ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΠΑΤΡΩΝ

Σχολή Διοίκησης και Οικονομίας

Τμήμα Επιχειρηματικού Σχεδιασμού και Πληροφοριακών Συστημάτων

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΑΛΓΟΡΙΘΜΟΙ ΤΑΞΙΝΟΜΗΣΗΣ ΣΤΗΝ ΕΞΟΡΥΞΗ ΔΕΔΟΜΕΝΩΝ

Σπουδαστές: Ζιώβας Θωμάς Μουνδρέας Γεώργιος

Επιβλέπων καθηγητής: Μαστρογιάννης Νικόλαος

ΠΑΤΡΑ-2011

ΠΡΟΛΟΓΟΣ

Η εποχή που διανύουμε χαρακτηρίζεται ως η εποχή της πληροφορίας. Οι διαδικασίες εξαγωγής της γνώσης από την πληροφορία είναι κάτι το οποίο εξελίσσετε ραγδαία τα τελευταία χρόνια. Η συνεχής αύξηση της πληροφορίας αλλά κυρίως η αύξηση των δεδομένων καθιστά επιτακτική την ανάγκη ανάπτυξης νέων τρόπων και τεχνολογιών για τη μετάλλαξη των πληροφοριών και των δεδομένων σε γνώση ευκολότερα, ταχύτερα και ευφυέστερα.

ΠΕΡΙΛΗΨΗ

Η παρούσα εργασία με αντικείμενο τους αλγόριθμους ταξινόμησης στην εξόρυξη δεδομένων ξεκινά με την επεξήγηση για την εξαγωγή της πληροφορίας, τον ορισμό της εξόρυξης δεδομένων και την ανεύρεση γνώσης, όπου γίνεται ανάλυση των εσωτερικών πηγών δεδομένων, των χαρακτηριστικών των δεδομένων καθώς και των τεχνικών εξόρυξης. Στη συνέχεια του πρώτου κεφαλαίου παρουσιάζεται η αλλαγή στον τύπο έκφρασης των ερωτήσεων και των αποτελεσμάτων της εξόρυξης δεδομένων, η διαδικασία και η περιγραφή των χαρακτηριστικών των αλγορίθμων εξόρυξης και της εξόρυξης από διαφορετικές πηγές δεδομένων. Ακολουθεί η ομαδοποίηση με τον ορισμό, τα κριτήρια και τις κατηγορίες αλγορίθμων ομαδοποίησης, η συσχέτιση όπου περιγράφονται οι κανόνες συσχέτισης αλλά και η διαδικασία εξαγωγής κανόνων συσχέτισης.

Περιεχόμενα

Κεφάλαιο 1 4
1.1 Εισαγωγή 4
1.2 Ορισμός εξόρυξης δεδομένων 4
1.2.1 Εξόρυξη δεδομένων και ανεύρεση γνώσης 5
1.2.1.1 Αλλαγή στο τύπο έκφρασης των ερωτήσεων και των αποτελεσμάτων της εξόρυξης δεδομένων 8
1.2.1.2 Διαδικασία 9
1.2.1.3 Εξόρυξη γνώσης από διαφορετικές πηγές δεδομένων 10
1.3 Ομαδοποίηση 10
1.4 Συσχέτιση 12
1.4.1 Εξαγωγή κανόνων συσχέτισης 12
1.5 Ταξινόμηση 14
1.6 Παλινδρόμηση 15
1.7 Απόδοση και εξελισσιμότητα αλγορίθμων 15
1.8 Χρησιμότητα και βεβαιότητα των αποτελεσμάτων της εξόρυξης 16
1.8.1 Ομοιότητα χρονολογικών σειρών 16
1.8.2 Απεικόνιση και μείωση διαστάσεων 16
Κεφάλαιο 2 18
2.1 Είδη ταξινόμησης 18
2.1.1 Δέντρα απόφασης 18
2.1.2 Μέθοδοι κανόνων απόφασης 21
2.1.3 Τεχνητά Νευρωνικά Δίκτυα 23
2.1.4 Στατιστικές μέθοδοι ταξινόμησης 26
2.1.4.1 Αφελής ταξινομητής Bayes 26
2.1.5 Δίκτυα Bayes 28
2.1.6 Μέθοδοι μάθησης κατά περίπτωση 29
2.1.7 Μηχανές Διανυσμάτων Υποστήριξης 30
2.1.8 Λοιπές μέθοδοι και αλγόριθμοι 32
Κεφάλαιο 3 33
3.1 Βασικοί αλγόριθμοι ταξινόμησης 33
3.1.1 Βασικοί αλγόριθμοι δέντρων απόφασης 33
3.1.1.1 Αλγόριθμος ID3 33
3.1.1.2 Αλγόριθμος C4.5 41
3.1.1.3 SLIQ 44
3.1.1.4 SPRINT 45
3.1.1.5 CART 46
3.1.2 Αλγόριθμοι στις μεθόδους κανόνων απόφασης 48
3.1.2.1 Ο αλγόριθμος CN2 48
3.1.2.2 Αλγόριθμος AQ 50
3.1.2.3 Ο αλγόριθμος CL^2 51
3.1.2.4 Κανόνες εκμάθησης 52
3.1.3 Μέθοδοι μάθησης κατά περίπτωση 59
3.1.3.1 K-κοντινότερος Γείτονας 59
3.1.3.2 Μηχανές διανυσμάτων υποστήριξης 60

3.1.3.3 Ώθηση καθορισμού του περιθωρίου (margin).....61
3.1.3.4 Παραδείγματα αλγορίθμων βασισμένα σε περιθώριο.....62
3.1.4 ADABOOST.....62
3.1.5 EM.....63
ΚΕΦΑΛΑΙΟ 4.....65
4.1 Εφαρμογές ταξινόμησης.....65
4.2 Περιγραφή Βάσεων.....67
4.3 Μεθοδος αξιολόγησης ταξινόμησης.....69
4.4 Πινάκες αποτελεσμάτων.....70
4.5 Αξιολόγηση αποτελεσμάτων.....74
4.6 Σημασία αποτελεσμάτων για κάθε βάση.....75
ΚΕΦΑΛΑΙΟ 5.....77
5.1 Σύνοψη.....77
5.2 Μελλοντικές προοπτικές της ταξινόμησης.....79
ΠΑΡΑΡΤΗΜΑ Α.....81
Α.1 Διεπαφή εφαρμογής.....82
Α.1.1 Δημιουργία νέου project.....83
Α.1.2 Άνοιγμα project.....83
Α.1.3 Επιλογή παλαιάς εκτελέσεις.....84
Α.1.4 Αποθήκευση project.....84
Α.1.5 Εύρεση ακρίβειας.....84
Α.1.6 Δημιουργία κατάστασης κανόνων.....86
Α.2 Εξόρυξη δεδομένων.....87
Α.2.1 Εφαρμογές ταξινόμησης (classification).....87
Α.2.1.1 CN2.....87
Α. 2.1.2 C4.5.....91
Διεθνής βιβλιογραφία.....97
Ελληνική βιβλιογραφία.....103

Κεφάλαιο 1

1.1 Εισαγωγή

Η εποχή που διανύουμε χαρακτηρίζεται ως η εποχή της πληροφορίας. Οι διαδικασίες εξαγωγής της γνώσης από την πληροφορία είναι κάτι το οποίο εξελίσσετε ραγδαία τα τελευταία χρόνια. Η συνεχής αύξηση της πληροφορίας αλλά κυρίως η αύξηση των δεδομένων καθιστά επιτακτική την ανάγκη ανάπτυξης νέων τρόπων και τεχνολογιών για τη μετάλλαξη των πληροφοριών και των δεδομένων σε γνώση ευκολότερα, ταχύτερα και ευφυέστερα.

Η εξαγωγή κρυμμένης πληροφορίας από βάσεις δεδομένων, χαρακτηρίζεται ως μια δυναμική και μεγάλη νέα τεχνολογία η οποία δίνει τη δυνατότητα σε οργανισμούς και επιχειρήσεις να εκμεταλλευτούν τις σημαντικές πληροφορίες που έχουν στις αποθήκες δεδομένων (data warehouse). Έτσι η διαδικασία αναγνώρισης έγκυρων, καινοτόμων, ίσως χρήσιμων και εν τέλει κατανοητών προτύπων στα δεδομένα καθιστάτε επιτακτική.

1.2 Ορισμός εξόρυξης δεδομένων

Η εξόρυξη δεδομένων είναι μια διαδικασία εξαγωγής κρυμμένης πληφοροφίας από μεγάλες βάσεις δεδομένων. Ένας πιο αυστηρός και τυπικός ορισμός της εξόρυξης δεδομένων, λαμβάνοντας υπόψη τους ορισμούς που δόθηκαν κατά καιρούς από τους Piatetsky-Shapiro & Frawley (1991), Piatetsky-Shapiro et al (1996) καθώς και τους Cabena et al (1998) και Hand et al (2001) όπως αυτοί αναφέρονται στο «Έννοιες και Αλγόριθμοι της Εξόρυξης δεδομένων» .

Σύμφωνα με τον Larose (2004; 2006), είναι ο εξής: «*Εξόρυξη δεδομένων είναι η διαδικασία εξαγωγής υπονοούμενης και εν πολλοίς άγνωστης, αλλά ενδεχομένως χρήσιμης γνώσης, υπό την μορφή συσχετίσεων, προτύπων και τάσεων, μέσω της εξέτασης, ανάλυσης και επεξεργασίας βάσεων δεδομένων, συνδυάζοντας και χρησιμοποιώντας τεχνικές από την μηχανική μάθηση, την αναγνώριση προτύπων, την στατιστική, τις βάσεις δεδομένων και την οπτικοποίηση*».

Παρά το γεγονός ότι υπάρχει μια γενικότερη συμφωνία ότι στόχος της εξόρυξηςδεδομένων είναι η ανακάλυψη νέας και χρήσιμης πληροφορίας σε βάσεις δεδομένων, τα μέσα για την επίτευξη του στόχου αυτού, ποικίλουν σε πολύ υψηλό βαθμό. Η εξόρυξη δεδομένων περιλαμβάνει ένα ευρύ πεδίο υπολογιστικών μεθόδων που μεταξύ άλλων περιλαμβάνουν, την *στατιστική ανάλυση* (*statistical analysis*), τα *δέντρα αποφάσεων* (*decision trees*), τα *νευρωνικά δίκτυα* (*neural networks*), την *εξαγωγή κανόνων* (*rule induction*) και την *γραφική οπτικοποίηση* (*graphic visualization*).

Τέτοιες μέθοδοι χρησιμοποιούνται για την εύρεση συσχετίσεων, προτύπων και δομών, σε μεγάλες και διαρκώς αυξανόμενες βάσεις δεδομένων. Ειδικά η εύρεση προτύπων είναι ένα ιδιαίτερα σημαντικό εξαγόμενο της εξόρυξης δεδομένων, αυτά δε περιγράφονται, μεσω σχέσεων μεταξύ των *χαρακτηρικών* (*attributes*) των βάσεων δεδομένων. Αξίζει επίσης να σημειώσουμε ότι η εξόρυξη δεδομένων δεν εξειδικεύεται σε ένα μόνο τύπο δεδομένων. Ωστόσο, οι αλγόριθμοι της, τόσο σε θεωρητικό όσο και σε πρακτικό επίπεδο, μπορεί να διαφέρουν εφαρμοζόμενοι σε διαφορετικά είδη δεδομένων.

1.2.1 Εξόρυξη δεδομένων και ανεύρεση γνώσης

Στην διεθνή βιβλιογραφία υπάρχει μια γενικότερη σύγχυση ανάμεσα στους όρους «Εξόρυξη δεδομένων» και «Ανεύρεση γνώσης στις βάσεις δεδομένων» (Knowledge discovery in databases). Σε πολλές περιπτώσεις οι δύο όροι ταυτίζονται.

Η συλλογή της πληροφορίας συντελείται μέσα από εσωτερικές και εξωτερικές πηγές. Ωστόσο η χρησιμοποίηση εξωτερικών πηγών μπορεί να κολλήσει σε προβλήματα νομικής φύσης που αφορούν στη μεταφορά, τη χρησιμοποίηση και το συνδυασμό πληροφοριών. Επιπρόσθετα θα πρέπει να επιλέξουμε την πληροφορία που θα χρειαστούμε με βάση ένα συγκεκριμένο κριτήριο. Ως εσωτερικές πήγες δεδομένων θεωρούμε:

τις αποθήκες δεδομένων (data warehouses)
τις σχεσιακές βάσεις δεδομένων (Relation Databases)
οι αποθήκες πληροφοριών (Information Repositories) και
οι προηγμένες βάσεις δεδομένων (Advanced Databases).

Τα δεδομένα πρέπει να ελεγχθούν για τριχών διαφορές ασάφειες, παραλήψεις ή ελλείψεις και να διορθωθούν όπου παρουσιάζουν τέτοια προβλήματα. Αυτή η προεπεξεργασίας δεδομένων καταλαμβάνει το 60% της συνολικής διαδικασίας εύρεσης γνώσης από δεδομένα με στόχο να στηριχθεί σε ποιοτικές πληροφορίες και δεδομένα.

Δηλαδή να διαθέτουν τα παρακάτω χαρακτηριστικά:

Πληρότητα (Completeness)
Συνέπεια (Consistency)
Ακρίβεια (Accuracy)
Επικαιρότητα (Timeliness)
Αξιοπιστία (Believability)
Προστιθέμενη Αξία (Added Value)
Ερμηνευτικότητα (Interpretability)
Προσιτότητα (Accessibility)

Η εξόρυξη δεδομένων (data mining) προβλέπει τις μέλλουσες τάσεις και συμπεριφορές δίνοντας τη δυνατότητα στην επιχειρήσει που τη χρησιμοποιεί να πάρει

αποφάσεις μέσα από τη γνώση, να απαντήσει ταχύτερα σε παραδοσιακά επιχειρηματικά ερωτήματα και τελικά να αναζητήσει λεπτομερειακά κρυμμένα πρότυπα (patterns), βρίσκοντας πληροφορίες που δύσκολα θα ανακαλύπτονταν με άλλες μεθόδους.

Η εξόρυξη γνώσης αποτελεί συνδυασμό διάφορων επιστημονικών πεδίων όπως η στατιστική, η μηχανική μάθηση (Machine Learning), οι βάσεις δεδομένων και η οπτικοποίηση (Visualization). Παρακολουθεί την πρόοδο και την εξέλιξη που συντελείται σε πεδία όπως η τεχνητή νοημοσύνη (Artificial Intelligence) και η στατιστική. Παρολ΄αυτά δεν μπορεί να αντικαταστήσει τη στατιστική θεωρία διότι αναζητεί πρότυπα από καθοδηγούμενα δεδομένα, ενώ η στατιστική καθοδηγείτε από ένα επαληθευτικό μοντέλο και εξαρτάται από μια υπόθεση.

Η εξόρυξη δεδομένων και η ανακάλυψη γνώσης από βάσεις δεδομένων είναι δυο διαφορετικές έννοιες με την εξόρυξη δεδομένων να αποτελεί το βασικό υποσύνολο της διαδικασίας ανακάλυψης γνώσης. Για την απόκτηση γνώσης από μια βάση δεδομένων πρέπει να ακολουθεί μια συγκεκριμένη επαναληπτική διαδικασία η οποία και ονομάζετε ανακάλυψη γνώσης στις βάσεις δεδομένων, με στόχο την εξόρυξη γνώσης και μοντέλων μέσα από τα δεδομένα Βασικές εφαρμογές της εξόρυξης δεδομένων συναντώνται σε διάφορους επιχειρηματικούς κλάδους όπως:

Στην ιατρική (πχ. χαρακτηρισμός συμπεριφοράς ασθενών για την πρόβλεψη ανάγκης χειρούργησης και αξιολόγηση θεραπειών για διάφορες παθήσεις), τις ασφάλειες (Πρόβλεψη/εκτίμηση ποιοι πελάτες θα αγοράσουν και άλλες υπηρεσίες), το χρηματοπιστωτικό σύστημα (Χορήγηση πιστωτικής κάρτας Πρόβλεψη/Εκτίμηση ποιοι πελάτες είναι πιστοί ή ποιοι υπάρχει μεγάλη πιθανότητα να φύγουν) και το μάρκετινγκ (Εύρεση δημογραφικών συσχετισμών για τους πελάτες, Πρόβλεψη της ανταπόκρισης σε μια προσφορά / προώθηση προϊόντος). Οι κυριότεροι στόχοι της εξόρυξης δεδομένων στην πράξη τείνουν να είναι η προβλεψημότητα και η περιγραφικότητα. Η πρόβλεψη χρησιμοποιεί την παλινδρόμηση τεχνική που δανείζεται από τη στατιστική θεωρία, πχ. Πρόβλεψη πωλήσεων νέου προϊόντος

δοθείσας της τιμής. Οι στόχοι τη προβλεψημότητας και της περιγραφικότητας μπορούν να επιτευχθούν χρησιμοποιώντας μερικές μεθόδους εξόρυξης δεδομένων όπως:

Συσταδοποίηση/ομαδοποίηση (clustering)
Συσχέτιση (association rules) και
Ταξινόμηση (classification).
Παλινδρόμηση

Η γνώση που έχει εξορυχτεί θα πρέπει να παρουσιάζει τα περιεχόμενα των βάσεων δεδομένων με ακριβές τρόπο. Η ακρίβεια αυτή θα μπορούσε να εκφραστεί μέσω των μέτρων βεβαιότητας. Εξαιρέσεις όπως ο λεγόμενος θόρυβος και οι outliers θα πρέπει να αντιμετωπισθούν αποτελεσματικά από τα συστήματα εξόρυξης. Έτσι δημιουργείτε ένα κίνητρο για τη συστηματική μελέτη της ποιότητας της γνώσης, των αναλυτικών μοντέλων, των μοντέλων προσομοίωσης καθώς και των εργαλείων.

1.2.1.1 Αλλαγή στο τύπο έκφρασης των ερωτήσεων και των αποτελεσμάτων της εξόρυξης δεδομένων

Από μεγάλα σύνολα δεδομένων μπορούν να εξαχθούν διαφορετικοί τύποι γνώσεων. Επίσης καλό θα ήταν να μπορούμε να εξετάσουμε τη γνώση μέσα από διαφορετικές απόψεις και διαφορετικές μορφές. Έτσι δημιουργείτε η ανάγκη να εκφράσουμε τις επερωτήσεις εξόρυξης δεδομένων και η εξορυγμένη γνώση σε γλώσσες υψηλού επίπεδου προκειμένου η όλη διαδικασία εξόρυξης να είναι εφαρμόσιμη από μη ειδικούς και να μπορούν εύκολα οι χρήστες να χρησιμοποιήσουν την εξορυγμένη γνώση.

1.2.1.2 Διαδικασία

Το μεγαλύτερο μέρος των αλγορίθμων εξόρυξης δεδομένων μπορεί να περιγράφει σε υψηλό επίπεδο με τον όρο ενός απλού πλαισίου δηλαδή να αντιμετωπισθούν ως σύνθεση των τριών παρακάτω χαρακτηριστικών:

1. Η περιγραφή του μοντέλου

Υπάρχουν δυο παράγοντες που σχετίζονται με το μοντέλο. Η λειτουργία η οποία καθορίζει τους βασικούς στόχους κατά τη διάρκεια της διαδικασίας πχ. Clustering και η παραστατική μορφή του μοντέλου. Η απεικόνιση του μοντέλου καθορίζει το ταίριασμα με την απεικόνιση των δεδομένων και τη δυνατότητα να ερμηνεύσουμε το μοντέλο με τους κατάλληλους όρους. Τα πιο γνωστά μοντέλα θεωρούνται τα δέντρα, τα νευρωνικά δίκτυα, τα μοντέλα βασισμένα σε πιθανότητες κ.α.

2. Αξιολόγηση του μοντέλου

Χρησιμοποιώντας κάποια κριτήρια αξιολόγησης μπορούμε να καθορίσουμε πόσο ταιριάζει το μοντέλο που επιλέξαμε με τα κριτήρια της διαδικασία εξόρυξης γνώσης από δεδομένα. Γενικότερα στην αξιολόγηση του μοντέλου αναφερόμαστε τόσο στην εγκυρότητα των προτύπων όσο και στην αξιολόγηση της ακρίβειας, της χρησιμότητας και της δυνατότητας να κατανοήσουμε το μοντέλο.

3. Αλγόριθμοι αναζήτησης

Αναφέρετε στη δυνατότητα ενός αλγορίθμου να εντοπίζει μοντέλα και παραμέτρους σε ένα συγκεκριμένο σύνολο δεδομένων. Υπάρχουν δυο τέτοιοι τύποι:

α) Αυτοί που αναζητούν παραμέτρους που να βελτιστοποιούν ένα κριτήριο αξιολόγησης του μοντέλου και

β) Αυτοί που εκτελούν μοντέλα κάνοντας μια επαναληπτική διαδικασία αναζήτησης για την αντιπροσώπευση των δεδομένων.

1.2.1.3 Εξόρυξη γνώσης από διαφορετικές πηγές δεδομένων

Η μεγάλη διάδοση της σύνδεσης υπολογιστών συμπεριλαμβανομένου και του διαδικτύου προηγείται στη σύνδεση διαφόρων πηγών δεδομένων με αποτέλεσμα να δημιουργούνται μεγάλες κατανεμημένες και ετερογενείς βάσεις δεδομένων. Το μεγάλο ποσό δεδομένων, η υψηλή κατανομή και η υπολογιστική πολυπλοκότητα μας οδηγούν στην ανάπτυξη παράλληλων και κατανεμημένων αλγορίθμων εξόρυξης.

1.3 Ομαδοποίηση

Η ομαδοποίηση είναι μια κοινή περιγραφική λειτουργία η οποία ζητά να ερευνήσει τα δεδομένα μέσα από ένα πεπερασμένο σύνολο κατηγόριων (Jain and Dubes 1988; Titterington, Smith, and Makov 1985). Συγκεκριμένα ως ομάδα χαρακτηρίζετε ένα σύνολο με δεδομένα τα οποία τείνουν να είναι όμοια μεταξύ τους και ανόμοια με δεδομένα άλλων ομάδων. Το μέτρο ομοιότητας καθορίζεται από τη συνάρτηση απόστασης. Οι κλάσεις δεν προσδιορίζονται έκ των πρότερων όπως στην ταξινόμηση αλλά ούτε και δύνονται παραδείγματα που να υποδεικνύουν τις έγκυρες και επιθυμητές σχέσεις ανάμεσα στα δεδομένα της ομάδας. Η διαδικασία της ομαδοποίησης υλοποιείτε μέσα από τέσσερα σταδία. Αρχικά γίνετε η επιλογή των κοινών χαρακτηριστικών της ομάδας, μετά επιλέγουμε τον αλγόριθμο που θα πραγματοποιήσει την ομαδοποίηση, σύμφωνα με το μέτρο εγγύτητας (δείχνει ποσοτικά την ομοιοτητα/ανομιοτητα μεταξύ χαρακτηριστικών διανυσμάτων) και το κριτήριο ομαδοποίησης. Στη συνέχεια πραγματοποιείται ο έλεγχος αξιοπιστίας των αποτελεσμάτων και τέλος η ερμηνεία τους.

Οι βασικότερες μέθοδοι ομαδοποίησης κατηγοριοποιούνται με βάση δυο κριτήρια

- Οι τεχνικές που χρησιμοποιούνται με στόχο τον καθορισμό ομάδων και
- Το είδος των μεταβλητών (αριθμητικά /στατιστικά, εννοιολογικά /κατηγορικά).

Οι τέσσερις κυριότερες μέθοδοι ομαδοποίησης με βάση τα δυο παραπάνω κριτήρια είναι οι :

1. Μέθοδοι διαμερισμού (partitioning methods)
2. Οι ιεραρχικές (hierarchy methods)
3. Οι βασιζόμενοι στην πυκνότητα (density-based) και
4. Οι βασιζόμενοι σε πλέγμα (grid-based).

Οι κυριότεροι αλγόριθμοι για τις παραπάνω κατηγορίες είναι οι:

- K-means
- K-windows
- PAM (Partitioning Around Medoid)
- CLARA (Clustering LARge Applications) και
- CLARANS ("Randomized" CLARA)

Για τη μέθοδο διαμερισμού είναι οι ακόλουθοι :

- Agnes – Agglomerative method, Diana – Divisive method
- BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies)
- CURE (Clustering using Representatives)
- ROCK (Robust Clustering using Links)
- CHAMELEON (Hierarchical Clustering using Dynamic Modeling)

Για την μέθοδο ιεραρχικής ομαδοποίησης είναι οι ακόλουθοι :

- DBSCAN (Density Based Spatial Clustering of Applications with Noise)
- DENCLUE
- GRID – BASED CLUSTERING
- STING (Statistical Information Grid Approach) και
- Wave Cluster για τις βασιζόμενες στη πυκνότητα μεθόδους
- και τέλος ο Fuzzy Clustering για τη βασιζόμενη σε πλέγμα μέθοδο.

