edinburgh.
Copyright © 2011, Association for the Advancement of Artificial Intelligence,
(www.aaai.org). All rights reserved.
url: http://www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/viewFile/2857/3251

[7] *Twitter as a Corpus for Sentiment Analysis and Opinion Mining.*
Alexander Pak, Patrick Paroubek .
In LREC, 2010.

[8] *Recognizing Contextual Polarity: An Exploration of Features for Phrase-Level Sentiment Analysis .*
Theresa Wilson , Janyce Wiebe , Paul Hoffmann .
Published in: Journal Computational Linguistics ,Volume 35 Issue 3,
September 2009 , Pages 399-433 , MIT Press Cambridge, MA, USA
url: https://dl.acm.org/citation.cfm?id=1618330

[9] *Characterizing debate performance via aggregated twitter sentiment .*
Nicholas A. Diakopoulos , David A. Shamma.
Published in: Proceeding CHI '10 Proceedings of the SIGCHI Conference on Human Factors in

Computing Systems , Pages 1195-1198 , ACM New York, NY, USA ©2010.
url: https://dl.acm.org/citation.cfm?doid=1753326.1753504

[10] *Tweet the debates: understanding community annotation of uncollected sources.*
David A. Shamma , Lyndon Kennedy , Elizabeth F. Churchill.
Published in: Proceeding WSM '09 Proceedings of the first SIGMM workshop on Social media , Pages 3-10 , ACM New York, NY, USA ©2009.
url: https://dl.acm.org/citation.cfm?doid=1631144.1631148

[11] *Twitter Polarity Classification with Label Propagation over Lexical Links and the Follower Graph.*
Michael Speriosu, Nikita Sudan, Sid Upadhyay, Jason Baldridge.
Proceedings of EMNLP 2011, Conference on Empirical Methods in Natural Language Processing , pages 53–63, Edinburgh, Scotland, UK, July 27–31, 2011.
© 2011 Association for Computational Linguistics
url: http://www.aclweb.org/anthology/W/W11/W11-22.pdf#page=63
url: https://dl.acm.org/citation.cfm?id=2140465

10.3 Λογισμικό

[1] *Weka 3: Data Mining Software in Java.*
Version 3.7.11 και classification packages από τον Package Manager του.
url: http://www.cs.waikato.ac.nz/ml/weka/
url: http://weka.wikispaces.com/Frequently+Asked+Questions

[2] *JInsect project, Παραγωγή Γράφων N-Γραμμάτων*,
George Giannakopoulos.
url: http://sourceforge.net/projects/jinsect/
url: http://users.iit.demokritos.gr/~ggianna/#Publications

[3] *Γλώσσα προγραμματισμού Java.*
Copyright © Oracle Corporation.

url: https://www.java.com/

[4] *Περιβάλλον προγραμματισμού Eclipse.*
Copyright © 2014 The Eclipse Foundation.
url: https://www.eclipse.org/

10.4 Datasets

[1] *Dataset for Characterizing Debate Performance via Aggregated Twitter Sentiment.*
Nicholas Diakopoulos and David A. Shamma.
url: http://www.ayman-naaman.net/2010/11/21/twitter-sentiment-dataset-online/
url: http://bit.ly/eu72Lr
Βλέπε δημοσιεύεσεις [9] και [10].

[2] *Updown package.*
Michael Speriosu, Nikita Sudan, Sid Upadhyay and Jason Baldridge.
url: https://bitbucket.org/speriosu/updown/get/1deb8fe45f60.zip
(data φάκελος, αρχεία: hcr-train.csv,hcr-test.csv,hcr-dev.csv)
url: https://bitbucket.org/speriosu/updown/wiki/Home
Βλέπε δημοσίευση [11].

[3] *Twitter Sentiment Analysis Training Corpus.*
Collected by Ibrahim Naji.
url: http://thinknook.com/twitter-sentiment-analysis-training-corpus-dataset-2012-09-22/
url: http://thinknook.com/wp-content/uploads/2012/09/Sentiment-Analysis-Dataset.zip
url: http://inclass.kaggle.com/c/si650winter11
url: http://www.sananalytics.com/lab/twitter-sentiment/

10.5 Online Courses

[1] *Waikato Courses , Data Mining with Weka*

Prof Ian H. Witten Department of Computer Science ,University of Waikato.

url: https://weka.waikato.ac.nz/dataminingwithweka/preview

slides,videos,transcripts:

url: http://www.cs.waikato.ac.nz/ml/weka/mooc/dataminingwithweka/

[2] *Waikato Courses , More Data Mining with Weka*

Prof Ian H. Witten Department of Computer Science, University of Waikato.

url: https://weka.waikato.ac.nz/moredataminingwithweka/preview

slides,videos,transcripts:

url: http://www.cs.waikato.ac.nz/ml/weka/mooc/moredataminingwithweka/

[3] *Gerstein Lab Courses : Introduction to Data Mining*

url: http://www.gersteinlab.org/courses/545/07-spr/outline.html

url: http://info.gersteinlab.org/index.php/Cs545-07