1.4 Συσχέτιση

Η συσχέτιση βασίζεται στην εύρεση των συχνότερα εμφανιζόμενων προτύπων, συσχετισμών ή αιτιολογικών δομών και αλληλεξαρτήσεων ανάμεσα στα πεδία μιας σχεσιακής βάσης δεδομένων ή άλλων πληροφοριών. Η συνάρτηση συσχέτισης έχει ως αποτέλεσμα την εύρεση προτύπων και σχέσεων Ανάμεσα σε δοθέντα αντικείμενα πχ προϊόντα και ενός συνόλου εγγραφών που περιέχουν συγκεκριμένο αριθμό αυτών. Τα πρότυπα αυτά εκφράζονται μέσω κανόνων (rules) πχ. Το 50% των εγγραφών που περιέχουν τα αντικείμενα Y, Z να περιέχουν και τα αντικείμενα W, X . το ποσοστό στα παράδειγμα αυτό ονομάζεται παράγων αξιοπιστίας του κανόνα (confidence factor).

Μερικά παραδείγματα καθημερινής εφαρμογής είναι η επεξεργασία ερωτηματολογίων, οι πολλαπλές ιατρικές εξετάσεις, η εύρεση σε ένα κείμενο λέξεων που συνδέονται και άλλα. Οι ποιο γνωστές μέθοδοι δημιουργίας κανόνων συσχέτισης είναι:

- Η αρχή apriori
- Η partitition technique
- Η sampling technique
- multy-level association
- quantitative association rule mining και
- Η constraint-based ή query-based association.

Η σημαντικότερη εξ αυτών θεωρείται ο αλγόριθμος apriori (Αδαμίδης, 2009).

1.4.1 Εξαγωγή κανόνων συσχέτισης

Θεωρείτε μια από τις σημαντικότερες διεργασίες της εξόρυξης. Έτσι προσελκύει μεγάλο ενδιαφέρον καθώς παρέχουν ένα σύντομο και περιεκτικό τρόπο ώστε να

εκφραστούν οι χρήσιμες πληροφορίες που θα γίνονται εύκολα κατανοητές από τον τελικό χρήστη.

Οι κανόνες αυτοί βρίσκουν τις κριμένες συσχετίσεις μεταξύ των γνωρισμάτων του συνόλου δεδομένων. Οι συσχετισμοί είναι της μορφής
A→B όπου A και B τα σύνολα μερισμάτων των υπό ανάλυση δεδομένων.
Υποθέτουμε ότι μας δίνετε ένα σύνολο συναλλαγών S=[S1,S2,…,Sn] όπου κάθε συναλλαγή Si είναι ένα υποσύνολο του A=[A1,A2,…,Ak] (Ai, i=1,2,…,k , είναι οι ιδιότητες του συνόλου δεδομένων). Για ένα δεδομένα σύνολο AjA, η υποστήριξη του A, sup(A), καθορίζετε ώστε να είναι ο αριθμός συναλλαγών στο S που είναι υπερσύνολα του A (δηλαδή το A εμφανίζετε σε αυτές τις συναλλαγές). Εάν η υποστήριξη ενός συνόλου αντικειμένων A είναι μεγαλύτερη από ένα καθορισμένο από το χρήστη κατώτατο όριο υποστήριξης T τότε ονομάζουμε το A ως συχνό σύνολο. Στη συνέχεα μπορούμε να περιγράψουμε το πρόβλημα της εξαγωγής κανόνων συσχέτισης ως εξής:

Λαμβάνοντας υπόψη ένα σύνολο από n συναλλαγές S κάθε υποσύνολο ενός συνόλου A=[A1,A2,…,Aκ] ένα κατώτατο όριο υποστήριξης T και ένα κατώτατο όριο εμπιστοσύνης s, παράγονται όλοι οι κανόνες A→B όπου AjA, BjA, $A \cap B = 0$, $\sup(A \cap B) \geq T$, και $\frac{\sup(A \cap B)}{\sup(A)} \geq s$.

Η σημασία ενός τέτοιου κανόνα είναι ότι οι συναλλαγές στο σύνολο δεδομένων που περιέχουν τις ιδιότητες του A, τείνουν επίσης να περιέχουν τις ιδιότητες του B. σημειώνουμε επίσης ότι οι κανόνες συσχέτισης που εξάγονται πρέπει να μπορούν να ικανοποιούν και άλλους περιορισμούς που καθορίζονται από το χρήστη, σχετικούς με τα μέτρα των κανόνων συσχέτισης.

Λαμβάνοντας υπόψη την παραπάνω περιγραφή μια σημαντική δευτερεύουσα λειτουργιά που συνήθως γίνετε πρώτη είναι αυτή του υπολογισμού των συχνών

συνόλων. Δηλαδή λαμβάνοντας υπόψη ένα σύνολο συναλλαγών S υπολογίζονται όλα τα συχνά υποσύνολα του A (για το δεδομένο κατώτατο όριο υποστήριξης T).

Όταν βρεθούν τα υποσύνολα τα συχνά σύνολα, το πρόβλημα του υπολογισμού των κανόνων συσχέτιση από αυτά γίνετε πιο απλό. Για κάθε συχνό σύνολο A και για κάθε BjA μπορεί να εξεταστεί η εμπιστοσύνη του κανόνα A/B→B.

1.5 Ταξινόμηση

Η ταξινόμηση με την οποία θα ασχοληθούμε διεξοδικά αποτελεί μια από τις δημοφιλέστερες και αποτελεσματικότερες τεχνικές εξόρυξης. Η διαδικασία ταξινόμησης(τεχνικές ή αλγόριθμοι) τοποθετεί μια ομάδα στοιχείων ή εγγραφών σε μια συγκεκριμένη σειρά (αύξουσα ή φθίνουσα) και σε συγκεκριμένες κλάσεις. Στοχος η εξαγωγή προτύπων τα οποία δύναται να χρησιμοποιηθούν για να ταξινομήσουμε δεδομένα με άγνωστη ταξινόμηση στις συγκεκριμένες κλάσεις είτε για να κατανοηθούν καλύτερα οι κλάσεις είτε για να προβλεφθούν συμπεριφορές. Μερικές πρακτικές εφαρμογές της μεθόδου της ταξινόμησης βρίσκονται σε περιπτώσεις αναζήτησης αριθμητικών και αλφαβητικών δεδομένων.

Ποιο συγκεκριμένα σε βιβλιοθηκονομικά συστήματα, τηλεφωνικούς κατάλογους, λεξικά, καταλόγους φόρου εισοδήματος, σε πεδία όπως αποδοχή πιστοληπτικής ικανότητας, άμεσο μάρκετινγκ, διάγνωση ασθενειών, αναγνώριση κατάλληλων πελατών για την αποστολή εντύπων, εύρεση επικινδυνότητας στη χορήγηση δανείων και αλλά. Ένας ειδικότερος ορισμός δίδετε παρακάτω :

Δοθέντων των στοιχείων $\alpha_1,\alpha_2,\ldots,\alpha_n$ η ταξινόμηση συνιστάτε στη διάταξη της θέσης των στοιχείων, ώστε να τοποθετηθούν σε μια σειρά $\alpha_{\kappa 1}$, $\alpha_{\kappa 2}$, …., $\alpha_{\kappa n}$ έτσι ώστε, δοθείσης μιας συνάρτησης διάταξης , f, να ισχύει :

$$f(a_{k1}) \leq f(a_{k2}) \leq \ldots \leq f(a_{kn})$$

Αξίζει να σημειωθεί ότι η προηγούμενη συνάρτηση διάταξης μπορεί να τροποποιηθεί, ώστε να καλύπτει και τη περίπτωση που η ταξινόμηση γίνετε με

φθίνουσα τάξη μεγέθους του κλειδιού. Γενικότερα μπορεί να θεωρηθεί ότι στηρίζετε σε δυο ή περισσότερα κλειδιά της εγγραφής. Για παράδειγμα, σε πολλά παιχνίδια της τράπουλας οι παίκτες ταξινομούν τα χαρτιά τους με πρώτο κλειδί το χρώμα του φύλλου και δεύτερο κλειδί την αξία του φύλλου. Οι τεχνικές ταξινόμησης θα αναλυθούν στο κεφάλαιο 2.

1.6 Παλινδρόμηση

Η παλινδρόμηση αναφέρετε στην εκμάθηση μιας λειτουργιάς που βάζει τα δεδομένα σε μια μεταβλητή πρόβλεψης στην οποία δίνουμε πραγματικές τιμές. Η παλινδρόμηση έχει πολλές εφαρμογές όπως για παράδειγμα ο υπολογισμός της πιθανότητας με την οποία κάποιος ασθενής πρόκειται να αναρρώσει βασιζόμενα στα αποτελέσματα της διάγνωσης που έγινε ή η πρόβλεψη της ζήτησης ενός προϊόντος ως συνάρτηση των δαπανών για διαφήμιση (Βαζιργιάννης & Χαλκίδη, 2003, Breinman et al., 1984).

1.7 Απόδοση και εξελισσιμότητα αλγορίθμων

Για να πετύχουμε εξόρυξη γνώσης με τα επιθυμητά αποτελέσματα από μεγάλα σύνολα δεδομένων θα πρέπει να προσαρμόσουμε τους αλγορίθμους κατάλληλα, πράγμα που σημαίνει ότι ο χρόνος εκτέλεσης πρέπει να είναι αναμενόμενος και αποδεκτός για αυτά τα μεγάλα σύνολα. Αλγόριθμοι όπως αυτοί με εκθετική και πολυωνυμικοί πολυπλοκότητα δεν μπορούν να θεωρούνται κατάλληλοι.

1.8 Χρησιμότητα και βεβαιότητα των αποτελεσμάτων της εξόρυξης

1.8.1 Ομοιότητα χρονολογικών σειρών

Μια χρονολογική σειρά είναι μια ακολουθία αριθμών καθένας από τους οποίους έχει και μια ετικέτα χρόνου. Υποθέτουμε ότι οι διαδοχικοί αριθμοί χωρίζονται από ένα σταθερό χρονικό διάστημα, και η πραγματική ετικέτα χρόνου παραλείπετε. Τα δεδομένα μιας χρονολογικής σειράς βρίσκονται παντού. Διάφορες φυσικές διεργασίες παράγουν δεδομένα υπό μορφή χρονολογικών σειρών οι οποίες εμφανίζονται στον οικονομικό, περιβαλλοντικό τομέα καθώς και στον τομέα της ασφάλειας.

Η διαδικασία εύρεσης χρονολογικών σειρών αποτελεί μείζον ζήτημα της ερευνητικής κοινότητας. Το κυρίως πρόβλημα που εξετάζετε είναι:

Δοσμένης μιας βάσης δεδομένων D με χρονολογικές σειρές και μιας ερώτησης Q (που δεν εμπεριέχετε ακόμα στη βάση δεδομένων) να βρεθεί η πιο κοντά στη Q ακολουθία D.

Το να απαντηθούν οι ερωτήσεις αυτές θα χρησιμοποιηθεί στην κατηγοριοποίηση καινούργιων χρονολογικών σειρών, ή έτσι ώστε ένα σύνολο δεδομένων χρονολογικών σειρών να αναλύετε σε απευθείας σύνδεση. Για απάντηση πρέπει να καθοριστεί μια συνάρτηση απόστασης κοντά στην πραγματικότητα αλλά και κοντά στην αντίληψη του χρήστη για το τι θεωρεί παρόμοιο και ένα αποδεκτό σχέδιο ευρετηρίασης με το οποίο θα γίνονται με μεγαλύτερη ταχύτητα οι ερωτήσεις από τους χρήστες.

1.8.2 Απεικόνιση και μείωση διαστάσεων

Η απεικόνιση των μεγάλων διαστάσεων συνόλων δεδομένων είναι μια αρκετά δύσκολη διαδικασία με το χρήστη να δυσκολεύετε να κατανοήσει πώς κατανέμονται

τα δεδομένα υψηλών διαστάσεων. Υπάρχουν αρκετές τεχνικές που χρησιμοποιούνται οπός αυτή των παράλληλων συντεταγμένων. Παρακάτω θα επικεντρωθούμε στις τεχνικές μείωσης των διαστάσεων. Η βασικότερη ιδέα είναι να μειωθούν οι διαστάσεις του χώρου.

Για να επιτευχτεί αυτό θα πρέπει να προβληθούν n διαστατά σύνολα που αντιπροσωπεύουν κάθε αντικείμενο σε ένα κ διάστατο χώρο. Με το (κ <<n) έτσι ώστε οι αποστάσεις να παραμένουν ίδιες όσο είναι δυνατόν. Αν κ=2 ή 3 μπορούμε να χρησιμοποιήσουμε τις κλασικές τεχνικές για να απεικονίσουμε το σύνολο των δεδομένων. Οι τεχνικές για τη μείωση των διαστάσεων υπολογίζουν μια μικρότερη αντιπροσώπευση του αρχικού συνόλου. Κάποιες πληροφορίες χάνονται όταν επιλέγετε μια μικρότερη αντιπροσώπευση. Παρόλα αυτά οι τεχνικές προσπαθούν να διατηρούν και να βρίσκονται όσο το δυνατόν πιο κοντά στην αρχική δομή. Μετά από αυτά διακρίνουμε δυο βασικές κατηγορίες την τοπική ή σχηματική συντήρηση και σφαιρική ή τυπολογική συντήρηση.

Στην πρώτη κατηγορία περιλαμβάνονται οι μέθοδοι που δεν εκμεταλλεύονται τις σφαιρικές ιδιότητες του συνόλου δεδομένων αλλά περισσότερο απλοποιούν την αντιπροσώπευση κάθε ακολουθίας ανεξάρτητα από τα υπόλοιπα του συνόλου δεδομένων. Η επιλογή των χαρακτηριστικών γνωρισμάτων πρέπει να είναι τέτοια έτσι ώστε να διατηρούν το περισσότερο μέρος των πληροφοριών του αρχικού σχήματος. Η δεύτερη κατηγορία χρησιμοποιείτε κυρίως για λογούς απεικόνισης και ο πρωταρχικός σκοπός τους είναι να βρουν μια χορική αντιπροσώπευση των αντικειμένων. Βρίσκονται σε αντίθεση με τις προηγούμενες προσεγγίσεις διότι προσπαθούν να βρουν τα χαρακτηριστικά γνωρίσματα κ που ελαχιστοποιούν μια σφαιρική αντικειμενική συνάρτηση (Μανωλόπουλος, 2009).

Κεφάλαιο 2

2.1 Είδη ταξινόμησης

Στο κεφάλαιο αυτό θα αναλύσουμε και θα περιγράψουμε τα είδη και τις βασικές κατηγορίες αλγορίθμων ταξινόμησης. Παρακάτω παραθέτουμε κατά σειρά σημαντικότητας τις βασικές κατηγορίες τεχνικών αλγορίθμων ταξινόμησης :

1. Δέντρα απόφασης
2. Μέθοδοι στους κανόνες απόφασης
3. Τεχνητά νευρωνικά δίκτυα
4. Στατιστικές μέθοδοι
5. Μέθοδοι μάθησης κατά περίπτωση
6. Μηχανές διανυσμάτων υποστήριξης
7. Λοιπές μέθοδοι και αλγόριθμοι

2.1.1 Δέντρα απόφασης

Τα δέντρα απόφασης (***decision trees***) είναι μια από τις πιο σημαντικές και ευρύτατα διαδεδομένες μεθόδους για την ταξινόμηση δεδομένων. Σύμφωνα με τους Quinlan (1986,1987, 1993) και Murthy (1998), τα δέντρα απόφασης (βλέπε Σχήμα 2.2) είναι δομές που ταξινομούν τα αντικείμενα μιας βάσης δεδομένων βάσει των τιμών των χαρακτηριστικών αυτών, κατασκευάζονται δε με βάση ένα σύνολο εκπαίδευσης, το οποίο περιλαμβάνει προ-ταξινομημένα δεδομένα. Κάθε κόμβος του δέντρου (Κ1, Κ2, Κ3, Κ4 στο Σχήμα 2.2) αναπαριστά ένα χαρακτηριστικό ενός αντικειμένου που πρόκειται να ταξινομηθεί, ενώ κάθε κλαδί που ξεκινά από τον κόμβο αυτό αντιστοιχεί σε μια από τις πιθανές τιμές του χαρακτηριστικού (α1, β1, γ1 κ.λ.π. στο Σχήμα 2.2),

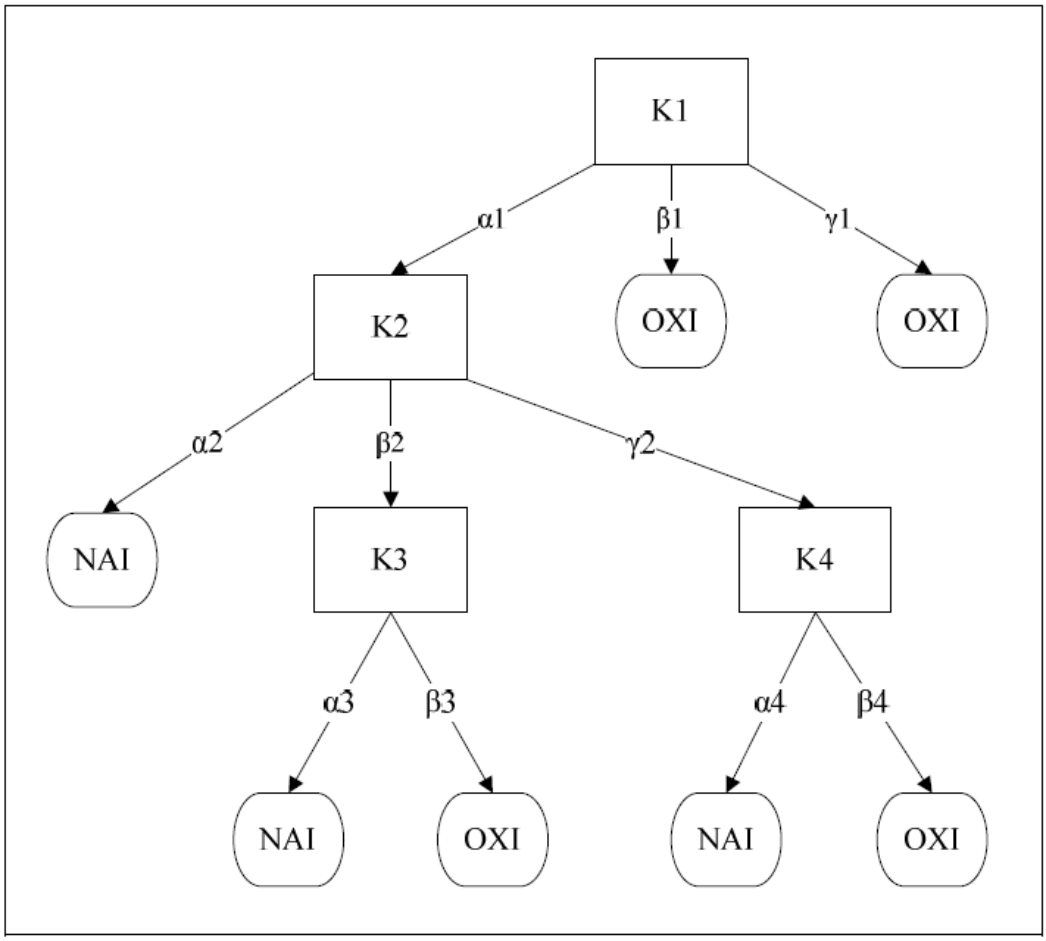
τις οποίες ο κόμβος μπορεί να λάβει. Επιπλέον, ένα φύλλο αντιστοιχεί σε μια από τις προκαθορισμένες κλάσεις (ΝΑΙ, ΟΧΙ στο Σχήμα 2.2) της διαδικασίας της ταξινόμησης (Parlante, 2000).

Η ταξινόμηση ενός νέου αντικειμένου μέσω ενός δέντρου απόφασης ακολουθεί τα εξής βήματα: Ξεκινώντας από την *ρίζα του δέντρου* (αρχικός κόμβος) και εξετάζοντας τα χαρακτηριστικά που καθορίζονται από τον κόμβο αυτό, προσδιορίζονται διαδοχικά οι εσωτερικοί κόμβοι του δέντρου που πρέπει να ακολουθηθούν, έως ότου καταλήξουμε σε ένα συγκεκριμένο φύλλο. Σε κάθε *εσωτερικό κόμβο*, εξετάζεται αν το προς ταξινόμηση αντικείμενο ικανοποιεί τον συγκεκριμένο κόμβο. Η έκβαση της εξέτασης αυτής καθορίζει το κλαδί που θα ακολουθηθεί στην συνέχεια, καθώς και τον επόμενο κόμβο. Η κλάση στην οποία θα ταξινομηθεί το νέο αντικείμενο αντιστοιχεί σε ένα από τα φύλλα του δέντρου απόφασης, είναι δε αυτή του τελικού κόμβου (Mitchell, 1997, Βαζιργιάννης & Χαλκίδη, 2003).

Οι αλγόριθμοι ταξινόμησης που βασίζονται στα δέντρα απόφασης, περιλαμβάνουν δύο διακριτές φάσεις: (1) τη *φάση οικοδόμησης* (***building phase***) και (2) τη *φάση κλαδέματος* (***pruning phase***). Στην πρώτη φάση, το σύνολο των δεδομένων εκπαίδευσης χωρίζεται πολλές φορές, έως ότου όλα τα αντικείμενα σε ένα τμήμα του ανωτέρω συνόλου να ανήκουν στην ίδια κλάση. Έπειτα, αφού έχει ήδη δημιουργηθεί το δέντρο απόφασης, οι περισσότεροι αλγόριθμοι εκτελούν τη φάση του κλαδέματος, περικόπτοντας κάποιους από τους κόμβους, προκειμένου αφενός να αποτραπούν επικαλύψεις, και αφετέρου το δέντρο να έχει υψηλότερη ακρίβεια ταξινόμησης.

Τα τελευταία χρόνια έχει αναπτυχθεί ένας μεγάλος αριθμός αλγορίθμων ταξινόμησης που βασίζονται σε δέντρα απόφασης. Δύο από τους πλέον σημαντικούς, είναι οι αλγόριθμοι ID3 (Quinlan, 1986) και C4.5 (Quinlan, 1993). Οι αλγόριθμοι αυτοί βασίζονται στην στατιστική ιδιότητα του *κέρδους πληροφορίας* (***information gain***), που με την σειρά της βασίζεται στην έννοια της *εντροπίας* (βλέπε Παράρτημα Α), για την επιλογή του χαρακτηριστικού που θα εξετάσουν σε κάθε κόμβο του δέντρου.

Εναλλακτικά, άλλοι αλγόριθμοι δέντρων απόφασης όπως οι SLIQ (Mehta et al., 1996) και SPRINT (Shafer et al., 1996), επιλέγουν το χαρακτηριστικό που θα εξετάσουν με βάση το δείκτη *Gini* (Gastwirth, 1972). Άλλοι ευρύτατα διαδεδομένοι αλγόριθμοι που βασίζονται στα δέντρα απόφασης, μεταξύ άλλων, είναι ο στατιστικός αλγόριθμος CART (Breinman et al., 1984), καθώς και ο αλγόριθμος Rainforest (Gehrke et al., 2000).



Σχήμα 2.1: Ένα απλό δέντρο απόφασης

2.1.2 Μέθοδοι κανόνων απόφασης

Μια πολύ σημαντική ιδιότητα των δέντρων απόφασης, είναι η ικανότητα μετατροπής τους σε ένα σύνολο κανόνων απόφασης (***decision rules***) (Quinlan, 1987, 1993). Συγκεκριμένα, δημιουργείται ένας ξεχωριστός κανόνας για κάθε μονοπάτι που ξεκινά από την κορυφή του δέντρου και καταλήγει σε ένα φύλλο που αναπαριστά μια κλάση.

Επιπλέον, τα περισσότερα από τα άλλα είδη τυποποίησης των εξαγομένων των αλγορίθμων της εξόρυξης δεδομένων, όπως οι *λίστες απόφασης* (***decision lists***), τα *προς τα κάτω αναπτυσσόμενα σύνολα κανόνων* (***ripple down rule sets***), τα επαγωγικά λογικά προγράμματα (***inductive logic programs***) ή τα *νευρωνικά δίκτυα* (***neural networks***), μπορούν επίσης να μετατραπούν σε κανόνες. Ειδικά για την μετατροπή των τελευταίων σε κανόνες απόφασης, η διεθνής βιβλιογραφία είναι ιδιαιτέρως πλούσια (Towell & Shavlik, 1994,Andrews et al., 1995,Boutsinas & Vrahatis, 2001, Zhou, 2004).

Ωστόσο, αξίζει να σημειωθεί ότι οι κανόνες απόφασης μπορούν επιπλέον να εξαχθούν και απευθείας από το σύνολο εκπαίδευσης μιας βάσης δεδομένων, μέσω μιας σειράς αλγορίθμων ταξινόμησης, οι οποίοι βασίζονται στους κανόνες απόφασης (***rule-based methods***) (Furnkranz, 1999). Στόχος των παραπάνω αλγορίθμων είναι η εξαγωγή του μικρότερου δυνατού συνόλου κανόνων απόφασης που είναι συνεπές με τα υπό εκπαίδευση δεδομένα. Οι εξαχθέντες κανόνες απόφασης έχουν την γενική μορφή «If *A* Then *B*», με το «If» κομμάτι να αποτελεί ένα συνδυασμό ζευγών από τιμές χαρακτηριστικών, αναπαριστώντας τις επαρκείς συνθήκες για την εφαρμογή - ανάθεση της τιμής της κλάσης που περιγράφεται στο «Then» κομμάτι του κανόνα,

στο υπό ταξινόμηση αντικείμενο της βάσης δεδομένων. Ένας αλγόριθμος που βασίζεται στους κανόνες απόφασης, πρέπει να παράγει κανόνες οι οποίοι έχουν υψηλές ικανότητες πρόβλεψης και ταυτόχρονα υψηλή αξιοπιστία.

Σημαντικό ρόλο σε αυτό διαδραματίζουν συνήθως μηχανισμοί, που είτε καθιστούν πολύ εξειδικευμένους κανόνες πιο γενικούς, σε μια ξεχωριστή φάση κλαδέματός τους (π.χ. Furnkranz, 1997), είτε σταματούν την διαδικασία εξειδίκευσης των κανόνων μέσω της χρήσης μέτρων ποιότητας. Αυτά τα μέτρα ποιότητας, χρησιμοποιούνται τόσο στην διαδικασία εξαγωγής των κανόνων όσο και στην διαδικασία ταξινόμησης του εκάστοτε αλγορίθμου. Αφενός, στην διαδικασία εξαγωγής των κανόνων, ένα μέτρο αξιολόγησης της ποιότητάς τους μπορεί να χρησιμοποιηθεί σαν κριτήριο της διαδικασίας εξειδίκευσης ή/και γενίκευσης των κανόνων, αφετέρου στην διαδικασία της ταξινόμησης, μια τιμή ενός μέτρου αξιολόγησης ποιότητας μπορεί να αντιστοιχιστεί σε κάθε κανόνα, για την επίλυση συγκρούσεων στην περίπτωση που πολλοί κανόνες ταυτόχρονα ικανοποιούν το προς ταξινόμηση αντικείμενο. Οι An & Cercone (2000) αναφέρονται αναλυτικά στα σημαντικότερα από τα μέτρα αξιολόγησης της ποιότητας των κανόνων. (Lavrac et al., 1999; Stefanowski & Vanderpooten, 2001; Flach & Lavrac, 2003; Tsumoto, 2003).

Στην διεθνή βιβλιογραφία υπάρχει ένας πολύ μεγάλος αριθμός αλγορίθμων ταξινόμησης που βασίζονται στους κανόνες απόφασης. Αναλυτική αναφορά σε αυτούς γίνεται στον Furnkranz (1999). Ένας από τους σημαντικότερους αλγορίθμους που βασίζεται στους κανόνες απόφασης είναι ο αλγόριθμος RIPPER (Cohen, 1995), ο οποίος διαμορφώνει κανόνες μέσα από μια συνεχή διαδικασία *ανάπτυξης* (***growing***) και *κλαδέματος* (***pruning***). Στην διάρκεια της πρώτης φάσης, οι δημιουργηθέντες κανόνες είναι πιο συνεπτυγμένοι, με στόχο την καλύτερη δυνατή προσαρμογή τους στα δεδομένα του συνόλου εκπαίδευσης, ενώ στην δεύτερη φάση συμβαίνει ακριβώς το αντίθετο, με στόχο την καλύτερη απόδοση του αλγορίθμου σε νέα δεδομένα.

Άλλοι σημαντικοί αλγόριθμοι είναι αυτοί της οικογένειας AQ (Michalski &

Chilausky, 1980), ο αλγόριθμος PART (Frank & Witten, 1998) καθώς και ο CN2 (Clark & Niblett, 1989).

Ειδικά ο αλγόριθμος CN2 είναι από τους πιο σημαντικούς αλγόριθμους που βασίζονται σε κανόνες. Βασισμένος στην «If *A* Then *B*» μορφή των κανόνων, χρησιμοποιεί μια ευρεστική συνάρτηση για τον τερματισμό της διαδικασίας κατασκευής τους, βάσει μιας εκτίμησης για τον θόρυβο που εμπεριέχεται στα δεδομένα. Το εξαγόμενο αποτέλεσμα του CN2 είναι ένα σύνολο διατεταγμένων «If *A* Then *B*» κανόνων, γνωστό και ως *λίστα απόφασης* (***decision list***) (Rivest, 1987). Αξίζει ακόμα να αναφέρουμε τον αλγόριθμο CL2 (Boutsinas et al., 2004), ο οποίος εξάγει κανόνες απόφασης χρησιμοποιώντας διαδικασίες ομαδοποίησης δεδομένων. Ο CN2 και ο CL2 θα χρησιμοποιηθούν στα τρεξίματα της μεθόδου CLEDM, γι'αυτό και θα γίνει μια πιο αναλυτική παρουσίαση τους στο Παράρτημα Α της διατριβής (Μαστρογιάννης, 2009).

2.1.3 Τεχνητά Νευρωνικά Δίκτυα

Πέρα από τις μεθόδους ταξινόμησης που βασίζονται στα δέντρα και τους κανόνες απόφασης, τα τεχνητά νευρωνικά δίκτυα (***artificial neural networks***) είναι επίσης μια διαδεδομένη μέθοδος ταξινόμησης (Michie et al, 1995, Kotsiantis, 2007).

Συγκεκριμένα, είναι μια δομή που αποτελείται από ένα δίκτυο *νευρώνων* (***neurons***) οι οποίοι συνδέονται μεταξύ τους. Η πιο διαδεδομένη κατηγορία νευρωνικών δικτύων είναι τα λεγόμενα δίκτυα πρόσθιας τροφοδότησης (***feed-forward neural networks***), τα οποία επιτρέπουν την κίνηση των δεδομένων μόνο προς μια κατεύθυνση, δηλαδή από μια είσοδο προς μια έξοδο. Δίκτυα που σχηματίζουν κυκλικές δομές ονομάζονται *ανατροφοδοτούμενα νευρωνικά δίκτυα* (***recurrent neural networks***) (Ρίζος, 1996).

Οι νευρώνες ενός δικτύου χωρίζονται σε τρείς βασικές κατηγορίες:
(1) τους *νευρώνες εισόδου* (***input neurons***), οι οποίοι δέχονται τις πληροφορίες που θα υποστούν επεξεργασία,
(2) τους *νευρώνες εξόδου* (***output neurons***), στους οποίους καταλήγουν τα αποτελέσματα της παραπάνω επεξεργασίας, και
(3) τους *ενδιάμεσους νευρώνες*, οι οποίοι βρίσκονται μεταξύ των νευρώνων εισόδου και εξόδου. Οι τελευταίοι εναλλακτικά ονομάζονται και *κρυφοί νευρώνες* (***hidden neurons***). Ουσιαστικά, οι νευρώνες σε ένα δίκτυο είναι αφενός ένα σύνολο εισερχόμενων τιμών και των αντίστοιχων βαρών τους, και αφετέρου μια συνάρτηση που αθροίζει τα παραπάνω βάρη, αντιστοιχώντας τα αποτελέσματα σε ένα νευρώνα εξόδου (Aggarwal & Yu, 1999).

Όπως φαίνεται και στο σχήμα 2.2, οι νευρώνες οργανώνονται σε *επίπεδα* (***layers***). Το *επίπεδο εισαγωγής* (***input layer***), περιλαμβάνει τις τιμές ενός αντικειμένο (και όχι πλήρεις νευρώνες), οι οποίες αποτελούν τις εισαγωγές στο επόμενο επίπεδο νευρώνων. Τα επόμενα επίπεδα καλούνται *κρυφά* (***hidden layers***). Το τελευταίο επίπεδο είναι η έξοδος, στην οποία υπάρχει ένας κόμβος για κάθε κλάση. Μια σάρωση του δικτύου με κίνηση των δεδομένων προς τα δεξιά, οδηγεί στην ανάθεση μιας τιμής σε κάθε κόμβο εξόδου, το δε αντικείμενο ανατίθεται στον κόμβο της κλάσης με την υψηλότερη τιμή (Batcher, 1968).

Σε ένα νευρωνικό δίκτυο πρόσθιας τροφοδότησης, τα κύρια βήματα για την κατασκευή ενός μοντέλου ταξινόμησης, είναι τα εξής (Aggarwal & Yu, 1999; Βαζιργιάννης & Χαλκίδη, 2003):

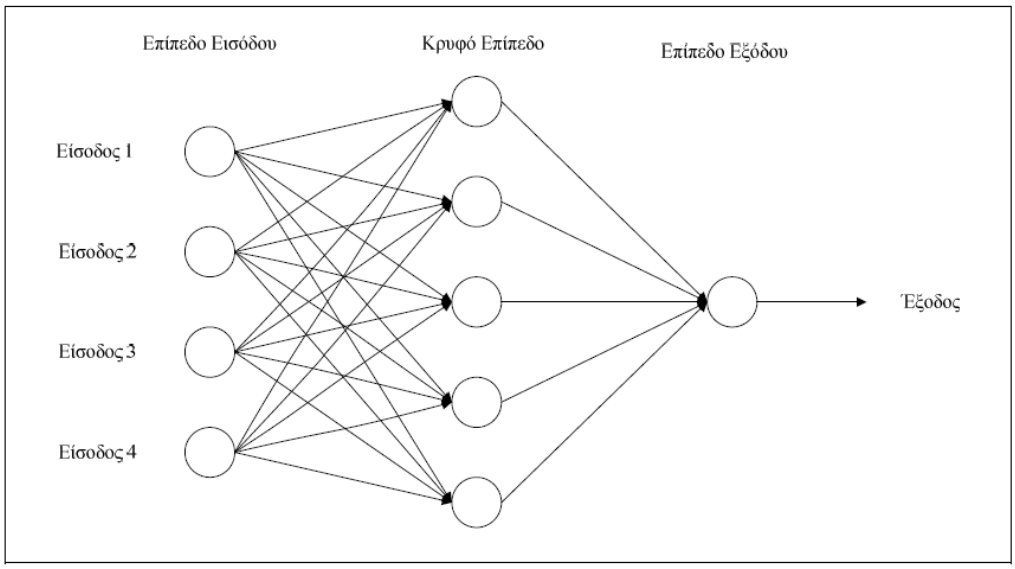
1. Η αναγνώριση των χαρακτηριστικών εισόδου και εξόδου.
2. Η κατασκευή ενός δικτύου με την κατάλληλη τοπολογία.
3. Η επιλογή του σωστού συνόλου εκπαίδευσης, το οποίο περιλαμβάνει δεδομένα που είναι ορισμένα ανά ζεύγη.
4. Η εκπαίδευση του δικτύου. Στην διάρκεια της φάσης αυτής, τα δεδομένα εισέρχονται στο νευρωνικό δίκτυο ένα ένα. Το νευρωνικό δίκτυο μαθαίνει συγκρίνοντας τα

αποτελέσματα ταξινόμησης ενός αντικειμένου με την γνωστή πραγματική ταξινόμηση αυτού. Τα λάθη από την αρχική ταξινόμηση του πρώτου αντικειμένου χρησιμοποιούνται για να διορθωθεί το δίκτυο μέσω της τροποποίησης των συναρτήσεων των νευρώνων. Η παραπάνω διαδικασία είναι επαναληπτική. Η επαναληπτική φύση ωστόσο της διαδικασίας εκπαίδευσης, σημαίνει ότι ένα νευρωνικό δίκτυο είναι αρκετά αργό.

5. Ο έλεγχος του δικτύου χρησιμοποιώντας ένα σύνολο ελέγχου, το οποίο είναι ανεξάρτητο από το σύνολο εκπαίδευσης.

Στην συνέχεια, το μοντέλο που παράγεται από το δίκτυο εφαρμόζεται για να ταξινομήσει νέα δεδομένα. Ο Zhang (2000) περιγράφει αναλυτικά μια σειρά από μεθόδους ταξινόμησης που βασίζονται στα νευρωνικά δίκτυα, ακολουθώντας την γενικότερη παραπάνω λογική. Αξίζει να σημειώσουμε ότι εν πολλοίς η εκπαίδευση ενός νευρωνικού δικτύου βασίζεται στον υπολογισμό των τιμών των βαρών που προ-αναφέρθηκαν.

Ο πιο γνωστός αλγόριθμος, μεταξύ άλλων (Neocleous & Schizas, 2002), στον οποίο βασίζεται ο παραπάνω υπολογισμός, είναι ο *αλγόριθμος ανάστροφης μετάδοσης* (***back propagation algorithm***) (Rumelhart et al., 1986). Άλλες προσεγγίσεις που χρησιμοποιούνται για την εκπαίδευση των νευρωνικών δικτύων, με κύριο στόχο την βελτίωση των χρονικών τους επιδόσεων, είναι αυτές των Weigend et al (1990) και Yam & Chow (2001). Επιπλέον, για την εκπαίδευση των νευρωνικών δικτύων μπορούν να χρησιμοποιηθούν τόσο γενετικοί αλγόριθμοι (Siddique & Tokhi, 2001), όσο και στατιστικές μέθοδοι Bayes (Vivarelli & Williams, 2001).



Σχήμα 2.2: Ένα απλό τεχνητό νευρωνικό δίκτυο

2.1.4 Στατιστικές μέθοδοι ταξινόμησης

Οι *στατιστικές μέθοδοι* (***statistical methods***) ταξινόμησης χαρακτηρίζονται από το γεγονός ότι χρησιμοποιούν μοντέλα πιθανότητας, τα οποία αντί για μια απλή ταξινόμηση ενός αντικειμένου που ανήκει σε μια βάση δεδομένων, δίνουν την πιθανότητα το αντικείμενο αυτό να ανήκει σε κάθε μια από τις κλάσεις της διαδικασίας της ταξινόμησης. Τα πιο συνηθισμένα στατιστικά μοντέλα ταξινόμησης ορίζονται βάσει της *θεωρίας του* ***Bayes*** (Cheeseman & Stutz; 1996). Πρόκειται για τον *αφελή ταξινομητή* ***Bayes*** αφενός, και τα *δίκτυα* ***Bayes*** αφετέρου.

2.1.4.1 Αφελής ταξινομητής Bayes

Ο *αφελής ταξινομητής* ***Bayes*** (***Naïve Bayes classifier***) χρησιμοποιήθηκε για πρώτη φορά στο πεδίο της μηχανικής μάθησης από τους Cestnik et al (1987). Υποθέτει ότι η παρουσία (ή απουσία) ενός συγκεκριμένου χαρακτηριστικού μιας κλάσης είναι

ανεξάρτητη από την παρουσία (ή απουσία) κάθε άλλου χαρακτηριστικού. Η υπόθεση αυτή ονομάζεται *υπό συνθήκη ανεξαρτησία* (*conditional independence*).

Ας θεωρήσουμε ότι *S* είναι μια βάση προς ταξινόμηση δεδομένων που περιγράφονται

από τα χαρακτηριστικά F_j , j =1,..,n και ότι C_i , i = 1,..,m είναι οι αντίστοιχες κλάσεις. Αν $X = (x_1, x_2, ...,x_n)$ είναι ένα αντικείμενο της βάσης δεδομένων *S*, τότε ο αφελής ταξινομητής Bayes θα αναθέσει το αντικείμενο αυτό σε εκείνη την κλάση που έχει την υψηλότερη *εκ των υστέρων πιθανότητα* (*posterior probability*) $p(C_i/X)$. Επομένως, το *X* θα ανατεθεί στην κλάση C_i αν και μόνο αν:

$$p(C_i/X) \ \mathbf{f} \ p(C_l/X) \quad 1 \le l \le m \text{ για κάθε } l \ne i \quad (2.1)$$

Όπου

$$p(C_i/X) = \left[\frac{p(X/C_i) \cdot p(C_i)}{p(X)} \right] \qquad (2.2)$$

Οι πιθανότητες *p*(*Ci*) , *p*(*X*) ονομάζονται *εκ των προτέρων πιθανότητες* (*prior probabilities*) και χαρακτηρίζουν κάθε κλάση C_i , και το αντικείμενο *X* αντίστοιχα, ενώ $p(X/C_i)$ είναι η πιθανότητα το αντικείμενο *X* να ανήκει στην κλάση C_i .

Ακόμα, η κλάση C_i για την οποία η πιθανότητα $p(C_i/X)$ μεγιστοποιείται, ονομάζεται μέγιστη μεταγενέστερη υπόθεση (Βαζιργιάννης & Χαλκίδη, 2003). Ο αφελής ταξινομητής Bayes είναι μια πολύ αποδοτική τεχνική, συγκρίσιμη ή και υπό προϋποθέσεις ανώτερη από άλλες τεχνικές όπως τα δέντρα απόφασης και οι ταξινομητές που βασίζονται στους κανόνες ή/και στα νευρωνικά δίκτυα (Domingos & Pazzani, 1997, Zhang, 2004). Αξίζει να σημειωθεί ότι οι Friedman et al (1997), δίνουν μια σημαντική παραλλαγή του βασικού αφελούς ταξινομητή Bayes, με στόχο τη βελτίωση της απόδοσης του, ορίζοντας ένα νέο πλαίσιο που υπερνικά την υπόθεση ανεξαρτησίας του τελευταίου.

2.1.5 Δίκτυα Bayes

Ένα *δίκτυο Bayes* (*Bayes network*) (Jensen, 1996) είναι ένα γραφικό μοντέλο που βασίζεται σε πιθανότητες, λαμβάνοντας υπόψη το σύνολο των μεταβλητών του μοντέλου και τις μεταξύ τους εξαρτήσεις. Πιο συγκεκριμένα, ένα δίκτυο Bayes είναι ένας *κατευθυνόμενος μη κυκλικός γράφος* (*directed acyclic graph*), κάθε κόμβος του οποίου αντιπροσωπεύει ένα αντικείμενο X , ενώ κάθε τόξο αντιπροσωπεύει τις μεταξύ των αντικειμένων εξαρτήσεις, υπό την μορφή πιθανοτήτων (*probabilistic dependencies*). Αν ένα τόξο ξεκινά από το αντικείμενο X_1 και καταλήγει στο αντικείμενο X_2, τότε το X_1 είναι ο γονέας του X_2 , και το X_2 είναι ο απόγονος του X_1. Κάθε μεταβλητή είναι ανεξάρτητη των μη προγόνων της, δεδομένων των γονιών της, δηλαδή η X_1 είναι ανεξάρτητη από την X_3 δεδομένης της X_2, αν

$P(X_1 / X_3 , X_2) = P(X_1/X_2)$ για κάθε X_1, X_2 , X_3

Η εκπαίδευση ενός δικτύου Bayes περιλαμβάνει δύο επιμέρους διαδικασίες (Kotsiantis, 2007). Πρώτον την εκπαίδευση του γράφου που αναπαριστά το δίκτυο και δεύτερον τον υπολογισμό των παραμέτρων του δικτύου. Η πρώτη διαδικασία μπορεί με την σειρά της να διακριθεί σε δύο περιπτώσεις. Στην πρώτη περίπτωση θεωρούμε ότι η δομή του δικτύου είναι γνωστή (π.χ. από έναν ειδικό). Στην δεύτερη περίπτωση, η δομή του δικτύου είναι άγνωστη, καθορίζεται δε από μια συνάρτηση η οποία αξιολογεί την προσαρμογή των πιθανών δικτύων στα δεδομένα εκπαίδευσης, επιλέγοντας εν τέλει το καλύτερο από αυτά.

Η δεύτερη διαδικασία, δηλαδή ο υπολογισμός των παραμέτρων του δικτύου, λαμβάνει χώρα μέσα από τον λεγόμενο *πίνακα υπό συνθήκη πιθανότητας* (*conditional probability table*), που ορίζεται για κάθε ένα από τα αντικείμενα-κόμβους του δικτύου. Ο πίνακας που ορίζεται για το αντικείμενο X , χρησιμοποιείται για τον προσδιορισμό της δεσμευμένης κατανομής $P(X \mid \text{γονέας}(X))$. Λαμβάνοντας υπόψη την παραπάνω κατανομή, η *συνδυασμένη κατανομή* (*joint distribution*) των τιμών των κόμβων του δικτύου Bayes που περιγράφονται από τα χαρακτηριστικά $F_j , j = 1,...,n$

δίνεται από την σχέση:

$$P(X_1, X_2, ..., X_n) = \prod_{i=1}^{n} P(X_i / gonéaV(X_i)) \quad (2.3)$$

Βάσει των παραπάνω δύο διαδικασιών, από ένα δεδομένο σύνολο εκπαίδευσης προκύπτει ένα δίκτυο Bayes. Ένας ταξινομητής που βασίζεται στο δίκτυο αυτό, και κατ'επέκταση στο σύνολο των δοθέντων αντικειμένων $X_1, X_2, ..., X_n$, επιστρέφει την κλάση C που μεγιστοποιεί την εκ των *υστέρων πιθανότητα* (***posterior probability***) $P(C \mid X_1, X_2, ..., X_n)$

Σημαντικές τροποποιήσεις στα δίκτυα Bayes που βελτιώνουν την ακρίβεια της ταξινόμησης, την θεωρητική τους θεμελίωση, αλλά και την γενικότερη απόδοσή τους έχουν προταθεί από μια σειρά ερευνητές όπως (π.χ. οι Heckerman et al (1999), Cheng & Greiner (2001), Chickering (2002) και Acid & De Campos (2003)). Αξίζει τέλος να σημειώσουμε ότι τα δίκτυα Bayes δεν είναι ιδιαίτερα καλοί ταξινομητές στην περίπτωση που η βάση των προς ταξινόμηση δεδομένων είναι αρκετά μεγάλη, κυρίως λόγω του μεγάλου χώρου και χρόνου που αυτό απαιτεί (Cheng et al., 2002).

2.1.6 Μέθοδοι μάθησης κατά περίπτωση

Μια άλλη διαδεδομένη μέθοδος ταξινόμησης είναι η *μέθοδος της μάθησης κατά περίπτωση* (***instance-based learning***). Οι αλγόριθμοι της μεθόδου αυτής (Aha, 1997, De Mantaras & Armengol, 1998), είναι *αλγόριθμοι αναβλητικής μάθησης* (***lazy learning algorithms***) (Mitchell, 1997). Αυτό σημαίνει ότι στους αλγορίθμους αυτούς, η γενίκευση πέρα από τα δεδομένα της εκπαίδευσης καθυστερεί μέχρις ότου γίνει μια πρώτη ταξινόμηση, σε αντίθεση με τους *αλγορίθμους έγκαιρης μάθησης* (***eager learning algorithms***), όπως οι αλγόριθμοι δέντρων απόφασης ή νευρωνικών δικτύων, στους οποίους το μοντέλο προσπαθεί πρώτα να γενικεύσει τα δεδομένα εκπαίδευσης και μετά να ταξινομήσει νέα δεδομένα. Κατ'επέκταση, οι αλγόριθμοι αναβλητικής

μάθησης έχουν μικρότερη υπολογιστική πολυπλοκότητα στην φάση της εκπαίδευσης σε σχέση με τους αλγορίθμους έγκαιρης μάθησης, αλλά μεγαλύτερη πολυπλοκότητα στην φάση της ταξινόμησης.

Ένας από τους πλέον διαδεδομένους αλγόριθμους μάθησης κατά περίπτωση, είναι ο αλγόριθμος του *k- Κοντινότερου Γείτονα* (*k-Nearest Neighbor* ή *kNN*) (Kotsiantis, 2007). Ο αλγόριθμος αυτός βασίζεται στην αρχή που υποστηρίζει ότι τα αντικείμενα μιας βάσης δεδομένων βρίσκονται σε εγγύτητα με άλλα αντικείμενα που έχουν περεμφερείς ιδιότητες (Cover & Hart, 1967). Αν κάθε ένα από τα αντικείμενα αυτά είναι προσκολημένα σε μια κλάση, τότε ο καθορισμός της κλάσης στην οποία θα ανατεθεί ένα μη ταξινομημένο αντικείμενο, γίνεται μέσα από την παρατήρηση των κλάσεων στις οποίες είναι αντιστοιχισμένα τα κοντινότερα σε αυτό αντικείμενα. Ο αλγόριθμος *kNN* βρίσκει τα *k* κοντινότερα αντικείμενα, του υπό ταξινόμηση αντικειμένου, και το ταξινομεί στην πιο συνιθησμένη κλάση των *k* αυτών αντικειμένων.

Παρά την αδιαμφισβήτητη χρησιμότητά του, που οδήγησε σε μοντέλα όπως ο PEBLS (Cost & Salzberg, 1993), ο *kNN* έχει και αρκετά μειονεκτήματα. Συγκεκριμένα, έχει μεγάλες απαιτήσεις αποθήκευσης, είναι ευαίσθητος στον θόρυβο που ενδεχομένως εμπεριέχεται στην συνάρτηση ομοιότητας που χρησιμοποιείται για την σύγκριση των αντικειμένων, ενώ ταυτόχρονα δεν υπάρχει ένας ολοκληρωμένος και αποτελεσματικός τρόπος υπολογισμού του *k*. Προκειμένου να αντιμετωπιστούν τα προβλήματα αυτά, εμφανίστηκαν σχετικά πρόσφατα μια σειρά από εργασίες, (όπως π.χ. αυτές των Wettschereck et al (1997), Kubat & Cooperson (2001), Sanchez et al (2002) και Okamoto & Yugami (2003)).

2.1.7 Μηχανές Διανυσμάτων Υποστήριξης

Οι *μηχανές διανυσμάτων υποστήριξης* (*support vector machines*) αποτελούν την πλέον πρόσφατη μέθοδο μηχανικής μάθησης (Vapnik, 1995, 1998, Burges, 1998). Ας

θεωρήσουμε ότι έχουμε στην διάθεση μας n αντικείμενα εκπαίδευσης, τα οποία αποτελούνται από ένα διάνυσμα $x_i \in R^n$ και μια τιμή κλάσης y_j. Μια μηχανή διανυσμάτων υποστήριξης παράγει ένα ταξινομητή, το επονομαζόμενο *βέλτιστο υπερ-επίπεδο διαχωρισμού*, μέσα από την μη γραμμική απεικόνιση των εισερχόμενων ανωτέρω διανυσμάτων στον πολύ-διάστατο χώρο των χαρακτηριστικών που περιγράφουν τα διανύσματα αυτά (Shin et al., 2005). Με άλλα λόγια, μια μηχανή διανυσμάτων υποστήριξης αντιστοιχεί τα δεδομένα μιας βάσης σε ένα πολύ-διάστατο χώρο, καθορίζοντας στον χώρο αυτό ένα βέλτιστο υπερ-επίπεδο διαχωρισμού τους.

Η βασική έννοια γύρω από την οποία δομείται μια μηχανή διανυσμάτων υποστήριξης είναι αυτή του *περιθωρίου* (***margin***), σε κάθε μια από τις πλευρές ενός καθορισμένου υπερ-επιπέδου που χωρίζει τα δεδομένα ενός συνόλου εκπαίδευσης. Συγκεκριμένα, μια μηχανή διανυσμάτων υποστήριξης κατασκευάζει ένα γραμμικό μοντέλο για την εκτίμηση του συνόλου των παραμέτρων a της συνάρτησης απόφασης $f(x, a)$, έτσι ώστε η τελευταία να πραγματοποιήσει την αντιστοίχιση $x_i \rightarrow y_j$ (η $f(x, a)$ ονομάζεται *μηχανή εκπαίδευσης*) (Βαζιργιάννης & Χαλκίδη, 2003).

Αν τα δεδομένα εκπαίδευσης είναι γραμμικά διαχωρισμένα, τότε η μηχανή διανυσμάτων υποστήριξης εκπαιδεύει μηχανές για την εκτίμηση ενός βέλτιστου υπερ-επιπέδου που διαχωρίζει τα δεδομένα, στην μεγαλύτερη δυνατή απόσταση ανάμεσα σε αυτό και τα κοντινότερα αντικείμενα εκπαίδευσης. Τα αντικείμενα εκπαίδευσης που είναι πιο κοντά στο βέλτιστο υπερ-επίπεδο διαχωρισμού ονομάζονται *διανύσματα υποστήριξης* (***support vectors***), η δε λύση αναπαρίσταται σαν ένας γραμμικός συνδυασμός των παραπάνω αντικειμένων. Σε πιο γενικές περιπτώσεις που τα δεδομένα δεν είναι γραμμικά διαχωρισμένα, η μηχανή διανυσμάτων υποστήριξης χρησιμοποιεί μη γραμμικές μηχανές για την εύρεση ενός υπερ-επιπέδου που ελαχιστοποιεί το πλήθος των λαθών για το σύνολο εκπαίδευσης (Cristianini & Shawe-Taylor, 2000, Shin et al., 2005).

Η μεγιστοποίηση του περιθωρίου και κατ'επέκταση η δημιουργία της μεγαλύτερης δυνατής απόστασης ανάμεσα στο υπερ-επίπεδο και τα αντικείμενα των δεδομένων

εκπαίδευσης σε κάθε πλευρά του πρώτου, έχει αποδειχτεί ότι ελαχιστοποιεί το άνω όριο του σφάλματος γενίκευσης. Αυτή η ελαχιστοποίηση επιτυγχάνεται με την εκπαίδευση του a, ώστε η $f(x, a)$ να ικανοποιεί την ιδιότητα του *μέγιστου περιθωρίου*, δηλαδή το όριο απόφασης που αντιπροσωπεύει να έχει την μέγιστη απόσταση από το κοντινότερο αντικείμενο εκπαίδευσης.

2.1.8 Λοιπές μέθοδοι και αλγόριθμοι

Πέρα από αυτές τις γενικές κατηγορίες μεθόδων ταξινόμησης και τους αντίστοιχους αλγορίθμους τους, υπάρχει και μια σειρά άλλων μεθόδων και αλγορίθμων ταξινόμησης, ίσως περισσότερο εξειδικευμένων. Ειδικότερα, μπορούμε να αναφέρουμε αλγόριθμους που βασίζονται στον *επαγωγικό λογικό προγραμματισμό* (***inductive logic programming***).

Οι αλγόριθμοι αυτοί, θεωρώντας δεδομένη την ύπαρξη κωδικοποιημένης γνώσης καθώς και ένα σύνολο από παραδείγματα που αναπαρίστανται σαν μια λογική βάση δεδομένων, δημιουργούν ένα λογικό πρόγραμμα, το οποίο συνεπάγεται σαν εξαγόμενο όλα τα θετικά από τα παραπάνω παραδείγματα, και κανένα από τα αρνητικά. Παραδείγματα αλγορίθμων αυτού του είδους είναι αυτοί των Muggleton (1992) και Dzeroski (1996).

Ακόμα, μπορούν να αναφερθούν αλγόριθμοι που βασίζονται σε *υβριδικά συστήματα* (***hybrid systems***) όπως αυτός των Boutsinas & Vrahatis (2001), *αλγόριθμοι σύμμορφων προβλέψεων* (***conformal predictors***) (Vork et al., 2005), οι οποίοι είναι σε θέση να καθορίσουν διαστήματα προβλέψεων εκμεταλλευόμενοι την συμφωνία νέων δεδομένων με δεδομένα που έχουν παρατηρηθεί παλαιότερα, καθώς και *γενετικοί αλγόριθμοι* (***genetic algorithms***) (Freitas, 2003,Bandyopadhyay & Pal, 2007).

Κεφάλαιο 3

3.1 Βασικοί αλγόριθμοι ταξινόμησης

3.1.1 Βασικοί αλγόριθμοι δέντρων απόφασης

3.1.1.1 Αλγόριθμος ID3

Περίληψη

Ο ID3 χτίζει ένα δέντρο απόφασης από ένα σταθερό σύνολο παραδειγμάτων. Το δέντρο που προκύπτει χρησιμοποιείται για να ταξινομήσει τα μελλοντικά δείγματα. Το παράδειγμα έχει διάφορες ιδιότητες και ανήκει σε μια κλάση (όπως ναι ή όχι). Οι κόμβοι φύλλων του δέντρου απόφασης περιέχουν το όνομα κλάσης ενώ ένας κόμβος μη-φύλλων είναι ένας κόμβος απόφασης. Ο κόμβος απόφασης είναι μια δοκιμή ιδιοτήτων με κάθε κλάδο (σε ένα άλλο δέντρο απόφασης) που είναι μια πιθανή αξία των ιδιοτήτων.

Ο ID3 χρησιμοποιεί το κέρδος πληροφοριών για να το βοηθήσει να αποφασίσει ποια ιδιότητα πηγαίνει σε έναν κόμβο απόφασης. Το πλεονέκτημα για ένα δέντρο απόφασης είναι ότι ένα πρόγραμμα, παρά έναν μηχανισμό γνώσης, αποσπά τη γνώση.

Εισαγωγή

Ο Ross Quinlan ανέπτυξε αρχικά τον ID3 στο πανεπιστήμιο του Σύδνεϋ. Παρουσίασε αρχικά τον ID3 το 1975 σε ένα βιβλίο, μηχανή μάθησης, εντάσεις 1,

αριθ. 1. Ο ID3 είναι βασισμένος στον αλγόριθμο συστημάτων εκμάθησης έννοιας (CLS). Ο βασικός αλγόριθμος CLS πέρα από ένα σύνολο περιπτώσεων Γ:

Βήμα 1: Εάν όλες οι περιπτώσεις στο Γ είναι θετικές, κατόπιν δημιουργούμε ΝΑΙ τον κόμβο και τη στάση.
Εάν όλες οι περιπτώσεις στο Γ είναι αρνητικές, δημιουργούμε έναν κόμβο Όχι και σταματάμε.
Διαφορετικά επιλέγουμε ένα χαρακτηριστικό γνώρισμα, Φ με τις τιμές v1,…, vn και δημιουργούμε έναν κόμβο απόφασης.
Βήμα 2: Χωρίζουμε τις περιπτώσεις κατάρτισης στο Γ στα υποσύνολα C1, C2,…, Cn σύμφωνα με τις τιμές του V.
Βήμα 3: εφαρμόζουμε τον αλγόριθμο κατ' επανάληψη σε κάθε ένα από τα σύνολα Ci.
Σημείωση, ο εκπαιδευτής (ο εμπειρογνώμονας) αποφασίζει ποιο χαρακτηριστικό γνώρισμα να επιλέξει.

Ο ID3 βελτιώνεται σε CLS με την προσθήκη μιας επιλογής χαρακτηριστικών γνωρισμάτων εύρεσης. Ο ID3 ξαναζητεί μέσω των ιδιοτήτων των περιπτώσεων και των αποσπασμάτων κατάρτισης την ιδιότητα που χωρίζει καλύτερα τα δεδομένα στα παραδείγματα.

Εάν η ιδιότητα ταξινομεί τέλεια τις στάσεις συνόλων κατάρτισης ο ID3 σταματά, διαφορετικά λειτουργεί κατ' επανάληψη στο ν (όπου ν = αριθμός πιθανών τιμών των χωρισμένων υποσυνόλων μιας ιδιότητας) για να πάρει τις «καλύτερες» ιδιότητές τους.

Ο αλγόριθμος χρησιμοποιεί μια πλεονεκτική αναζήτηση, δηλ., αυτό επιλέγει τις καλύτερες ιδιότητες και δεν κοιτάζει ποτέ πίσω επανεξετάζει τις προηγούμενες επιλογές.

Ο ID3 είναι ένας μη αυξητικός αλγόριθμος, σημαίνοντας ότι αντλεί τις κλάσεις του από ένα σταθερό σύνολο περιπτώσεων κατάρτισης. Ένας επαυξητικός αλγόριθμος αναθεωρεί τον παρόντα καθορισμό έννοιας εάν είναι απαραίτητο, με ένα νέο δείγμα.

Οι κλάσεις που δημιουργούνται από τον ID3 είναι επαγωγικές, δηλ., λαμβάνοντας υπόψη ένα μικρό σύνολο περιπτώσεων κατάρτισης, οι συγκεκριμένες κλάσεις που δημιουργούνται από τον ID3 αναμένονται για να λειτουργήσουν για όλες τις μελλοντικές περιπτώσεις. Η διανομή των άγνωστων πρέπει να είναι η ίδια με τις περιπτώσεις δοκιμής.

Οι κλάσεις επαγωγής δεν μπορούν να αποδείξουν ότι λειτουργούν σε κάθε περίπτωση δεδομένου ότι μπορούν να ταξινομήσουν έναν άπειρο αριθμό περιπτώσεων. Σημειώστε ότι ο ID3 (ή οποιοσδήποτε επαγωγικός αλγόριθμος) μπορεί να καταχωρήσει λάθος τα στοιχεία.

Περιγραφή στοιχείων

Το στοιχείο δειγμάτων που χρησιμοποιείται από τον ID3 έχει ορισμένες απαιτήσεις, οι οποίες είναι:

- Περιγραφή ιδιότητα-αξίας - οι ίδιες ιδιότητες πρέπει να περιγράψουν κάθε παράδειγμα και να έχουν έναν σταθερό αριθμό τιμών.
- Προκαθορισμένες κλάσεις - οι ιδιότητες ενός παραδείγματος πρέπει ήδη να καθοριστούν, δηλ., αυτοί δεν μαθαίνονται από τον ID3.
- Ιδιαίτερες κλάσεις - οι κλάσεις πρέπει να σκιαγραφηθούν αισθητά. Οι συνεχείς κλάσεις που χωρίζονται στις ασαφείς κατηγορίες όπως ένα μέταλλο που είναι «σκληρά, αρκετά σκληρός, εύκαμπτος, μαλακός, αρκετά μαλακός» είναι ύποπτες.
- Ικανοποιητικά παραδείγματα - δεδομένου ότι η επαγωγική γενίκευση χρησιμοποιείται (δηλ. μη αποδείξιμος) εκεί πρέπει να είναι αρκετές περιπτώσεις δοκιμής για να διακρίνει τα έγκυρα πρότυπα από τα περιστατικά πιθανότητας.

Επιλογή ιδιοτήτων

Για να αποφασίσει οID3 ποια ιδιότητα είναι η καλύτερη χρησιμοποιεί μια στατιστική διαδικασία, αποκαλούμενη κέρδος πληροφοριών. Το κέρδος μετρά πόσο

καλά μια δεδομένη ιδιότητα χωρίζει τα παραδείγματα κατάρτισης στις στοχοθετημένες κλάσεις.

Ένας με τις υψηλότερες πληροφορίες (πληροφορίες που είναι ο πιό χρήσιμος για την ταξινόμηση) επιλέγεται. Προκειμένου να καθοριστεί το κέρδος, δανειζόμαστε αρχικά μια ιδέα από την αποκαλούμενη θεωρία εντροπία πληροφοριών. Η εντροπία μετρά το ποσό πληροφοριών σε μια ιδιότητα.

Λαμβάνοντας υπόψη μια συλλογή S των εκβάσεων γ

$$Entropía = S - p(I) \log 2\, p(I)$$

όπου το π (I) είναι το ποσοστό του S που ανήκει στην κλάση I. S είναι άνω του c. Log2 είναι log με βάση το 2.
Σημειώνουμε ότι το S δεν είναι μια ιδιότητα αλλά ολόκληρο το σύνολο δειγμάτων.

Παράδειγμα 1

Εάν το S είναι μια συλλογή 14 παραδειγμάτων με 9 ΝΑΙ και 5 ΚΑΝΕΝΑ παράδειγμα έπειτα

$$Entropía = -\left(\frac{9}{14}\right)\text{Log2}\left(\frac{9}{14}\right) - \left(\frac{5}{14}\right)\text{Log2}\left(\frac{5}{14}\right) = 0.940$$

Η εντροπία ειδοποίησης είναι 0 εάν όλα τα μέλη του S ανήκουν στην ίδια κλάση (το στοιχείο είναι τέλεια ταξινομημένο). Η σειρά της εντροπίας είναι 0 («τέλεια ταξινομημένος») σε 1 («συνολικά τυχαίος»).
Το κέρδος (S, A) είναι κέρδος πληροφοριών του παραδείγματος που το καθορισμένο S στις ιδιότητες A ορίζεται ως

$$KέρδoV\ (S, A) = entropía - S\left(\left(\frac{|Sv|}{|S|}\right) * entropía\ (Sv)\right)$$

Το S είναι κάθε αξία β όλων των πιθανών τιμών των ιδιοτήτων A
Sv = υποσύνολο του S για το οποίο η ιδιότητα A έχει την αξία v
|Sv| = αριθμός στοιχείων στο Sv
|S| = αριθμός στοιχείων στο S

Παράδειγμα 2

Υποθέστε ότι το S είναι ένα σύνολο 14 παραδειγμάτων στα οποία μια από τις ιδιότητες είναι ταχύτητα ανέμου. Οι τιμές του αέρα μπορούν να είναι αδύνατες ή ισχυρές. Η ταξινόμηση αυτών των 14 παραδειγμάτων είναι 9 ΝΑΙ και 5 αριθ. Για τον αέρα ιδιοτήτων, υποθέστε ότι υπάρχουν 8 περιστατικά του αέρα = αδύνατος και 6 περιστατικά του αέρα = ισχυρός. Για τον αέρα = αδύνατος, 6 των παραδειγμάτων είναι ΝΑΙ και 2 είναι αριθ. Για τον αέρα = ισχυρός, 3 είναι ΝΑΙ και 3 είναι αριθ. Επομένως

Κέρδος (S, αέρας) = Εντροπία - $\left(\frac{8}{14}\right)$* Εντροπία (S αδύνατος) - $\left(\frac{6}{14}\right)$* Εντροπία (S ισχυρός)

= 0.940 - $\left(\frac{8}{14}\right)$*0.811 - $\left(\frac{6}{14}\right)$*1.00

= 0.048

Εντροπία (S αδύνατος) = - $\left(\frac{6}{8}\right)$*log2 $\left(\frac{6}{8}\right)$- $\left(\frac{2}{8}\right)$*log2 $\left(\frac{2}{8}\right)$= 0.811

Εντροπία (S ισχυρός) = - $\left(\frac{3}{6}\right)$*log2 $\left(\frac{3}{6}\right)$ - $\left(\frac{3}{6}\right)$*log2 $\left(\frac{3}{6}\right)$= 1.00

Για κάθε ιδιότητα, το κέρδος υπολογίζεται και το υψηλότερο κέρδος χρησιμοποιείται στον κόμβο απόφασης.

Παράδειγμα 3

Υποθέστε ότι θέλουμε τον ID3 για να αποφασίσουμε εάν ο καιρός είναι υποκείμενος στο παιχνίδι του μπέιζ-μπώλ. Κατά τη διάρκεια 2 εβδομάδων, το στοιχείο συλλέγεται για να βοηθήσει τον ID3 την κατασκευή ένα δέντρο απόφασης (δείτε τον πίνακα 1).

Η ταξινόμηση στόχων είναι «εάν παίξουμε το μπέιζ-μπώλ;» όποιος μπορεί να είναι ναι ή όχι.

Οι καιρικές ιδιότητες είναι Πρόβλεψη, θερμοκρασία, Επίπεδο Υγρασίας, και Αέρας. Μπορούν να έχουν τις ακόλουθες τιμές:

Πρόβλεψη = { Ηλιοφάνεια, Νεφώσεις, Βροχόπτωση }

θερμοκρασία = { Ζέστη, Ήπια, Κρύο }

υγρασία = { Υψηλό, Κανονικό }

αέρας = { Αδύναμος, Δυνατός }

Τα παραδείγματα του συνόλου S είναι:

Πίνακας 3.1. Απόφαση διεξαγωγής αγώνα με βάση τις καιρικές συνθήκες

Ημέρα	Πρόβλεψη	Θερμοκρασία	Επίπεδο Υγρασίας	Αέρας	Διεξαγωγή Αγώνα
Η1	Ηλιοφάνεια	Ζέστη	Υψηλό	Αδύναμος	Όχι
Η2	Ηλιοφάνεια	Ζέστη	Υψηλό	Δυνατός	Όχι
Η3	Νεφώσεις	Ζέστη	Υψηλό	Αδύναμος	Ναι
Η4	Νεφώσεις	Ήπια	Υψηλό	Αδύναμος	Ναι
Η5	Βροχόπτωση	Κρύο	Κανονικό	Αδύναμος	Ναι
Η6	Βροχόπτωση	Κρύο	Κανονικό	Δυνατός	Όχι
Η7	Νεφώσεις	Κρύο	Κανονικό	Δυνατός	Ναι
Η8	Ηλιοφάνεια	Ήπια	Υψηλό	Αδύναμος	Όχι
Η9	Ηλιοφάνεια	Κρύο	Κανονικό	Αδύναμος	Ναι
Η10	Βροχόπτωση	Ήπια	Κανονικό	Αδύναμος	Ναι
Η11	Ηλιοφάνεια	Ήπια	Κανονικό	Δυνατός	Ναι
Η12	Νεφώσεις	Ήπια	Υψηλό	Δυνατός	Ναι
Η13	Νεφώσεις	Κρύο	Κανονικό	Αδύναμος	Ναι
Η14	Βροχόπτωση	Ήπια	Υψηλό	Δυνατός	Όχι

Πρόβλεψη

Πρέπει να βρούμε ποιες ιδιότητες θα έχει ο κόμβος ρίζας στο δέντρο απόφασής μας. Το κέρδος υπολογίζεται και για τις τέσσερις ιδιότητες:

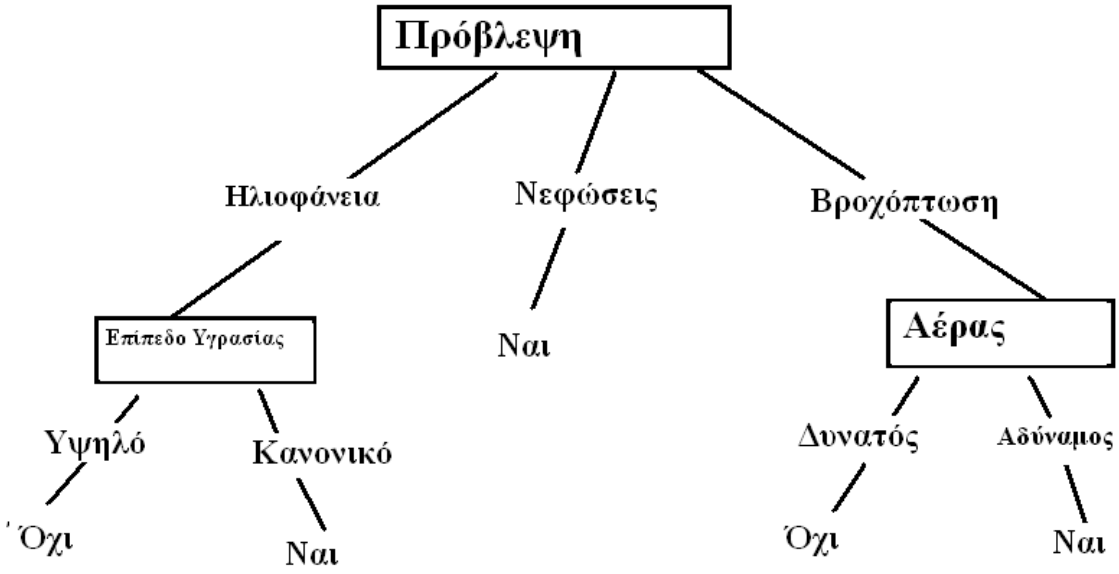
Κέρδος (S, πρόβλεψη) = 0.246

Κέρδος (S, θερμοκρασία) = 0.029
Κέρδος (S, υγρασία) = 0.151
Κέρδος (S, αέρας) = 0.048 (υπολογισμένος στο παράδειγμα 2)
Η ιδιότητα πρόβλεψη έχει το υψηλότερο κέρδος, επομένως χρησιμοποιείται ως ιδιότητες απόφασης στον κόμβο ρίζας.

Δεδομένου ότι η πρόβλεψη έχει τρεις πιθανές τιμές, ο κόμβος ρίζας έχει τρεις κλάδους (Ηλιοφάνεια, Νεφώσεις, Βροχόπτωση). Η επόμενη ερώτηση είναι «ποιες ιδιότητες πρέπει να εξεταστούν στον κόμβο κλάδων ηλιοφάνεια;» Χρησιμοποίει την πρόβλεψη στη ρίζα, αποφασίζουμε μόνο σχετικά με τις υπόλοιπες τρεις ιδιότητες: Επίπεδο Υγρασίας, θερμοκρασία, ή αέρας.

Ηλιοφάνεια = { Η 1, Η 2, Η 8, Η 9, Η 11} = 5 παραδείγματα από τον πίνακα 1 με την πρόβλεψη = Ηλιοφάνεια
Κέρδος (Ηλιοφάνεια, Επίπεδο Υγρασίας) = 0.970
Κέρδος (Ηλιοφάνεια, θερμοκρασία) = 0.570
Κέρδος (Ηλιοφάνεια, αέρας) = 0.019

Η υγρασία έχει το υψηλότερο κέρδος επομένως, χρησιμοποιείται ως κόμβος απόφασης. Αυτή η διαδικασία συνεχίζεται έως ότου ταξινομείται τέλεια όλο το στοιχείο ή τρέχουμε από τις ιδιότητες.



Σχήμα 3.1 Πρόβλεψη καιρού

Η τελική απόφαση = δέντρο

Το δέντρο απόφασης μπορεί επίσης να εκφραστεί με τη μορφή κανόνα:
ΕΑΝ πρόβλεψη = Ηλιοφάνεια ΚΑΙ υγρασία = υψηλό ΕΠΕΙΤΑ Διεξαγωγή Αγώνα = Όχι
ΕΑΝ πρόβλεψη = Βροχόπτωση ΚΑΙ υγρασία = υψηλό ΕΠΕΙΤΑ Διεξαγωγή Αγώνα = Όχι

ΕΑΝ πρόβλεψη = Βροχόπτωση ΚΑΙ αέρας = Δυνατός ΕΠΕΙΤΑ Διεξαγωγή Αγώνα = ναι
ΕΑΝ η πρόβλεψη = Νεφώσεις ΕΠΕΙΤΑ Διεξαγωγή Αγώνα = ναι
ΕΑΝ πρόβλεψη = Βροχόπτωση ΚΑΙ αέρας = Αδύναμος ΕΠΕΙΤΑ Διεξαγωγή Αγώνα = ναι

Ο ID3 έχει ενσωματωθεί σε διάφορες εμπορικές συσκευασίες ως κανόνας-επαγωγής. Μερικές συγκεκριμένες εφαρμογές περιλαμβάνουν την ιατρική διάγνωση, την αξιολόγηση του πιστωτικού κινδύνου των εφαρμογών δανείου , και την ταξινόμηση αναζήτησης στο διαδίκτυο.

3.1.1.2 Αλγόριθμος C4.5

Ο αλγόριθμος C4.5 (Quinlan, 1993), είναι ένας από τους πλέον γνωστούς και ευρύτατα διαδεδομένους σε εμπορικές εφαρμογές, αλγορίθμους ταξινόμησης που βασίζεται στα δέντρα απόφασης. Ουσιαστικά αποτελεί επέκταση του παλαιότερου αλγορίθμου ID3 του Quinlan (1986). Βασισμένος στην τεχνική *διαίρεση-και-κατάκτηση* (***divide-and-conquer***) των Hunt et al (1966) για την ανάπτυξη δέντρων απόφασης, ο C4.5 αναπτύσσει δέντρα απόφασης από ένα σύνολο δεδομένων εκπαίδευσης, όπως ακριβώς και ο ID3, χρησιμοποιώντας την έννοια της *εντροπίας* (***information entropy***).

Ας θεωρήσουμε ένα σύνολο δεδομένων εκπαίδευσης *T* που περιλαμβάνει προ-ταξινομημένα δεδομένα. Σε κάθε κόμβο του παραγόμενου δέντρου απόφασης, ο αλγόριθμος επιλέγει ένα *χαρακτηριστικό* (***attribute***) του συνόλου των δεδομένων εκπαίδευσης, το οποίο χωρίζει με τον πλέον αποτελεσματικό τρόπο το *T*, σε υποσύνολα $T_1, T_2, ..., T_n$, κάθε ένα από τα οποία περιλαμβάνει αντικείμενα που ανήκουν σε μια μόνο κλάση. Το κριτήριο για την ανωτέρω επιλογή και διαχωρισμό, είναι το *κριτήριο* ***Gain Ratio*** (***Gain Ration criterion***) που στην ουσία είναι το *κανονικοποιημένο κέρδος πληροφορίας* (***normalized information gain***).

Συγκεκριμένα, το ***Gain Ratio Criterion*** είναι παρόμοιο με το ***Gain Criterion*** που χρησιμοποιείται από τον αλγόριθμο ταξινόμησης ID3 (Quinlan, 1986), με την προσθήκη ενός επιπλέον βήματος στο τέλος. Ο αλγόριθμος ID3 προσπαθεί να χωρίσει το σύνολο εκπαίδευσης *T* σε *n* υποσύνολα με βάση έναν έλεγχο *Q*, που είναι ένα από τα χαρακτηριστικά του συνόλου εκπαίδευσης. Το καλύτερο χαρακτηριστικό/έλεγχος επιλέγεται μέσω του ***Gain Criterion***. Στην αρχή ο αλγόριθμος ψάχνει το σύνολο εκπαίδευσης *T* και υπολογίζει, για κάθε πιθανή τιμή *qz* (*z = 1,..,n*), κάθε χαρακτηριστικού *Q*, τον αριθμό των θετικών, των αρνητικών και των συνολικών εμφανίσεων (αντικείμενα που φέρουν ή όχι την τιμή *qz* στο χαρακτηριστικό *Q*). Στην συνέχεια, ο αλγόριθμος υπολογίζει την *εντροπία* (***entropy***) του συνόλου *T* ως ακολούθως:

$$Info(T) = -\sum_{h=1}^{f}\left[\left(\frac{freq(C_h,T)}{T}\right)\cdot \log_2\left(\frac{freq(C_h,T)}{|T|}\right)\right] \quad (3.1)$$

όπου $|T|$ είναι ο αριθμός των αντικειμένων στο T, $freq(Ch,T)$ είναι ο αριθμός των αντικειμένων στο T που ανήκουν στην κλάση Ch , ενώ f είναι ο αριθμός των κλάσεων. Έπειτα, το *Gain Criterion* υπολογίζει για κάθε χαρακτηριστικό Q τις απαιτήσεις πληροφορίας για την περίπτωση που το σύνολο χωρίζεται βάσει του Q, μέσω του κανονικοποιημένου αθροίσματος όλων των υποσυνόλων (Tz):

$$Info_q(Q) = -\sum_{z=1}^{n}\left[\frac{|T_z|}{|T|} * Info(T_z)\right] \quad (3.2)$$

Τελικά, η ποσότητα $Gain(Q) = Info(T) – Infoq(Q)$ είναι το *κέρδος πληροφορίας* (*information gain*), αν το T χωρίζεται χρησιμοποιώντας το χαρακτηριστικό Q.

Εφαρμόζοντας την παραπάνω επαναληπτική διαδικασία για κάθε πιθανό διαχωρισμό του T, τελικά το *Gain Criterion* επιλέγει το χαρακτηριστικό/έλεγχο με το μεγαλύτερο κέρδος πληροφορίας. Από την άλλη μεριά το *Gain Ratio Criterion* στοχεύει να αφαιρέσει από το *Gain Criterion* την τάση να επιλέγει χαρακτηριστικά/ελέγχους με τον μεγαλύτερο αριθμό πιθανών τιμών. Σύμφωνα με το *Gain Criterion*, ένα χαρακτηριστικό/έλεγχος με διαφορετική τιμή για κάθε αντικείμενο του συνόλου εκπαίδευσης έχει πάντα το μέγιστο κέρδος πληροφορίας. Η επιλογή ωστόσο ένα τέτοιου χαρακτηριστικού για τον χωρισμό του αρχικού συνόλου εκπαίδευσης, οδηγεί σε ένα μεγάλο αριθμό αχρείαστων υποσυνόλων, καθώς κάθε ένα από αυτά περιλαμβάνει ένα μόνο αντικείμενο. Για την αποφυγή αυτής της ακραίας περίπτωσης, το *Gain Ratio Criterion* περιορίζει τα χαρακτηριστικά με πολλές τιμές, ακόμα και αν αυτά παρέχουν μεγαλύτερη πληροφορία. Έτσι, το επιπρόσθετο βήμα για τον υπολογισμό του *Gain Ratio Criterion* αφορά την κανονικοποίηση των αποτελεσμάτων του *Gain Criterion*. Εν τέλει, υπάρχουν οι ακόλουθοι δύο υπολογισμοί:

$$SplitInfo(Q) = -\sum_{z=1}^{n}\left[\frac{|T_z|}{|T|} * \log_2\left(\frac{|T_z|}{|T|}\right)\right] \quad (3.3)$$

$$Gainratio(Q) = \frac{Gain(Q)}{SplitInfo(Q)} \quad (3.4)$$

όπου *Gain ratio(Q)* είναι το κανονικοποιημένο *Gain(Q)*.

Αφού υπολογιστεί το κριτήριο *Gain Ratio* για κάθε πιθανό χαρακτηριστικό/έλεγχο του συνόλου εκπαίδευσης *T*, ο αλγόριθμος επιλέγει ένα από τα παραπάνω χαρακτηριστικά, ως το πλέον κατάλληλο για τον διαχωρισμό του συνόλου εκπαίδευσης *T*. Συγκεκριμένα, επιλέγεται το χαρακτηριστικό με το μεγαλύτερο κανονικοποιημένο κέρδος πληροφορίας, ανάμεσα στα χαρακτηριστικά που έχουν τουλάχιστον το μέσο κέρδος πληροφορίας. Η συγκεκριμένη επαναληπτική στρατηγική *χωρίσματος* (*partitioning*) έχει σαν αποτέλεσμα την δημιουργία δέντρων απόφασης, τα οποία είναι συνεπή με τα δεδομένα εκπαίδευσης, στο μέτρο του δυνατού.

Αξίζει να σημειωθεί ότι στις περισσότερες πρακτικές εφαρμογές τα δεδομένα περιλαμβάνουν θόρυβο (λανθασμένες τιμές χαρακτηριστικών ή/και λανθασμένη ταξινόμηση αντικειμένων). Προκειμένου να αντιμετωπίσει αποτελεσματικά τον θόρυβο αυτό, ο C4.5 δημιουργεί δέντρα αποφάσεων με αρκετά μεγάλο μέγεθος. Υπάρχει ωστόσο η δυνατότητα *κλαδέματος* (*pruning*) των δέντρων αυτών, αναγνωρίζοντας υποδέντρα τους με μικρές δυνατότητες ακριβούς ταξινόμησης, και αντικαθιστώντας τα με φύλλα (Kohavi & Quinlan, 1999).

Στην συνέχεια, το παραγόμενο δέντρο χρησιμοποιείται για την ταξινόμηση νέων δεδομένων. Αρχίζοντας από την ρίζα του παραγόμενου δέντρου απόφασης, ελέγχεται η τιμή του χαρακτηριστικού που υπάρχει εκεί. Ανάλογα με την τιμή αυτή, γίνεται η κατάλληλη διακλάδωση, γίνεται ξανά έλεγχος κ.λ.π. Η παραπάνω διαδικασία από

πάνω προς τα κάτω συνεχίζεται, έως ότου καταλήξουμε σε ένα φύλλο το οποίο προσδιορίζει την κλάση στην οποία ανήκει το προς ταξινόμηση αντικείμενο.

Βελτιώσεις από τον ID3 αλγόριθμο

Το C4.5 έκανε διάφορες βελτιώσεις στον ID3. Μερικές από αυτές είναι:

- Διαχειρισμένος και τις συνεχείς και ιδιαίτερες ιδιότητες - προκειμένου να αντιμετωπιστούν οι συνεχείς ιδιότητες, C4.5 δημιουργεί μια ευαισθησία και χωρίζει έπειτα τον κατάλογο σε εκείνοι η των οποίων αξία ιδιοτήτων είναι επάνω από την ευαισθησία και εκείνοι που είναι λιγότερο ή ίσο προς αυτό.

- Τα διαχειριζόμενα στοιχεία κατάρτισης με να λείψουν αποδίδουν τις τιμές - C4.5 επιτρέπει στις τιμές ιδιοτήτων για να χαρακτηριστεί όπως; για να λείψει. Οι ελλείπουσες τιμές ιδιοτήτων απλά δεν χρησιμοποιούνται στους υπολογισμούς κέρδους και εντροπίας.

- Διαχειριζόμενες ιδιότητες με τις διαφορετικές δαπάνες.

- Δέντρα περικοπής μετά από τη δημιουργία - C4.5 επιστρέφει μέσω του δέντρου μόλις δημιουργηθεί και προσπαθήσει να αφαιρέσει τους κλάδους που δεν βοηθούν με την αντικατάσταση τους με τους κόμβους φύλλων (Μαστρογιάννης, 2009).

3.1.1.3 SLIQ

Ο SLIQ (Supervised Learning In Quest) που αναπτύχθηκε από την ομάδα IBM Quest το 1996, είναι ένας ταξινομητής δέντρων απόφασης με σκοπό να ταξινομήσει τα μεγάλα δεδομένα. Χρησιμοποιεί μια τεχνική προταξηνόμισης στο στάδιο αύξησης του δέντρου. Αυτό βοηθά στο να αποφεύγετε η δαπανηρή ταξινόμηση σε κάθε κόμβο. Ο

SLIQ κρατά έναν χωριστό ταξινομημένο κατάλογο για κάθε συνεχή ιδιότητα και έναν χωριστό κατάλογο αποκαλούμενο κατάλογο κλάσης. Μια είσοδος στον κατάλογο κλάσης αντιστοιχεί σε ένα αντικείμενο στοιχείων, και έχει μια ετικέτα κλάσης και ένα όνομα του κόμβου που ανήκει στο δέντρο απόφασης. Μια είσοδος στον ταξινομημένο κατάλογο ιδιοτήτων έχει μια αξία ιδιοτήτων και το ευρετήριο του αντικειμένου στοιχείων στον κατάλογο κλάσης.

Ο SLIQ αυξάνει το δέντρο απόφασης με τη μέθοδο πρώτου εύρους. Για κάθε ιδιότητα, ανιχνεύει τον αντίστοιχο ταξινομημένο κατάλογο και υπολογίζει τις τιμές εντροπίας των ευδιάκριτων τιμών όλων των κόμβων στα σύνορα του δέντρου απόφασης ταυτόχρονα. Αφότου έχουν υπολογιστεί οι τιμές εντροπίας για κάθε ιδιότητα, μια ιδιότητα επιλέγεται για τη διάσπαση κάθε κόμβου στα παρόντα σύνορα, και επεκτείνονται για να έχουν νέα σύνορα.

Κατόπιν μια παραπάνω ανίχνευση του ταξινομημένου καταλόγου ιδιοτήτων εκτελείται για να ενημερώσει τον κατάλογο κλάσης για τους νέους κόμβους. Ενώ ο SLIQ χειρίζεται τα στοιχεία που είναι εγκατεστημένα σε ένα δίσκο και είναι πολύ μεγάλα για να χωρέσουν στη μνήμη, απαιτεί ακόμα κάποιες πληροφορίες για να εγκατασταθούν στη μνήμη η οποία αυξάνεται ανάλογα με τον αριθμό αρχείων εισόδου, βάζοντας ένα όριο στο μέγεθος των στοιχείων κατάρτισης. Η ομάδα αναζήτησης έχει σχεδιάσει πρόσφατα έναν νέο αλγόριθμο ταξινόμησης βασισμένο σε δέντρο απόφαση αποκαλούμενο SPRINT (Scalable PaRallelizable INduction of decision Trees) που αφαιρεί όλους τους περιορισμούς μνήμης (Agrawal et al., 1996; Mehta et al., 1996).

3.1.1.4 SPRINT

Από το IBM Almaden Research Center (John Shafer, Rakeeh Agrawal, Manish Mehta) το 1981 παρουσιάστηκε ένας νέος αλγόριθμος ταξινόμησης βασισμένος στα

δέντρα απόφασης, καλούμενος SPRINT που αφαιρεί όλους τους περιορισμούς μνήμης και είναι γρήγορος και επιδεκτικότερος στη μέτρησης με μια πρότυπη κλίμακα.

Ο σκοπός του αλγόριθμου είναι να είναι ευκολότερος στην παραλληλοποίηση, επιτρέποντας σε πολλούς επεξεργαστές να εργαστούν μαζί ώστε να χτίσουν ένα ενιαίο συνεπές μοντέλο. Ο Sprint φέρετε να είναι εξαιρετικός στη μέτρηση με μια πρότυπη κλίμακα, στην επιτάχυνση και την επιμήκυνση ιδιοτήτων.

Ο συνδυασμός αυτών των χαρακτηριστικών κάνει τον αλγόριθμο Sprint ένα ιδανικό εργαλείο για την ανάσυρση δεδομένων.

Δομές δεδομένων

Ο SPRINT δημιουργεί αρχικά έναν κατάλογο ιδιοτήτων για κάθε ιδιότητα στα στοιχεία. Μπαίνει σε αυτούς τους καταλόγους, τους όποιους αποκαλούμε αρχεία ιδιοτήτων, αποτελούμενα από μια αξία ιδιοτήτων, μια ετικέτα κλάσης, και το ευρετήριο του αρχείου από το οποίο ελήφθησαν. Οι αρχικοί κατάλογοι για τις συνεχείς ιδιότητες ταξινομούνται κατά την αξία ιδιοτήτων μία φορά όταν δημιουργούνται. Εάν όλα τα στοιχείο δεν χωρούν στη μνήμη, αποδίδονται σε δίσκο.

Οι αρχικοί κατάλογοι που δημιουργούνται από το σύνολο κατάρτισης είναι συνδεμένοι με τη ρίζα του δέντρου ταξινόμησης. Δεδομένου ότι το δέντρο αυξάνεται και οι κόμβοι είναι χωρισμένοι για να δημιουργήσουν ένα νέο κόμβο, οι κατάλογοι ιδιοτήτων που ανήκουν σε κάθε κόμβο χωρίζονται και συνδέονται με τον κόμβο που παρήχθηκε. Όταν μια λίστα έχει παραλληλιστεί η σειρά των στοιχείων στη λίστα διατηρείτε. Αυτή η παραλληλισμένη λίστα δεν χρειάζεται να επαναταξηνομηθεί (Shafer et al., 1996).

3.1.1.5 CART

Ο CART (classification and regression tree) είναι ένας σύγχρονος αλγόριθμος που

αναπτύχτηκε από τον Breiman το 1984, μέθοδος της ανάσυρσης δεδομένων που χρησιμοποιεί τα δέντρα απόφασης και μπορεί να χρησιμοποιηθεί για ποικίλες επιχειρησιακές και επιστημονικές εφαρμογές. Τα πλεονεκτήματά του περιλαμβάνουν τη γρήγορη διορατικότητα στα πρότυπα βάσεων δεδομένων και στις σημαντικές σχέσεις χρησιμοποιώντας απλά εργαλεία όπως γραφικές παραστάσεις, διαγράμματα και εκθέσεις.

Είτε για αρχάριες είτε για προχωρημένες μεθόδους καθοδηγούμενης από το μενού λειτουργίας, ο Cart είναι διαθέσιμος για να προσαρμοστεί στα επίπεδα άνεσης του χρήστη. Για τους πιο απαιτητικούς χρήστες, μια εντολή εκκίνησης είναι διαθέσιμη. Το πρόγραμμα μπορεί εύκολα να χειριστεί τα πολύ μεγάλα σύνολα δεδομένων, και τα καταφέρνει καλύτερα στους σημερινούς γρήγορους και αποδοτικούς βιομηχανικούς κεντρικούς υπολογιστές.

Η θεωρία πίσω από τα δέντρα απόφασης είναι σχετικά απλή από την άποψη της επεξηγηματικότητας τι κάνει το μοντέλο, ακόμη και οι επανάληψη δραστηριοτήτων έχουν εξεταστεί καλά από ποικίλους όρους.

Λειτουργικά χαρακτηριστικά

Η μηχανή μεταφράσεων στοιχείων σε αυτό το λογισμικό μπορεί να χειριστεί τις μετατροπές για 80 μορφές αρχείου, συμπεριλαμβανομένων των δημοφιλών προγραμμάτων στατιστικών όπως τη SAS και SPSS, και τους πρότυπους υπολογισμούς με λογιστικό φύλλο (spreadsheet) όπως το EXCEL και το Lotus 123. Οποιοδήποτε μοντέλο Cart μπορεί να επεκταθεί εύκολα όταν μεταφράζεται σε μια από τις υποστηριζόμενες γλώσσες. Η λογική απόφασης που χρησιμοποιείται σε αυτά τα μοντέλα εφαρμόζεται αυτόματα, και τον κωδικό πηγής που προκύπτει τον εναποθέτει στις εξωτερικές εφαρμογές.

3.1.2 Αλγόριθμοι στις μεθόδους κανόνων απόφασης

3.1.2.1 Ο αλγόριθμος CN2

Ο αλγόριθμος CN2 (Clark & Niblett, 1989) είναι ένας από τους βασικότερους αλγορίθμους που βασίζονται στους κανόνες απόφασης. Συνδυάζει την αποτελεσματικότητα του ID3 (Quinlan, 1986), καθώς και την ικανότητα του τελευταίου να διαχειρίζεται δεδομένα με θόρυβο, με την χρήση κανόνων της μορφής «If *A* Then *B*» και την ευέλικτη στρατηγική έρευνας των αλγορίθμων της οικογένειας AQ (Michalski & Chilausky, 1980). Ο CN2, χρησιμοποιώντας μια ευρεστική συνάρτηση για τον τερματισμό της διαδικασίας εύρεσης κανόνων, εξάγει ένα σύνολο «If *A* Then *B*» κανόνων, που είναι γνωστό ως *λίστα απόφασης* (***decision list***) (Rivest, 1987). Έπειτα, το σύνολο αυτό των κανόνων χρησιμοποιείται για την ταξινόμηση νέων δεδομένων.

Ειδικότερα, ο CN2, είναι ένας επαναληπτικός αλγόριθμος, ο οποίος σε κάθε επανάληψή του ψάχνει για ένα *σύμπλεγμα* (***complex***) που είναι σε θέση να καλύψει έναν μεγάλο αριθμό παραδειγμάτων μιας κλάσης *c*, και μερικών από τις άλλες κλάσεις. Το σύμπλεγμα αυτό, αφενός πρέπει να έχει υψηλές ικανότητες πρόβλεψης και αφετέρου πρέπει να είναι αξιόπιστο, βάσει των *συναρτήσεων αξιολόγησης* (***evaluation functions***) του αλγορίθμου. Η διαδικασία εύρεσης τέτοιων συμπλεγμάτων λαμβάνει χώρα βάσει μιας διαδικασίας κλαδέματος, από γενικότερες σε ειδικότερες περιπτώσεις.

Σε κάθε στάδιο της παραπάνω διαδικασίας έρευνας, ο αλγόριθμος CN2 διατηρεί ένα περιορισμένο σύνολο, το επονομαζόμενο «***star S***», των καλύτερων συμπλεγμάτων που έχουν βρεθεί μέχρι στιγμής. Ο αλγόριθμος εξετάζει μόνο εξειδικεύσεις του συγκεκριμένου συνόλου, ερευνώντας ειδικότερα τον χώρο των συμπλεγμάτων. Ένα σύμπλεγμα εξειδικεύεται, είτε με την πρόσθεση σε αυτό ενός *ενωτικού όρου* (***conjunctive term***), είτε με την αφαίρεση ενός *διαζευκτικού όρου* (***disjunctive term***)

από έναν από τους *επιλογείς* (***selectors***) του. Σε κάθε περίπτωση, με το τέλος του βήματος αυτού, το σύνολο «***star S***» οριστικοποιείται, με την αφαίρεση από αυτό εκείνων των στοιχείων του, που παρουσιάζουν την χαμηλότερη «βαθμολογία» βάσει των δύο συναρτήσεων αξιολόγησης των ισάριθμων ευρεστικών διαδικασιών απόφασης που λαμβάνουν χώρα στην εκπαίδευση του αλγορίθμου.

Συγκεκριμένα, στην πρώτη από τις ευρεστικές του αποφάσεις, ο αλγόριθμος CN2 πρέπει να αποτιμήσει την ποιότητα των συμπλεγμάτων, καθορίζοντας, αφενός την ενδεχόμενη αντικατάσταση του καλύτερου εξ'αυτών από ένα νέο, και αφετέρου εκείνα τα συμπλέγματα που πρέπει να παραβλεφτούν στην περίπτωση που το μέγεθος του «***star S***» ξεπεράσει ένα καθορισμένο όριο. Για να ληφθούν οι παραπάνω αποφάσεις πρέπει να υπολογιστεί το μέγεθος της *εντροπίας* (***entropy***). Έτσι, αν *E* είναι το σύνολο των παραδειγμάτων που καλύπτει ένα σύμπλεγμα, και $P = (p_1, p_2, ..., p_n)$ είναι η κατανομή πιθανότητας των παραδειγμάτων αυτών στις ***n*** διαθέσιμες κλάσεις, τότε:

$$Entropy = -\sum_i \left[p_i \cdot \log_2 (p_i) \right] \quad (3.5)$$

Η συγκεκριμένη συνάρτηση αξιολόγησης τείνει να επιλέγει συμπλέγματα που καλύπτουν ένα μεγάλο αριθμό παραδειγμάτων μιας συγκεκριμένης κλάσης και έναν αντίστοιχο μικρό αριθμό άλλων κλάσεων.

Η δεύτερη συνάρτηση αξιολόγησης εξετάζει πόσο σημαντικό είναι ένα σύμπλεγμα. Για να καθοριστεί η παραπάνω *σημαντικότητα* (***significance***), υπολογίζεται το επονομαζόμενο «***likelihood ratio statistic***» (Kalbfleish, 1979). Είναι:

$$LRS = 2 \cdot \sum_{i=1}^{n} \left[f_i \cdot \log (f_i / e_i) \right] \quad (3.6)$$

Όπου, $F = (f_1, f_2, ..., f_n)$ είναι η παρατηρηθείσα συχνότητα κατανομής των παραδειγμάτων ανάμεσα στις διάφορες κλάσεις σε ένα δεδομένο σύμπλεγμα και

$E = (e_1, e_2, ..., e_n)$ είναι η αναμενόμενη συχνότητα κατανομής του ίδιου αριθμού παραδειγμάτων, υπό την παραδοχή ότι το σύμπλεγμα επιλέγει παραδείγματα τυχαία.

Με άλλα λόγια, οι παραπάνω δύο συναρτήσεις αξιολόγησης, καθορίζουν αν τα συμπλέγματα που βρέθηκαν στην διάρκεια της έρευνας είναι ταυτόχρονα «καλά» και «αξιόπιστα». Η εύρεση ενός τέτοιου συμπλέγματος, έχει σαν αποτέλεσμα, ο αλγόριθμος CN2 να αφαιρέσει από το σύνολο εκπαίδευσης τα παραδείγματα που το σύμπλεγμα καλύπτει, εξάγοντας ταυτόχρονα έναν «If *A* Then *B*» κανόνα. Η διαδικασία επαναλαμβάνεται μέχρι εκείνο το σημείο, στο οποίο δεν μπορούν να βρεθούν πλέον άλλα ικανοποιητικά συμπλέγματα. (Clark & Niblett, 1989; Μαστρογιάννης, 2009).

3.1.2.2 Αλγόριθμος AQ

ΠΕΡΙΓΡΑΦΗ

Η λειτουργία του Aq προκαλεί ένα σύνολο κανόνων από τα παραδείγματα που δίνονται σαν σχέση στη βάση δεδομένων Prolog. Η έξοδος είναι ένα πλαίσιο που περιέχει το νέο κανόνα.

Όπως τους άλλους προτασιακούς αρχαρίους αλγοριθμους, ο Aq απαιτεί μια δήλωση τρόπου για να προσδιορίσει τους τύπους ιδιοτήτων που έχει η σχέση.

Η δήλωση τρόπου καθορίζει τη σειρά των τιμών που επιτρέπεται να πάρει κάθε ιδιότητα. Η τελευταία ιδιότητα λαμβάνεται πάντα ως η κλάση (Βαζιργιάννης & Χαλκίδη, 2003).

3.1.2.3 Ο αλγόριθμος CL^2

Ο CL^2 (Boutsinas et al., 2004) είναι ένας αλγόριθμος ταξινόμησης που βασίζεται σε κανόνες, ο οποίος εν πολλοίς δομείται χρησιμοποιώντας αρχές της διαδικασίας της ομαδοποίησης. Ειδικότερα, η βασική ιδέα του αλγορίθμου, είναι ο διαχωρισμός της βάσης των διαθέσιμων (αριθμητικών ή κατηγορικών) δεδομένων (που εννοείται ότι είναι προ-ταξινομημένη) σε ομοιογενή υποσύνολα, βάσει των τιμών των κλάσεων. Δηλαδή, τα αντικείμενα κάθε τέτοιου υποσυνόλου ανήκουν όλα στην ίδια κλάση. Στην συνέχεια, εφαρμόζοντας έναν αλγόριθμο ομαδοποίησης σε κάθε ένα από τα παραπάνω υποσύνολα (π.χ. τον k-means για αριθμητικά δεδομένα ή τον k-modes για κατηγορικά δεδομένα), βρίσκουμε τις ομάδες στις οποίες διαχωρίζεται κάθε υποσύνολο. Θεωρώντας τα κέντρα των παραπάνω ομάδων σαν μια πιθανή περιγραφή της κλάσης του υποσυνόλου, ο αλγόριθμος εξάγει από κάθε ομάδα έναν πιθανό κανόνα (υπο την μορφή ένωσης ζευγών χαρακτηριστικών-τιμών) που περιγράφει την συγκεκριμένη ομάδα και κατ'επέκταση την κλάση του υποσυνόλου.

Ο τρόπος με τον οποίο εξάγονται οι κανόνες από τα αντικείμενα των εκάστοτε ομάδων, βασίζεται στην θεώρηση όλων των πιθανών χαρακτηριστικών για όλα τα πιθανά μήκη κανόνων, χρησιμοποιώντας ταυτόχρονα μια ευρεστική διαδικασία περιορισμού του ενδεχομένως μεγάλου αριθμού των παραπάνων συνδυασμών. Με δεδομένη την διαβάθμιση των χαρακτηριστικών βάσει της σπουδαιότητάς τους, η ομαδοποίηση των υποσυνόλων και η εξαγωγή των πιθανών κανόνων από τις ομάδες που βρέθηκαν, προχωρούν, αφαιρώντας σταδιακά από την διαδικασία εύρεσης των κανόνων τα λιγότερο σημαντικά χαρακτηριστικά. Πρέπει να σημειωθεί ότι ένας πιθανός κανόνας, οριστικοποιείται σαν κανόνας ταξινόμησης του αλγορίθμου, όταν η ακρίβεια του υπερβαίνει ένα καθορισμένο από το χρήστη όριο, το οποίο πρέπει να υπερβαίνει το 50%.

3.1.2.4 Κανόνες εκμάθησης

Περίληψη

Η βασισμένη στους κανόνες εκμάθησης είναι μια μέθοδος, όπως το δέντρο απόφασης στην ταξινόμηση:
από τα στοιχεία, η έννοια ή η υποδομή που εξάγεται ως σύνολο
κανόνων είναι παρόμοιοι με αυτούς της ανθρώπινης γνώσης (Cohen, 1995.).

Οι κανόνες που παράγονται από το RIPPER, που χρησιμοποιούν το λογισμικό Weka, αναλύθηκαν λεπτομερώς προκειμένου να λάβουν κάποια άμεση γνώση για τα θέματα του διαβήτη και του Εθνικού Ελληνικού Εισοδήματος.

Εισαγωγή

Η ανάγκη για τις τεχνικές εκμάθησης μηχανών προόδου είναι όχι μόνο για τη διευκόλυνση των διαδικασιών αυτοματοποίησης αλλά και λόγω της ανικανότητάς μας να καταλάβουμε αυτά τα μεγάλα στοιχεία. Για να εξαγάγουν τη γνώση για μια δικτυακή περιοχή από τα στοιχεία που συλλέχτηκαν, ειδικά για λόγους ταξινόμησης, πολλές τεχνικές έχουν προταθεί στις διάφορες προσεγγίσεις (όπως τις παραμετρικές, μη παραμετρικές ή μη αριθμητικές μεθόδους).

Εντούτοις, η προσέγγιση ότι είναι καταλληλότερη και παρόμοια με την ανθρώπινη γνώση που καταλαβαίνει είναι η προσέγγιση βασισμένη στην εκμάθηση. Η βασισμένη στους κανόνες εκμάθηση εξάγει την κριμένη έννοια ή τη δομή των στοιχείων υπό μορφή συνόλου κανόνων, οι οποίοι βρίσκουν την αναλογία τους στα δέντρα απόφασης. Παρόλα αυτά , τα σύνολα κανόνων είναι σχετικά κατανοητά από τους ανθρώπους, και σύμφωνα με τον Sedgewick (1998), τα συστήματα εκμάθησης κανόνων ξεπερνούν τα δέντρων απόφασης σε πολλά προβλήματα.

Ένα πείραμα της εκμάθησης από τα στοιχεία πραγματοποιήθηκε χρησιμοποιώντας θηλυκού γένους ασθενείς διαβήτη βασισμένο στις τεχνικές εκμάθησης κανόνων (αλγόριθμος RIPPER). Εκτός αυτού, η τεχνική επιλογής χαρακτηριστικών γνωρισμάτων εκτελέσθηκε επίσης στα στοιχεία για να συγκρίνει με τους κανόνες και να επισημάνει μερικά πιο πληροφοριακά χαρακτηριστικά γνωρίσματα στην πρόβλεψη της αρχής των περιπτώσεων διαβήτη.

Επιπλέον, ένα στοιχείο απογραφής όσον αφορά εάν ένα εισόδημα ενός προσώπου είναι μεγαλύτερο από 50χιλ. Euro εξετάστηκε επίσης προκειμένου να επεξηγήσει στον τρόπο με τον οποίο οι αποκτηθέντες κανόνες απεικόνισαν τη γνώση μιας δικτυακής περιοχής (Fellbaum, 1998).

Μέθοδοι βασισμένες στους κανόνες εκμάθησης

Ο κανόνας εκμάθησης και η εκμάθηση από δέντρα απόφασης ανήκουν στο εποπτευμένο σχέδιο εκμάθησης, μέσα από το όποιο ένας δάσκαλος παρέχει μια ετικέτα κλάσης για κάθε περίπτωση σε ένα σύνολο κατάρτισης και επιδιώκει να κατασκευάσει ένα μοντέλο ταξινόμησης των στοιχείων, υπολογίζει τις άγνωστες παραμέτρους που μπορούν να βοηθήσουν να μειώσουν το σφάλμα στο σύνολο κατάρτισης.

Η προσληφθείσα περιγραφή ενός στοιχείου μπορεί να μπει σε διάφορες μορφές, εξαρτώμενη από τον ταξινομητή που υιοθετείται, όπως σύνορα απόφασης (hyperplane) ή λίγο περισσότεροι διαισθητικοί τρόποι όπως από μια ακολουθία ερωτήσεων (δηλ. δέντρο απόφασης) ή ένα σύνολο κανόνων (δηλ. εάν μια περίπτωση ικανοποιεί αυτούς τους όρους, τότε ταξινομείτε ως W1).

Είναι προφανές ότι το δέντρο και οι κανόνες απόφασης είναι αρκετά συνδεδεμένοι μεταξύ τους δεδομένου ότι το δέντρο απόφασης μπορεί να ερμηνευθεί ως σύνολο κανόνων με το τρέξιμο του δέντρου με τον τρόπο πρώτη εις βάθος αναζήτησης και αντίστροφα.

Στην πραγματικότητα, πολλές από τις τεχνικές που χρησιμοποίησαν τους σύγχρονους κανόνες μάθησης έχουν προσαρμοστεί από δέντρα απόφασης. Πολλά συστήματα εκμάθησης δέντρων απόφασης χρησιμοποιούν υπερκάλυψη και απλοποίηση , ή τεχνική περικοπής ,παρά τον προσδιορισμό μερικών περιπτώσεων στάσης ,για τη δόμηση του δέντρο. Εδώ η υπόθεση διαμορφώνεται με το να αυξηθεί πρώτα ένα σύνθετο δέντρο που μπορεί να υπερκαλύψει τα δεδομένα, και έπειτα να απλουστεύσει ή να περικόψει το δέντρο. Μια αποτελεσματική τεχνική περικοπής μέσα στα δέντρα είναι η μείωση περικοπής σφάλματος (REP), η οποία έχει προσαρμοστεί στους κανόνες.

Μείωση περικοπή σφάλματος (REP)

Στο REP για τους κανόνες, το στοιχείο κατάρτισης είναι χωρισμένο σε ένα αυξανόμενο σύνολο και σε ένα σύνολο περικοπής. Κατ' αρχάς, διαμορφώνεται ένα αρχικό σύνολο κανόνων που υπερκαλύπτει το αυξανόμενο σύνολο, το οποίο χρησιμοποιεί κάποια ευρετική μέθοδο (δηλ. κέρδος πληροφοριών). Το σύνολο κανόνων απλοποιείται έπειτα από μερικούς χειριστές περικοπής.

Οι χειριστές θα διέγραφαν οποιοδήποτε μονό όρο ή μονό κανόνα που παράγει τη μέγιστη μείωση του σφάλματος στο σύνολο περικοπής. Η απλοποίηση τελειώνει όταν οποιοσδήποτε χειριστής περικοπής θα αύξανε το σφάλμα στο σύνολο περικοπής (Sedgewick, 1998).

Εντούτοις, το REP είναι υπολογιστικά ακριβό για τα μεγάλα σύνολα δεδομένων (δηλ. Ο (n^4), λαμβάνοντας υπόψη τα αρκετά θορυβώδη στοιχεία).
Σε απάντηση στην ανεπάρκεια του REP, ένας νέος επαυξητικός αλγόριθμος εκμάθησης καλούμενος ως μείωση περικοπής σφάλματος (IREP) προτάθηκε από Furnkranz και Widmer.

Επαυξητική μείωση περικοπής σφάλματος (IREP)

Ο IREP αποδεικνύεται ανταγωνιστικός σε σχέση με τον REP όσον αφορά τα ποσοστά σφάλματος, αλλά και στη βελτίωσή της ταχύτητας στο στάδιο περικοπής. Στο IREP, τα κριτήρια στο στάδιο περικοπής εντοπίζονται σε έναν μονό κανόνα και όχι σε ένα σύνολο κανόνων.

Η διαδικασία IREP για έκδοση 2 κλάσης, εμφανίζεται στον αλγόριθμο 1. Όπως φαίνεται στον ψευδοκώδικα, ο IREP ενσωματώνει στενά τη διαδικασία περικοπής με τον κανόνα διαχώρισε κ κατέκτησε (π.χ. GrowRule) κάθε κανόνας απλοποιείται αμέσως μετά το στάδιο εκμάθησης και στιγμιαία καλύπτεται από τον κανόνα που θα διαγραφεί από το σύνολο κατάρτισης.

- «GrowRule η εφαρμογή GrowRule είναι μια προτασιακή έκδοση του FOIL. Αρχίζει με τις κενές περιπτώσεις κλίσεων και προσθέτει επανειλημμένα τους όρους που μεγιστοποιούν το κέρδος των πληροφοριών του FOIL, με κριτήριο μέχρι τον κανόνα να μην καλύπτει κανένα αρνητικό παράδειγμα από το αυξανόμενο σύνολο δεδομένων (GrowPos).

- PruneRule – Μετά τη μάθηση ενός κανόνα, ο κανόνας περικόπτετε αμέσως. Θεωρούμε μια διαγραφή της συχνότητας των περιπτώσεων του κανόνα ότι εξετάζεται και αυτός που μεγιστοποιεί τη λειτουργία επιλέγεται. Αυτή η διαδικασία επαναλαμβάνεται μέχρι καμία διαγραφή να μην βελτιώνει την αξία του ν.

$$v(Rule, \Pr unePos, \Pr uneNeg) = \frac{p + (N - n)}{p + N} \quad (3.7)$$

όπου το P (αντίστοιχα N) είναι ο συνολικός αριθμός του παραδείγματος σε PrunePos (PruneNeg) και το p (n) είναι ο αριθμός παραδείγματος σε PrunePos (PruneNeg) που καλύπτεται από τον κανόνα.

- «Διαχώρισε και Κατέκτησε - Μια στρατηγική εκμάθησης επικεντρώνεται στη δημιουργία ενός κανόνα τη φορά, κάθε ένας από τους οποίους καλύπτουν ένα μέρος

των παραδειγμάτων κατάρτισης. Τα παραδείγματα που καλύπτονται από τον τελευταίο κανόνα μάθησης αφαιρούνται από το σύνολο κατάρτισης (που χωρίζεται) πριν μαθευτούν οι κανόνες (πριν κατακτηθούν τα υπόλοιπα παραδείγματα κατάρτισης). Είναι διαφορετικό από τη στρατηγική διαίρει και βασίλευε, που χρησιμοποιείται από τους αλγορίθμους δέντρων απόφασης στο στάδιο κατασκευής δέντρων, όπως ID3, το KAPPO, C4.5. Σε στρατηγική διαίρει και βασίλευε, όλοι οι κανόνες ανακαλύπτονται μόνο μόλις τελειώσει ο αλγόριθμος ένα δέντρο απόφασης.

Επίσης, ο IREP υποστηρίζει το χάσιμο ιδιοτήτων, αριθμητικά χαρακτηριστικά γνωρίσματα και πολλαπλάσιες κλάσεις. Αυτό τον καθιστά εφαρμόσιμο σε ένα ευρύτερο φάσμα προβλημάτων συγκριτικής μέτρησης επιδόσεων (Cohen, 1995).

Αλγόριθμος 1- Ο αλγόριθμος IREP

procedure IREP(Pos, Neg)
begin
Ruleset := ∅
while Pos 6= ∅ **do**
grow and prune a new rule
split(Pos,Neg) into (GrowPos, GrowNeg) and (PrunePos, PruneNeg)
GrowRule - propositional version of FOIL
Rule = GrowRule(GrowPos, GrowNeg)
prune the rule immediately
Rule = PruneRule(Rule, PrunePos, PruneNeg)
if the error rate of Rule on (PrunePos, PruneNeg) exceeds 50% **then**
return Ruleset
else
add Rule to Ruleset
remove examples covered by Rule from (Pos, Neg)

endif
endwhile
return Ruleset
end

Επαναλαμβανόμενη επαυξητική περικοπή για να προκαλέσει τη μείωση σφάλματος (RIPPER)

Ο RIPPER προτάθηκε ως βελτιωμένη έκδοση IREP (αποκαλούμενου IREP*) ο οποίος βασίστηκε στην ελαχιστοποίηση του μήκος περιγραφής. Εισάγει επίσης τα πρόσθετα βήματα αφού παράγει τους αρχικούς κανόνες από τον IREP*, αποκαλούμενο ως «βελτιστοποίηση κανόνα», της επαναπερικοπής κάθε κανόνα στον κανόνα που τέθηκε ώστε να ελαχιστοποιήσει το σφάλμα του πλήρους συνόλου κανόνων. Δύο τροποποιήσεις σε IREP είναι :

- «Όταν ο IREP φάνηκε να είναι αδικαιολόγητα ευαίσθητος στο «μικρό διαζευκτικό πρόβλημα» ,που οφείλεται στις περιπτώσεις στάσεών στο στάδιο ανάπτυξη, η νέα περίπτωση στάσεων προτάθηκε μέσα [4] βασισμένη στο ελάχιστο μήκος περιγραφής (MDL). Αφότου κάθε κανόνας προστίθενται στο συνολικό μήκος περιγραφής D του παρόντος συνόλου κανόνων και τα παραδείγματα είναι υπολογισμένα. Το στάδιο αύξησης συνεχίζεται όταν υπάρχει ακόμα μη καλυμμένο θετικό παράδειγμα και μήκος $D_{new} < d$ (bits) + minD_i που έχει βρεθεί.

- «Ένα παράδειγμα μέσα [4] για το πρόβλημα του IREP στη μέτρηση του σταδίου περικοπής: η μέτρηση $\frac{p+(N-n)}{p+N}$ προτιμά έναν κανόνα R_1 που καλύπτει $p_1 = 2000$ θετικά παραδείγματα και $n_1 = 1000$ αρνητικά παραδείγματα σε έναν κανόνα R_2 που καλύπτει p1 = 1000 θετικά παραδείγματα και n1 = 1 αρνητικό παράδειγμα ακόμη και αν ο R2 είναι πιο προβλέψιμος από τονR_1R. Ως εκ τούτου, η μέτρηση

$v(Rule, \Pr unePos, \Pr uneNeg) = \frac{p + (N - n)}{p + N}$ του IREP αντικαθίσταται με $u^*(Rule, \Pr unePos, \Pr uneNeg) = \frac{p - n}{p + n}$ ο όποιος φαίνεται να έχει μια ικανοποιητικότερη συμπεριφορά.

Σύνολο κανόνων R_1, R_2. Το R_k που παράγεται από αυτήν την τροποποιημένη έκδοση του IREP (IREP*) μπορεί να βελτιστοποιηθεί ώστε να ελαχιστοποιήσει το σφάλμα ολόκληρου του συνόλου κανόνων που τίθεται στα στοιχεία που θα περικοπούν. Η διαδικασία βελτιστοποίησης εξετάζει κάθε κανόνα με τη σειρά με την οποία μαθεύτηκαν. Για κάθε κανόνα, δύο εναλλακτικοί κανόνες κατασκευάζονται : ο κανόνας αντικατάστασης R_i' και ο κανόνας αναθεώρησης.

Η απόφαση εάν η τελική υπόθεση πρέπει να περιλάβει τους αναθεωρημένους κανόνες ή τον αρχικό κανόνα λαμβάνεται χρησιμοποιώντας το MDL. Η βελτιστοποίηση μπορεί επίσης να επαναληφθεί με τη βελτιστοποίηση του κανόνα της καθορισμένης εξόδου από το RIPPER. Ως εκ τούτου, ο RIPPERk χρησιμοποιείται για τον αλγόριθμο τον οποίο βελτιστοποιεί επανειλημμένα τους χρόνους k (Cohen, 1995).

Αλγόριθμος 2- Ο αλγόριθμος RIPPER

procedure RIPPER(Pos, Neg)
begin
Ruleset := ∅
Ruleset := IREP*(Pos, Neg)
Repeat k times
Ruleset := MDLOptimize(RuleSet)
Pos* := positive examples not covered by RuleSet
RuleSet := RuleSet + IREP*(Pos*, Neg)
return Ruleset
end

Πειράματα

Λογισμικό ανάσυρσης δεδομένων – WEKA

Το Weka είναι μια συλλογή των αλγορίθμων μηχανών εκμάθησης για τις στοιχειώδεις εργασίες ανάσυρσης δεδομένων, γραμμένο σε Java. Οι αλγόριθμοι μπορούν είτε να εφαρμοστούν άμεσα σε ένα σύνολο δεδομένων είτε να κληθούν ως APIs από ένα άλλο πρόγραμμα Java. Κατά τρόπο ενδιαφέροντα, ο Weka παρέχει ένα ευρύ φάσμα εργαλείων από τα δεδομένα προεπεξεργασίας, ταξινόμησης (συμπεριλαμβανομένου JRip- μια εφαρμογή της Java του κανόνα μάθησης RIPPER), οπισθοδρόμησης, συγκέντρωση, κανόνες ένωσης, στην απεικόνιση για να ικανοποιήσει τις διάφορες ανάγκες των χρηστών. Το GUI υπάρχει σε 4 τρόπους, που ταιριάζουν με τις απλές καθώς επίσης και τις σύνθετες στοιχειώδεις εργασίες ανάλυσης

- «Κονσόλα
- «Εξερευνητής
- «Πειραματιστής
- «Ροη Γνώσης

Διανεμημένο ως λογισμικό ανοιχτού κώδικα με άδεια GNU, ο Weka επίσης ταιριάζει απόλυτα και στα νέα σχέδια μηχανών εκμάθησης.

3.1.3 Μέθοδοι μάθησης κατά περίπτωση

3.1.3.1 Κ-κοντινότερος Γείτονας

Οι ταξινομητές κοντινότερων γειτόνων βασίζονται στη μάθηση με βάση την αναλογία.Ν-διάστατα αριθμητικά χαρακτηριστικά περιγράφουν εκπαιδευμένα δείγματα. Στο ν-διάστατο χώρο κάθε σημείο αντιπροσωπεύει ένα δείγμα.

Με την εισαγωγή μιας νέας εγγραφής ο ταξινομητής ψάχνει στο χώρο προτύπων για κ εγγραφές που βρίσκονται πιο κοντά στην άγνωστη εγγραφή. Αυτές οι κ εκπαιδευόμενες εγγραφές είναι οι κ-κοντινότεροι γείτονες της νέας-άγνωστης εγγραφής. Η κοντινότερη απόσταση προσδιορίζετε σε σχέση με την ευκλείδεια απόσταση μεταξύ δυο σημείων $X=(c_1, c_2,\ldots, c_n)$ και $\Psi=(y_1,y_2,\ldots y_n)$ ορίζετε ως :

$$d(X,\Psi)=\sqrt{[\Sigma^n(c_i-y_i)^2]} \quad (3.8)$$

Στα αριθμητικά δεδομένα ο υπολογισμός της απόστασης είναι απλός , ενώ μεγάλο πρόβλημα ανακύπτει με τις ποιοτικές μεταβλητές π.χ. ορισμός της απόστασης ανάμεσα σε δυο χρώματα. Οι ταξινομητές αυτού του είδους αποθηκεύουν όλες τις εκπαιδευμένες εγγραφές ενώ δε φτιάχνουν ταξινομητή έως ότου η καινούργια εγγραφή απαιτείτε να ταξινομηθεί (lazy learners) και λειτουργεί αντίθετα με τα δέντρα απόφασης και τους αλγόριθμους αναστροφής μετάδοσης λάθους (eager learners). Το βασικότερο πλεονέκτημα της μεθόδου είναι ότι οι lazy learners είναι γρηγορότεροι στην εκπαίδευση από τους eager learners. Μειονεκτήματα της μεθόδου αυτής είναι το πιθανά μεγάλο υπολογιστικό κόστος αν ο αριθμός των γειτόνων είναι αρκετά μεγάλος, το ότι αν και αρκετά γρήγορη στην εκπαίδευση είναι πιο αργοί στην ταξινόμηση (Βαζιργιάννης & Χαλκίδη, 2003).

3.1.3.2 Μηχανές διανυσμάτων υποστήριξης

Στη μηχανή μάθησης, ένας ταξινομητής margin είναι σε θέση να δώσει μια σχετική απόσταση από το όριο απόφασης για κάθε παράδειγμα. Για παράδειγμα, εάν ένας γραμμικός ταξινομητής (π.χ. perceptron) χρησιμοποιείται, η απόσταση ενός παραδείγματος από τα διαχωρισμένα υπερεπίπεδα είναι το περιθώριο(margin) εκείνου του παραδείγματος.

Η έννοια του περιθωρίου είναι σημαντική σε διάφορους αλγορίθμους ταξινόμησης εκμάθησης μηχανών, δεδομένου ότι μπορεί να χρησιμοποιηθεί για να οριοθετήσει το σφάλμα γενίκευσης του ταξινομητή. Αυτά τα όρια εμφανίζονται συχνά χρησιμοποιώντας τη VC διάσταση. Από την ιδιαίτερη προεξοχή είναι το σφάλμα γενίκευσης που δεσμεύεται στην ώθηση των αλγορίθμων και των διανυσματικών μηχανών υποστήριξης (Wu et al., 2008).

3.1.3.3 Ώθηση καθορισμού του περιθωρίου (margin)

Το περιθώριο για έναν επαναληπτικό αλγόριθμο ώθησης, δεδομένου ενός συνόλου παραδειγμάτων με δύο κλάσεις μπορεί να καθοριστεί ως εξής. Στον ταξινομητή δίνεται ένα ζευγάρι παραδείγματος (x,y) πού $c \in X$ είναι ένα διάστημα δικτυακών γειτονιών και $y \in Y = \{-1,+1\}$ είναι η ετικέτα του παραδείγματος. Ο επαναληπτικός αλγόριθμος ώθησης επιλέγει έναν ταξινομητή $h_j \in C$ σε κάθε επανάληψη j όπου το C είναι ένα διάστημα των πιθανών ταξινομητών που προβλέπουν τις πραγματικές τιμές.

Αυτή η υπόθεση είναι σταθμισμένη $a_j \in R$ όπως επιλέγεται από τον αλγόριθμο ώθησης. Στην επανάληψη t, το περιθώριο ενός παραδείγματος X μπορεί έτσι να οριστεί ως:

$$\frac{y\sum_j^t a_j h_j(x)}{\Sigma|a_j|} \quad (3.9)$$

Εξ αυτού του ορισμού, το περιθώριο είναι θετικό εάν το παράδειγμα ονομάζεται σωστά και αρνητικό εαν το παράδειγμα ονομάζεται ανακριβώς.

Αυτός ο καθορισμός μπορεί να τροποποιηθεί και δεν είναι ο μόνος τρόπος για να καθοριστεί το περιθώριο για τους αλγορίθμους.

3.1.3.4 Παραδείγματα αλγορίθμων βασισμένα σε περιθώριο

Πολλοί ταξινομητές μπορούν να δώσουν ένα σχετικό περιθώριο για κάθε παράδειγμα. Εντούτοις, μόνο μερικοί ταξινομητές χρησιμοποιούν τις πληροφορίες του περιθωρίου μαθαίνοντας από ένα σύνολο στοιχείων.

Πολλοί αλγόριθμοι ώθησης στηρίζονται στην έννοια ενός περιθωρίου για να δώσουν βάρος στα παραδείγματα. Εάν μια κυρτή απώλεια χρησιμοποιείται ,ένα παράδειγμα με το υψηλότερο περιθώριο θα λάβει το λιγότερο βάρος από ένα παράδειγμα με το χαμηλότερο περιθώριο. Αυτό οδηγεί τον αλγόριθμο ώθησης για να στρέψει το βάρος στα χαμηλού περιθωρίου παραδείγματα. Στους nonconvex αλγορίθμους (π.χ. BrownBoost), το περιθώριο υπαγορεύει ακόμα τη στάθμιση ενός παραδείγματος, αν και η στάθμιση είναι μη μονοτονική όσον αφορά το περιθώριο. Υπάρχουν οι αλγόριθμοι ώθησης που μεγιστοποιούν ευαπόδεικτα το ελάχιστο περιθώριο. Οι διανυσματικές μηχανές υποστήριξης μεγιστοποιούν ευαπόδεικτα το περιθώριο χωρίζοντας υπερεπίπεδα. Οι διανυσματικές μηχανές υποστήριξης που εκπαιδεύονται χρησιμοποιώντας τα θορυβώδη στοιχεία (δεν υπάρχει κανένας τέλειος χωρισμός των στοιχείων στο δεδομένο διάστημα) μεγιστοποιούν το μαλακό περιθώριο. Περισσότερη συζήτηση αυτού μπορεί να βρεθεί στο διανυσματικό άρθρο μηχανών υποστήριξης.
Ο αλγόριθμος ψηφίζω-συνειδητά είναι ένα περιθώριο που μεγιστοποιεί τον αλγόριθμο βασισμένο σε μια επαναληπτική εφαρμογή του κλασικού συνειδητού αλγορίθμου (Wu et al., 2008).

3.1.4 ADABOOST

Το AdaBoost είναι μια από τις δημοφιλέστερες μεθόδους ταξινόμησης. Σε αντίθεση με άλλες μεθόδους συνόλων το AdaBoost είναι εγγενώς διαδοχικό. Σε πολλές εντατικές πραγματικές καταστάσεις στοιχείων αυτό μπορεί να περιορίσει την πρακτική δυνατότητα εφαρμογής της μεθόδου. Η P-AdaBoost είναι ένα νέο σχέδιο για

τον παραλληλισμό AdaBoost, ο οποίος χτίζει επάνω στα προηγούμενα αποτελέσματα σχετικά με τη δυναμική των βαρών AdaBoost. Η P-AdaBoost παράγει τις προσεγγίσεις στα πρότυπα μοντέλα AdaBoost που μπορούν εύκολα και αποτελεσματικά να κατανείμουν ένα δίκτυο κόμβων υπολογισμού. Τα πειράματα αναφέρονται και στα σύνολα συνθετικών και στοιχείων συγκριτικής μέτρησης επιδόσεων.

Διάφορα σημαντικά ζητήματα που περιλαμβάνονται στην εκμάθηση adaboost παραμένουν ακόμα ανοικτά . Ο σημαντικότερος αλγόριθμος είναι ο νέος προσανατολισμος προς την τοπολογία, ο αλγόριθμος Adaboost (TOBoost). Το TOBoost ελαχιστοποιεί αμέσως το σφάλμα ταξινόμησης κάθε επιλεγμένου χαρακτηριστικού γνωρίσματος, και επιτρέπει έτσι στον τελικό ανιχνευτή για να είναι πιό χαρακτηριστικό και για να συγκλίνει γρηγορότερα. Επιπλέον, ένα απλό σχέδιο απότομου πεσίματος παρουσιάζεται για το συντονισμό των παραμέτρων καταρρακτών TOBoost και η εκτίμηση Γκάους πυκνότητας πυρήνων εισάγεται για να ενισχύσει τη δυνατότητα γενίκευσης του TOBoost.

Μια άλλη σημαντική συμβολή είναι η διαμόρφωση τοπολογίας των ομοειδών χαρακτηριστικών γνωρισμάτων (HL), η οποία αποκαλύπτει ότι μια ενδιαφέρουσα ιδιοκτησία των αρνητικών HL χαρακτηρίζει και αποφεύγει σημαντικά τους περιττούς υπολογισμούς κατάρτισης. Οι ανωτέρω αυξήσεις οδηγούν σε έναν αποδοτικότερο και σταθερό ανιχνευτή με λιγότερα χαρακτηριστικά γνωρίσματα. Τα εκτενή πειράματα στην εφαρμογή της ανίχνευσης ίριδων πραγματοποιούνται και η ενθάρρυνση της απόδοσης επιτυγχάνεται (Freund & Shapire, 1999).

3.1.5 EM

Ο αλγόριθμος της EM είναι ένας δημοφιλής και χρήσιμος αλγόριθμος για τον εκτιμητή μέγιστης πιθανότητας στα ελλιπή προβλήματα στοιχείων. Κάθε επανάληψη του αλγορίθμου αποτελείται από δύο απλά βήματα: ένα E-βήμα, στο οποίο

υπολογίζεται μια υπό όρους προσδοκία , και ένα Μ-βήμα, όπου η προσδοκία μεγιστοποιείται. Σε μερικά προβλήματα, εντούτοις, ο αλγόριθμος της ΕΜ δεν μπορεί να εφαρμοστεί δεδομένου ότι η υπό όρους προσδοκία που απαιτείται στο Ε-βήμα δεν μπορεί να υπολογιστεί. Αντ' αυτού η προσδοκία μπορεί να υπολογιστεί από την προσομοίωση. Καλούμε αυτό μιμούμενο αλγόριθμο της ΕΜ. Οι προσομοιώσεις μπορούν, τουλάχιστον σε γενικές γραμμές, να γίνουν με δύο τρόπους. Είτε οι νέες ανεξάρτητες τυχαίες μεταβλητές σύρονται σε κάθε επανάληψη, ή οι ίδιες στολές επαναχρησιμοποιούνται σε κάθε επανάληψη.

Είναι γνωστό ότι ο αλγόριθμος της ΕΜ συγκλίνει γενικά σε μια τοπική εκτίμηση μέγιστης πιθανότητας. Εντούτοις, έχουν υπάρξει πολλά στοιχεία για να δειχτεί ότι ο αλγόριθμος της ΕΜ μπορεί να συγκλίνει σωστά στις αληθινές παραμέτρους εφ' όσον η επικάλυψη Gaussians στα στοιχεία δειγμάτων είναι αρκετά μικρή. Έχει αποδειχθεί ότι ο αλγόριθμος της ΕΜ γίνεται μια χαρτογράφηση συστολής των παραμέτρων μέσα σε μια γειτονιά της συνεπούς λύσης της μέγιστης πιθανότητας όταν το μέτρο της μέσης επικάλυψης μεταξύ Gaussians στο αρχικό μίγμα είναι αρκετά μικρό και ο αριθμός δειγμάτων είναι αρκετά μεγάλος.

Δηλαδή εάν οι αρχικές παράμετροι τίθενται μέσα στη γειτονιά, ο αλγόριθμος της ΕΜ θα συγκλίνει πάντα στη συνεπή λύση, δηλ., το αναμενόμενο αποτέλεσμα. Επιπλέον, τα αποτελέσματα προσομοίωσης περαιτέρω καταδεικνύουν ότι αυτή η σωστή γειτονιά σύγκλισης γίνεται μεγαλύτερη καθώς η μέση επικάλυψη γίνεται μικρότερη (Lange, 2003).

ΚΕΦΑΛΑΙΟ 4

4.1 Εφαρμογές ταξινόμησης

Ο τομέας εξόρυξης δεδομένων αναπτύχθηκε από την ανάγκη ανεύρεσης χρήσιμων πληροφοριών μέσα στα διαρκώς αυξανόμενα δεδομένα αποθηκών σε εμπορικές επιχειρήσεις και έχει τις ρίζες του σε καθιερωμένους κλάδους όπως η στατιστική, η τεχνητή νοημοσύνη, η μηχανική μάθηση και η αναγνώριση προτύπων. Οι ολοένα αυξανόμενες απαιτήσεις από το μέγεθος των επιχειρησιακών εφαρμογών απαιτούσαν τη χρήση τεχνολογιών βάσεων δεδομένων και υπολογιστές υψηλής απόδοσης στις εφαρμογές εξόρυξης δεδομένων.

Δεδομένης της επιτυχίας της εξόρυξης δεδομένων στους εμπορικούς τομείς, δεν χρειάστηκε μεγάλο διάστημα για τους επιστήμονες και τους μηχανικούς να συνειδητοποιήσουν τη χρησιμότητα των τεχνικών εξόρυξης δεδομένων στους επιστημονικούς κλάδους. Για παράδειγμα, η ανάλυση μεγάλου όγκου βάσεων δεδομένων προσομοίωσης που παράγονται από την υπολογιστική προσομοίωση σωματικών και μηχανικών συστημάτων είναι δύσκολη και χρονοβόρα χρησιμοποιώντας τις παραδοσιακές μεθόδους. Η διαθεσιμότητα των κατάλληλων τεχνικών εξόρυξης δεδομένων επιτρέπει σε μηχανικούς και επιστήμονες να αναλύσουν ανάλογα δεδομένα και να αποκτήσουν θεμελιώδεις γνώσεις στους υποκείμενους μηχανισμούς που εμπλέκονται στις εκάστοτε διαδικασίες υπό διερεύνηση.

Παραδείγματα ογκωδών και σύνθετων βάσεων δεδομένων παρουσιάζονται συχνά σε τομείς όπως η αστρονομία, η ιατρική απεικόνιση, η χημεία και η βιοπληροφορική, οι τηλεπικοινωνίες, το τραπεζικό σύστημα, τα τμήματα εξυπηρέτησης πελατών κ.α. Από τη στιγμή που ο ρυθμός παραγωγής αυτών των δεδομένων υπερβαίνει κατά πολύ τη δυνατότητά μας να τα αναλύσουμε, έχει δημιουργηθεί ένα αυξανόμενο ενδιαφέρον σε

διαφορετικές επιστημονικές κοινότητες για την εκμετάλλευση τεχνικών εξόρυξης δεδομένων που θα βοηθήσουν στο δύσκολο έργο της ανάλυσης αυτών των μεγάλων βάσεων δεδομένων.

Οι τεχνικές ταξινόμησης και εξόρυξης δεδομένων έχουν διεισδύσει σε εξειδικευμένες επιστημονικές εφαρμογές. Για παράδειγμα, ο προσδιορισμός της δομής σύνθετων οργανικών μορίων, ο συσχετισμός της δομής και της λειτουργίας του μορίου, αλλά και η κατασκευή-δημιουργία ενός μορίου με συγκεκριμένη λειτουργία της επιλογής μας αποτελούν σημαντικές προκλήσεις της βιοπληροφορικής. Το μέγεθος και η πολυπλοκότητα των μορίων καθιστά τις παραπάνω διαδικασίες εξαιρετικά δύσκολες και εντατικές σε υπολογισμούς. Οι τεχνικές που έχουν αναπτυχθεί για να συνδράμουν σε αυτό τον σκοπό καλύπτουν τους τομείς της μηχανικής μάθησης (ομαδοποίηση, ταξινόμηση), αλγόριθμους (συστοιχίες, ταίριασμα προτύπων) και της στατιστικής (μάθηση Bayesian, προσαρμογή μοντέλων). Απώτερος στόχος αυτών τα των προσπαθειών είναι η καλύτερη κατανόηση των υποκείμενων βιολογικών, μοριακών και βιοχημικών διαδικασιών της λειτουργίας των οργανισμών (Grossman et al., 2001).

Η ιατρική έχει επίσης εκμεταλλευτεί τη χρησιμότητα των τεχνικών της εξόρυξης και ταξινόμησης δεδομένων. Η πλειονότητα των ερευνών στον τομέα της γονιδιακής ιατρικής επιδιώκει την ανάλυση δεδομένων που αφορούν την έκφραση των γονιδίων σε αλληλουχίες του DNA, αποτελούμενο από χιλιάδες γονίδια για κάθε ασθενή, με απώτερο σκοπό τη διάγνωση (υπο)τύπων ασθενειών και την πρόγνωση της ασθένειας που δύναται να οδηγήσει σε εξατομικευμένες θεραπευτικές προτάσεις. Τα ερευνητικά άρθρα σχετίζονται κυρίως με την ογκολογία, όπου υπάρχει έντονη ανάγκη για τον καθορισμό εξατομικευμένων στρατηγικών θεραπείας (Bellazzi & Zupan, 2008).

Απλές εφαρμογές της ταξινόμησης και άλλων τεχνικών εξόρυξης δεδομένων συναντώνται σε πολλές δραστηριότητες της καθημερινότητας. Οι τράπεζες χρησιμοποιούν τεχνικές εξόρυξης δεδομένων όπως η ταξινόμηση για να διερευνήσουν τη ροή των χρημάτων, τη διασπορά επενδύσεων και κεφαλαίων, τη δανειοδότηση και

άλλες διαδικασίες. Οι αστυνομικές υπηρεσίες κάνουν χρήση αντίστοιχων τεχνικών για τον εντοπισμό θέσης κινητών τηλεφώνων και την υποκλοπή συνομιλιών, ενώ εταιρείες πληροφορικής χρησιμοποιούν τεχνικές ταξινόμησης δεδομένων για την εξέταση κλήσεων του τμήματος εξυπηρέτησης πελατών αποσκοπώντας στη διαμόρφωση προτύπων παραπόνων και τη βελτίωση των υπηρεσιών τους.

4.2 Περιγραφή Βάσεων

Οι βάσεις που θα χρησιμοποιήσουμε είναι οι ακόλουθες:

1) Splice Junction
2) Car
3) Tic Tac Toe

Splice

Η βάση δεδομένων με όνομα Primate splice-junction gene sequences (DNA) with associated imperfect domain theory δημιουργήθηκε την 1/1/1992 με παραδείγματα από την Genbank 64.1 με κατηγορίες “ei” και “ie” που περιλαμβάνουν κάθε διαιρεμένο γονίδιο για πρωτεύοντα. Μη συνδυασμένα παραδείγματα χρησιμοποιήθηκαν σε αυτή τη βάση. Έχει χρησιμοποιηθεί α) σε έρευνες μέσω της μηχανικής μάθησης και νευρωνικών δικτύων για αναγνώριση ακολουθιών DNA, β) σε πρόβλεψη χαρακτηριστικών νουκλεοτιδίων του DNA και γ) σε αποτελέσματα που υποδεικνύουν ότι η μηχανική μάθηση (νευρωνικά δίκτυα, κοντινότερος γείτονας) αποδίδει καλύτερα από τις μεθόδους που βασίζονται στην ταξινόμηση.

Η εν λόγω βάση δεδομένων έχει αναπτυχθεί για να βοηθήσει στην εκτίμηση του «υβριδικού» αλγόριθμου KBANN. Περιέχει 3190 παραδείγματα (instances) και 62 χαρακτηριστικά (attributes).

Car Evaluation Database

Η βάση δεδομένων με όνομα Car Evaluation Database δημιουργήθηκε από τον Marco Bohanec τον Ιούνιο του 1997. Πρωτοεμφανίστηκε στην εργασία των Bohanec και Rajkovic "Knowledge acquisition and explanation for multi-attribute decision making" (1988). Η βάση δεδομένων περιέχει 1278 στοιχεία και 6 χαρακτηριστικά.
Το μοντέλο εκτιμά τα αυτοκίνητα σύμφωνα με την παρακάτω δομή:

Αυτοκίνητο	αποδοχή αυτοκινήτου
Τιμή	συνολική τιμή
Αγορά	τιμή αγοράς
διατήρηση/service	τιμή συντήρησης
Τεχνολογία	τεχνικά χαρακτηριστικά
Άνεση	άνεση
πόρτες	αριθμός θυρών
άτομα	άτομα που χωρά
χωρητικότητα αποσκευών	μέγεθος αποσκευών που μεταφέρει
ασφάλεια	εκτιμώμενη ασφάλεια αυτοκινήτου

Tic Tac Toe

Η βάση δεδομένων με όνομα TIC TAC TOE endgame database δημιουργήθηκε από τον David W. Aha στις 19/8/1991. Αυτή η βάση δεδομένων κωδικοποιεί ολοκληρωμένα σύνολα πιθανών συνθέσεων πίνακα στο τέλος του tic-tac-toe game, όπου "x" υποθέτουμε ότι έχει παιχθεί πρώτο. Ο κύριος στόχος είναι "η νίκη για το x" (π.χ. αληθεύει όταν το x παίρνει μία από τις οκτώ πιθανές μορφές για να δημιουργήσει ένα "δένδρο στη σειρά"). Είναι ενδιαφέρον ότι η συγκεκριμένη βάση δεδομένων αποδίδει έναν αλγόριθμο δένδρου αποφάσεων που μοιάζει με τον ID3. Παρόλα αυτά, ο βασισμένος σε κανόνες CN2 αλγόριθμος, αλλά και άλλοι όπως ο CITRE ή ο IB1 ανταπεξέρχονται καλά σε αυτή τη βάση δεδομένων. Περιλαμβάνει 958 περιπτώσεις (instances) με 9 χαρακτηριστικά (attributes) και καμία μεταβλητή δεν απουσιάζει.

4.3 Μεθοδος αξιολόγησης ταξινόμησης

Σε όσες βάσεις υπάρχει ο αριθμός 10 θα γίνουν 10 τρεξίματα. Αυτό ονομάζεται 10-fold-cross validation. Το περιγράφουμε παρακάτω:10-fold-cross validation: Η μέθοδος ελέγχου αξιοπιστίας (validation test) που χρησιμοποιήθηκε για τις βάσεις δεδομένων Promoter recognition, Tic, Tac Toe Endgame Car Evaluation και Wisconsin Breast Cancer, είναι αυτή του ελέγχου αξιοπιστίας με διασταύρωση, με την χρήση 10 συνόλων (ten-fold cross-validation) (Stone, 1974). Με βάση αυτή την μέθοδο, το σύνολο εκπαίδευσης Y, υπακούοντας στην αντιμεταθετική ιδιότητα, χωρίζεται σε 10 τυχαία, ισότιμα μεταξύ τους σύνολα.

Ο έλεγχος αξιοπιστίας με διασταύρωση περιλαμβάνει ισάριθμες με τα παραπάνω σύνολα, φάσεις αξιολόγησης. Σε κάθε μια από αυτές τις δέκα φάσεις, ένα από τα σύνολα που δεν έχει μέχρι στιγμής εμφανιστεί στην διαδικασία μάθησης, χρησιμοποιείται για έλεγχο (σύνολο ελέγχου) ενώ τα υπόλοιπα εννέα σύνολα χρησιμοποιούνται σαν ένα εννιαίο σύνολο εκπαίδευσης για την εφαρμογή των αλγορίθμων ταξινόμησης και την εξαγωγή των κανόνων απόφασης. Αξίζει να σημειωθεί ότι οι αναλογίες των κλάσεων σε κάθε ένα από τα δέκα σύνολα ελέγχου είναι παρόμοιες, αν όχι ίδιες στην καλύτερη των περιπτώσεων, με τις αναλογίες των κλάσεων στο αρχικό σύνολο εκπαίδευσης Y.

Βάσει των παραπάνω, είναι προφανές ότι τόσο το σύνολο των κανόνων απόφασης X όσο και ο αριθμός των κανόνων d είναι διαφορετικά για κάθε μία από τις δέκα φάσεις αξιολόγησης.

Οι υπόλοιπες βάσεις είναι χωρισμένες εξ'ορισμού σε ένα σύνολο εκπαίδευσης και ένα σύνολο ελέγχου. Έτσι, στην περίπτωση αυτή, το μόνο που χρειάζεται είναι να εξαχθούν οι κανόνες απόφασης μέσα από την διαδικασία εκπαίδευσης των αλγορίθμων ταξινόμησης επί του συνόλου εκπαίδευσης, και στην συνέχεια να εφαρμοστούν οι κανόνες αυτοί μέσω της μεθόδου στο σύνολο ελέγχου.

4.4 Πινάκες αποτελεσμάτων

Στους παρακάτω πίνακες παρουσιάζετε η ακρίβεια ταξινόμισης για την εκάστοτε βάση δεδομένων αναλυτικά για κάθε τρέξιμο αλλά και συνολικά για όλες τις περιπτώσεις του diogenis.

Βάση δεδομένων Car

Στον ακόλουθο πίνακα 4.1, φαίνονται τα αποτελέσματα της βάσης Car για κάθε ένα από τα δέκα σύνολα στα οποία χωρίζεται η βάση, ακολουθώντας την διαδικασία τρεξιμάτων 10-fold cross validation, για κάθε μια από τις κλάσεις Unacceptable (Unacc.), Acceptable (Acc.), Good και Very Good. Επίσης στην τελευταία γραμμή του πίνακα φαίνεται η συνολική ακρίβεια ταξινόμησης για κάθε ένα από τα παραπάνω σύνολα. Η συνολική ακρίβεια ταξινόμησης για όλη την βάση είναι 81%.

Πίνακας Car C4.5 Ratio

	Sum. 1	Sum. 2	Sum. 3	Sum. 4	Sum. 5	Sum. 6	Sum. 7	Sum. 8	Sum. 9	Sum.10
Unacc.	100 %	92%	90%	70%	95%	89%	88%	99%	96%	98%
Acc.	5%	36%	44%	100 %	73%	78%	92%	64%	64%	30%
Good	57%	57%	42%	50%	57%	57%	42%	57%	42%	85%
Very Good	71%	57%	42%	71%	57%	66%	50%	66%	83%	50%
Σύνολο	76%	77%	75%	75%	86%	83%	85%	87%	86%	80%

Πίνακας 4.1

Στον ακόλουθο πίνακα 4.2, φαίνονται τα αποτελέσματα της βάσης Car για κάθε ένα από τα δέκα σύνολα στα οποία χωρίζεται η βάση, ακολουθώντας την διαδικασία τρεξιμάτων 10-fold cross validation, για κάθε μια από τις κλάσεις Unacceptable (Unacc.), Acceptable (Acc.), Good και Very Good. Επίσης στην τελευταία γραμμή του πίνακα φαίνεται η συνολική ακρίβεια ταξινόμησης για κάθε ένα από τα παραπάνω σύνολα. Η συνολική ακρίβεια ταξινόμησης για όλη την βάση είναι 80%.

Πίνακας Car C4.5 Ratio Prck 1

	Sum. 1	Sum. 2	Sum. 3	Sum. 4	Sum. 5	Sum. 6	Sum. 7	Sum. 8	Sum. 9	Sum.10
Unacc.	100 %	90%	84%	69%	84%	89%	86%	98%	96%	98%
Acc.	5%	36%	39%	100 %	78%	86%	92%	64%	56%	35%
Good	57%	85%	57%	50%	42%	71%	71%	57%	57%	85%
Very Good	42%	57%	42%	85%	57%	83%	50%	83%	83%	50%
Σύνολο	74%	77%	71%	75%	79%	86%	85%	87%	85%	81%

Πίνακας 4.2

Στον ακόλουθο πίνακα 4.3, φαίνονται τα αποτελέσματα της βάσης Car για κάθε ένα από τα δέκα σύνολα στα οποία χωρίζεται η βάση, ακολουθώντας την διαδικασία τρεξιμάτων 10-fold cross validation, για κάθε μια από τις κλάσεις Unacceptable (Unacc.), Acceptable (Acc.), Good και Very Good. Επίσης στην τελευταία γραμμή του πίνακα φαίνεται η συνολική ακρίβεια ταξινόμησης για κάθε ένα από τα παραπάνω σύνολα. Η συνολική ακρίβεια ταξινόμησης για όλη την βάση είναι 61,8%.

Πίνακας Car CN2

	Sum. 1	Sum. 2	Sum. 3	Sum. 4	Sum. 5	Sum. 6	Sum. 7	Sum. 8	Sum. 9	Sum.10
Unacc.	71%	99%	100 %	77%	100 %	95%	88%	11%	93%	0%
Acc.	89%	10%	100 %	69%	0%	0%	15%	0%	15%	0%
Good	0%	0%	0%	66%	100 %	100 %	100 %	100 %	0%	0%
Very Good	100 %	85%	0%	28%	100 %	100 %	100 %	83%	0%	100%
Σύνολο	73%	74%	91%	72%	77%	74%	72%	14%	68%	3%

Πίνακας 4.3

Βάση δεδομένων Tic Tac Toe

Στον ακόλουθο πίνακα 4.4, φαίνονται τα αποτελέσματα της βάσης Tic Tac Toe για κάθε ένα από τα δέκα σύνολα στα οποία χωρίζεται η βάση, ακολουθώντας την διαδικασία τρεξιμάτων 10-fold cross validation, για κάθε μια από τις κλάσεις Ναι και Οχι. Επίσης στην τελευταία γραμμή του πίνακα φαίνεται η συνολική ακρίβεια ταξινόμησης για κάθε ένα από τα παραπάνω σύνολα. Η συνολική ακρίβεια ταξινόμησης για όλη την βάση είναι 73,2%.

Πίνακας Tic Tac Toe C4.5 Ratio

	Sum. 1	Sum. 2	Sum. 3	Sum. 4	Sum. 5	Sum. 6	Sum. 7	Sum. 8	Sum. 9	Sum.10
Ναι	62%	83%	48%	83%	88%	69%	77%	85%	93%	88%
Οχι	79%	58%	42%	72%	51%	69%	57%	72%	78%	63%
Σύνολο	68%	75%	45%	79%	76%	69%	70%	81%	88%	80%

Πίνακας 4.4

Στον ακόλουθο πίνακα 4.5, φαίνονται τα αποτελέσματα της βάσης Tic Tac Toe για κάθε ένα από τα δέκα σύνολα στα οποία χωρίζεται η βάση, ακολουθώντας την διαδικασία τρεξιμάτων 10-fold cross validation, για κάθε μια από τις κλάσεις Ναι και Οχι. Επίσης στην τελευταία γραμμή του πίνακα φαίνεται η συνολική ακρίβεια ταξινόμησης για κάθε ένα από τα παραπάνω σύνολα. Η συνολική ακρίβεια ταξινόμησης για όλη την βάση είναι 73,2%.

Πίνακας Tic Tac Toe C4.5 Ratio Prck

	Sum. 1	Sum. 2	Sum. 3	Sum. 4	Sum. 5	Sum. 6	Sum. 7	Sum. 8	Sum. 9	Sum.10
Ναι	62%	85%	50%	87%	87%	73%	82%	92%	98%	88%
Οχι	79%	95%	42%	63%	48%	69%	51%	72%	75%	48%
Σύνολο	68%	75%	46%	78%	73%	71%	71%	85%	90%	75%

Πίνακας 4.5

Στον ακόλουθο πίνακα 4.6, φαίνονται τα αποτελέσματα της βάσης Tic Tac Toe για κάθε ένα από τα δέκα σύνολα στα οποία χωρίζεται η βάση, ακολουθώντας την διαδικασία τρεξιμάτων 10-fold cross validation, για κάθε μια από τις κλάσεις Ναι και Οχι. Επίσης στην τελευταία γραμμή του πίνακα φαίνεται η συνολική ακρίβεια ταξινόμησης για κάθε ένα από τα παραπάνω σύνολα. Η συνολική ακρίβεια ταξινόμησης για όλη την βάση είναι 64,9%.

Πίνακας Tic Tac Toe CN2

	Sum. 1	Sum. 2	Sum. 3	Sum. 4	Sum. 5	Sum. 6	Sum. 7	Sum. 8	Sum. 9	Sum.10
Ναι	88%	72%	64%	67%	88%	71%	88%	100 %	55%	39%
Οχι	47%	47%	30%	33%	45%	45%	33%	18%	81%	18%
Σύνολο	73%	63%	52%	55%	73%	62%	69%	71%	64%	67%

Πίνακας 4.6

Βάση δεδομένων Splice Junction

Στον ακόλουθο πίνακα 4.7, φαίνονται τα αποτελέσματα της βάσης Splice Junction για κάθε ένα από τα δέκα σύνολα στα οποία χωρίζεται η βάση, ακολουθώντας την διαδικασία τρεξιμάτων 10-fold cross validation, για κάθε μια από τις κλάσεις Ei, Ie γονιδίων και κανένα από τα δύο. Επίσης στην τελευταία γραμμή του πίνακα φαίνεται η συνολική ακρίβεια ταξινόμησης για κάθε ένα από τα παραπάνω σύνολα. Η συνολική ακρίβεια ταξινόμησης για όλη την βάση είναι 89,7%.

Πίνακας Splice Junction C4.5 Ratio

	Sum. 1	Sum. 2	Sum. 3	Sum. 4	Sum. 5	Sum. 6	Sum. 7	Sum. 8	Sum. 9	Sum.10
Ei	94%	72%	90%	85%	98%	92%	57%	93%	85%	93%
Ie	90%	90%	88%	74%	81%	84%	76%	74%	76%	92%
Κανένα από τα δύο	95%	91%	94%	93%	96%	95%	96%	90%	90%	94%
Σύνολο	94%	86%	92%	86%	93%	91%	89%	87%	86%	93%

Πίνακας 4.7

Στον ακόλουθο πίνακα 4.8, φαίνονται τα αποτελέσματα της βάσης Splice Junction για κάθε ένα από τα δέκα σύνολα στα οποία χωρίζεται η βάση, ακολουθώντας την διαδικασία τρεξιμάτων 10-fold cross validation, για κάθε μια από τις κλάσεις Ei, Ie γονιδίων και κανένα από τα δύο. Επίσης στην τελευταία γραμμή του πίνακα φαίνεται η συνολική ακρίβεια ταξινόμησης για κάθε ένα από τα παραπάνω σύνολα. Η συνολική ακρίβεια ταξινόμησης για όλη την βάση είναι 92,8%.

Πίνακας Splice Junction C4.5 Ratio Prck

	Sum. 1	Sum. 2	Sum. 3	Sum. 4	Sum. 5	Sum. 6	Sum. 7	Sum. 8	Sum. 9	Sum.10
Ei	100 %	80%	96%	90%	100 %	93%	88%	98%	90%	96%
Ie	92%	96%	92%	83%	89%	83%	92%	85%	88%	97%
Κανένα από τα δύο	94%	94%	97%	92%	96%	95%	93%	95%	90%	95%
Σύνολο	95%	91%	95%	89%	95%	92%	92%	93%	90%	96%

Πίνακας 4.8

4.5 Αξιολόγηση αποτελεσμάτων

Το πρόγραμμα Διογένης δημιουργήθηκε από το εργαστήριο Πληροφορικής και Τεχνητής Νοημοσύνης του Πανεπιστημίου Πατρών το 2001. Το περιβάλλον εργασίας είναι απλό, αλλά σχετικά δύσκολο για χρήστες με μικρή εξοικείωση. Προσφέρει αρκετά εργαλεία ταξινόμησης, αλλά περιορίζεται στους αλγόριθμους CN2 και C4.5. Θα μπορούσαμε να πούμε ότι είναι ένα καλό και αξιόπιστο πρόγραμμα ταξινόμησης δεδομένων.

Όπως φαίνεται από τα τρεξίματα των βάσεων (Βλέπε Πίνακες 4.1 – 4.9) το πρόγραμμα Διογένης έδωσε αποτελέσματα για τις περισσότερες εξ αυτών εκτός από τη βάση Congressional voting για τον CN2 αλγόριθμο, τη βάση Audiology και για

τους δύο αλγόριθμους, τη βάση tic-tac-toe για την ακρίβεια ταξινόμησης (classified accuracy). Για όλα τα υπόλοιπα έδωσε αποτελέσματα στα απλά τρεξίματα αλλά και σε αυτά που μας ενδιαφέρουν περισσότερο για την αξιολόγηση της ταξινόμησης, δηλαδή την εύρεση ακρίβειας ταξινόμησης για κάθε βάση. Αυτό το αποτέλεσμα για κάθε αλγόριθμο και βάση θα είναι αυτό που θα τεθεί σε σύγκριση για να δούμε ποιος αλγόριθμος είναι ο καταλληλότερος για την ταξινόμηση της βάσης αυτής. Όπου η ακρίβεια είναι μεγαλύτερη και ο χρόνος τρεξίματος ικανοποιητικός, αυτός ο αλγόριθμος θα θεωρηθεί καταλληλότερος.

Τα αποτελέσματα της ακριβείας ταξινόμησης (classified accuracy) για το C4.5 μπορούν να θεωρηθούν αρκετά καλά αφού όπου υπάρχει αποτέλεσμα είναι σχεδόν πάντα πάνω από 80% (εκτός από μια βάση) και τις περισσότερες φορές πάνω από 90%. Αντίθετα τα αποτελέσματα για το CN2 αλγόριθμο δε μπορούν να θεωρηθούν καλά αφού στις περισσότερες βάσεις δεν έδωσε αποτελέσματα ακριβείας άλλα και σε αυτές που έδωσε η ακρίβεια ήταν χαμηλή (εκτός από μια που ήταν 100%).

Συνοψίζοντας, ο C4.5 είναι ταχύτερος, ακριβέστερος και έτρεξε σε βάσεις στις οποίες ο CN2 δεν έδωσε αποτέλεσμα.

4.6 Σημασία αποτελεσμάτων για κάθε βάση

- Car: Στη βάση αυτή γίνεται αξιολόγηση των οχημάτων με γνώμονα τρεις κατηγορίες: την τιμή, την τεχνολογία και την άνεση. Για αυτές τις κατηγορίες η βάση δίνει κάποια αποτελέσματα σωστών τιμών. Ο CN2 προέβλεψε σωστά τις τιμές αυτών των κατηγοριών κατά 61,8% και ο C4.5 σωστά κατά 81%.

- Tic-tac-toe: Η βάση κωδικοποιεί κάποια πιθανά σύνολα συνθέσεων πίνακα σε ένα παίγνιο όπου το x παίζει πρώτο και εξετάζει την πιθανότητα αν θα νικήσει ή οχι. Για αυτές τις δύο περιπτώσεις ο CN2 έδωσε σωστά αποτελέσματα κατα 64,9% και ο C4.5 73%.

- Splice Junction: Η βάση περιέχει ακολουθίες DNA τις οποίες ταξινομεί σε δύο κατηγορίες ανάλογα με τη σειρά και τη θέση που διατηρούν και αποσυνδέουν τα τμήματα του γενετικού κώδικα. Ο C4.5 ήταν ακριβής στην ταξινόμηση κατά 91% ενώ ο CN2 έκανε το πρόγραμμα να μην ανταποκρίνεται.

ΚΕΦΑΛΑΙΟ 5

5.1 Σύνοψη

Η παρούσα εργασία με αντικείμενο τους αλγόριθμους ταξινόμησης στην εξόρυξη δεδομένων ξεκινά με την επεξήγηση για την εξαγωγή της πληροφορίας, τον ορισμό της εξόρυξης δεδομένων και την ανεύρεση γνώσης, όπου γίνεται ανάλυση των εσωτερικών πηγών δεδομένων, των χαρακτηριστικών των δεδομένων καθώς και των τεχνικών εξόρυξης. Στη συνέχεια του πρώτου κεφαλαίου παρουσιάζεται η αλλαγή στον τύπο έκφρασης των ερωτήσεων και των αποτελεσμάτων της εξόρυξης δεδομένων, η διαδικασία και η περιγραφή των χαρακτηριστικών των αλγορίθμων εξόρυξης και της εξόρυξης από διαφορετικές πηγές δεδομένων. Ακολουθεί η ομαδοποίηση με τον ορισμό, τα κριτήρια και τις κατηγορίες αλγορίθμων ομαδοποίησης, η συσχέτιση όπου περιγράφονται οι κανόνες συσχέτισης αλλά και η διαδικασία εξαγωγής κανόνων συσχέτισης. Στη μέση του πρώτου κεφαλαίου βρίσκεται η ταξινόμηση με τον ορισμό της και την περιγραφή της διαδικασίας, η απόδοση και η εξελισσιμότητα αλγορίθμων όπως και η χρησιμότητα των αποτελεσμάτων εξόρυξης. Έπειτα, η περιγραφή της ομοιότητας χρονολογικών σειρών και οι διαδικασίες εύρεσής τους. Το πρώτο κεφάλαιο τελειώνει με τον ορισμό και τις τεχνικές της απεικόνισης και μείωσης διαστάσεων.

Το δεύτερο κεφάλαιο αρχίζει με την ονομαστική παράθεση των ειδών της ταξινόμησης. Πρώτο από αυτά, αναλύεται το είδος δέντρων απόφασης με τον ορισμό, τα βήματα κατασκευής, τις φάσεις, κάποιους αλγόριθμους και παραδείγματα για αυτό το είδος. Ακολουθούν οι μέθοδοι στους κανόνες απόφασης με τον ορισμό, την ανάλυση και την αναφορά των πιο διαδεδομένων αλγόριθμων. Στη συνέχεια, αναλύονται ο ορισμός, οι κατηγορίες και τα βήματα των τεχνητών νευρωνικών δικτύων. Έπειτα, γίνεται μία αναφορά στις στατιστικές μεθόδους ταξινόμησης και στον ορισμό του αφελή ταξινομητή Bayes. Επίσης παρουσιάζονται η διαδικασία και ο

ορισμός για τα δίκτυα Bayes, η περιγραφή και ο σημαντικότερος αλγόριθμος των μεθόδων μάθησης κατά περίπτωση K-κοντινότερος γείτονας, η περιγραφή του είδους μηχανών διανυσμάτων υποστήριξης και τέλος οι λοιπές μέθοδοι και αλγόριθμοι, όπου αναφέρονται ο επαγωγικός λογικός προγραμματισμός, τα υβριδικά συστήματα και άλλα.

Το τρίτο κεφάλαιο ξεκινά με τους βασικούς αλγόριθμους των δέντρων απόφασης και συγκεκριμένα ο ID3. Ο αλγόριθμος περιγράφεται αναλυτικά ως προς τη λειτουργία, τα στοιχεία, την επιλογή ιδιοτήτων καθώς και τρία παραδείγματα. Το κεφάλαιο συνεχίζεται με τον πολύ σημαντικό αλγόριθμο C4.5 με μία βασική περιγραφή της λειτουργίας του, της ιστορίας του, των κριτηρίων αλλά και τις βελτιώσεις από τον ID3. Ακολουθεί το ιστορικό, ο ορισμός και η περιγραφή της λειτουργίας των αλγορίθμων Sliq και Sprint. Ο αλγόριθμος CART είναι ο επόμενος, με τα πλεονεκτήματα και τα λειτουργικά χαρακτηριστικά του που κλείνουν το είδος δέντρων απόφασης. Κατόπιν, εξετάζονται οι αλγόριθμοι στις μεθόδους κανόνων απόφασης. Αρχικά, ο βασικότερος εξ αυτών – CN2, όπου δίδονται τα χαρακτηριστικά, η διαδικασία και η λειτουργία του. Επόμενος αλγόριθμος είναι ο AQ με την περιγραφή του και ένα παράδειγμα σε γλώσσα Prolog. Μετέπειτα, παρατίθεται η περιγραφή και η βασική ιδέα του αλγορίθμου CL^2 .

Στην ίδια κατηγορία βρίσκονται οι κανόνες εκμάθησης όπου αναλύονται ο ορισμός και η τεχνική , οι μέθοδοι βασισμένες στους κανόνες απόφασης, η μείωση περικοπής σφάλματος REP, η επαυξητική μείωση περικοπής σφάλματος IREP, η διαδικασία και ο αλγόριθμος IREP και τέλος η επαναλαμβανόμενη επαυξητική περικοπή για τη μείωση σφάλματος RIPPER. Επόμενη κατηγορία, οι μέθοδοι μάθησης κατά περίπτωση με την λειτουργία και την περιγραφή του αλγόριθμου kNN. Συνεχίζοντας, παρατίθεται η έννοια και ένα παράδειγμα για τις μηχανές διανυσμάτων υποστήριξης και ο ορισμός της ώθησης καθορισμού του περιθωρίου (margin). Ακολουθούν μερικά παραδείγματα αλγορίθμων βασισμένα στο περιθώριο. Τέλος, η περιγραφή και συμβολή του αλγορίθμου ADABOOST και τα χαρακτηριστικά του αλγορίθμου EM.

Το τέταρτο κεφάλαιο ξεκινά με μία αναφορά στους λόγους της εξάπλωσης της εξόρυξης δεδομένων και στις καθημερινές και επιστημονικές πρακτικές εφαρμογές των τεχνικών εξόρυξης δεδομένων και συγκεκριμένα της ταξινόμησης. Στη δεύτερη ενότητα, γίνεται η περιγραφή των βάσεων που χρησιμοποιούνται παρακάτω, οι λόγοι δημιουργίας τους και η επεξήγησή τους. Στο τρίτο μέρος δίνονται οι πίνακες των αποτελεσμάτων των τρεξιμάτων των αλγορίθμων C4.5 και CN2, ο χρόνος τρεξίματος καθώς και η ακρίβεια ταξινόμησης όπως εμφανίζονται μετά την επεξεργασία των βάσεων στο πρόγραμμα Διογένης. Στο τέταρτο μέρος με τίτλο «αξιολόγηση αποτελεσμάτων» γίνεται η αξιολόγηση του προγράμματος Διογένης, η αξιολόγηση για κάθε βάση και η σύγκριση μεταξύ C4.5 και CN2. Στο τέλος του κεφαλαίου παρουσιάζεται συνοπτικά η πρακτική επεξήγηση των αποτελεσμάτων για κάθε βάση.

Στο τέλος της εργασίας βρίσκεται το Παράρτημα Α, όπου γίνεται μία αναλυτική περιγραφή του προγράμματος Διογένης και των βημάτων για τη χρήση του. Δίνονται πληροφορίες για τη δημιουργία νέου project, το άνοιγμα ενός project, την αποθήκευσή του, την επιλογή παλαιάς εκτέλεσης, την εύρεση ακρίβειας και τη δημιουργία κατάστασης κανόνων. Τέλος, οδηγίες χρήσης για τις εφαρμογές ταξινόμησης και το τρέξιμο δεδομένων για τους αλγόριθμους C4.5 και CN2.

5.2 Μελλοντικές προοπτικές της ταξινόμησης

Μία νέα έμφαση στην συμμόρφωση, την ανεύρεση, την αρχειοθέτηση και την προέλευση αποτελεί ουσιαστική πρόκληση για τις υπάρχουσες κατηγορίες ταξινόμησης δεδομένων. Οι σημερινές πρακτικές επιχειρηματικής αξίας περιλαμβάνουν πολιτικές διατήρησης τύπου «μη διαγραφής» όπως επίσης και απόδοσης, διαθεσιμότητας και χαρακτηριστικά ανάκτησης που υποστηρίζουν τις προσπάθειες ταξινόμησης δεδομένων. Αν και το βασισμένο στο χρόνο σχήμα εξακολουθεί να υπερισχύει, οι προσπάθειες πρέπει να εξελιχθούν ώστε να ενσωματώσουν πιο πλούσια χαρακτηριστικά ταξινόμησης.

Ειδικότερα, αυτού του είδους η επέκταση θα πρέπει να πραγματοποιηθεί αποσκοπώντας στην αυτοματοποίηση αναθέτοντας δυναμικά τα μεταδεδομένα στις βάσεις δεδομένων κατά τη δημιουργία ή τη χρήση. Οι μελλοντικές προσπάθειες ταξινόμησης δεδομένων θα περιλαμβάνουν ευρύτερες προοπτικές και θα χρησιμεύουν ως έναυσμα πολλαπλών επιχειρησιακών πρωτοβουλιών, συμπεριλαμβανομένου των πληροφοριών για τη διαχείριση του κύκλου ζωής, της κλιμακωτής αποθήκευσης, της αρχειοθέτησης ηλεκτρονικού ταχυδρομείου, της υποστήριξης λήψης αποφάσεων, της εξόρυξης δεδομένων και της ηλεκτρονικής διαχείρισης περιεχομένου. Εν συντομία, η ταξινόμηση δεδομένων θα χρησιμεύσει στο μέλλον ως το θεμέλιο για τη διαχείριση πληροφοριών και χωρίς την αυτόματη ταξινόμηση είναι λίγες οι πιθανότητες να πετύχει στην υποστήριξη αυτών των συχνά πολύπλοκων προσπαθειών.

ΠΑΡΑΡΤΗΜΑ Α

Στην εφαρμογή Διογένης έχουν υλοποιηθεί επιμέρους εφαρμογές απλής κατανεμημένης εξόρυξης δεδομένων. Όλες οι εφαρμογές έχουν παρόμοιο περιβάλλον και λειτουργούν σε μεγάλο βαθμό με τον ίδιο τρόπο.

Με την εκκίνηση του προγράμματος εμφανίζεται η κεντρική φόρμα. Σε αυτήν περιέχεται και μια δεντρική δομή με τις εκτελέσεις που έχουν γίνει ως τώρα. Από το menu αυτής της φόρμας τρέχουν οι εφαρμογές εξόρυξης και κατανεμημένης εξόρυξης δεδομένων.

Γενικά, οι εφαρμογές απλής εξόρυξης δεδομένων έχουν παρόμοια μορφή. Στην πρώτη σελίδα πρέπει να οριστούν τα δεδομένα που θα χρησιμοποιηθούν, ποια άση δεδομένων, ποιος πίνακας ποια πεδία κλπ. Στην δεύτερη σελίδα ορίζονται οι παράμετροι της εφαρμογής, ανάλογα με τη φύση και τα χαρακτηριστικά της. Στην συνέχεια γίνεται η εκτέλεση και εμφανίζονται τα αποτελέσματα σε μία η περισσότερες σελίδες. Τα αποτελέσματα καταχωρούνται παράλληλα και σε αρχεία για περαιτέρω εξέταση.

Οι εφαρμογές απλής εξόρυξης δεδομένων βρίσκονται κάτω από το menu Algorithms. Υπάρχουν δυο εφαρμογές ταξινόμησης, αυτή που βασίζετε στον αλγόριθμο CN2 και αυτή του C4.5 , μια εφαρμογή ομαδοποίησης που βασίζετε στον αλγόριθμο K-modes και μια εφαρμογή συσχέτισης που βασίζετε στον αλγόριθμο Apriori.

Οι εφαρμογές κατανεμημένης εξόρυξης έχουν την ιδιομορφία ότι χρησιμοποιούν, στην πλειοψηφία τους , αποτελέσματα από τις αντίστοιχες εφαρμογές εξόρυξης δεδομένων. Στην πρώτη σελίδα συνήθως επιλέγουμε τις έτοιμες εκτελέσεις που θα χρησιμοποιήσουμε . Πρέπει παράλληλα να συνδεθούμε σε κάποια βάση οπού θα γίνει η κατανεμημένη εξόρυξη δεδομένων . Στην συνεχεία τίθενται οι παράμετροι και

λαμβάνονται τα αποτελέσματα όπως ακριβώς και στην περίπτωση της απλής εξόρυξης δεδομένων. Η λογική της κατανεμημένης εξόρυξης είναι η βελτίωση του χρόνου εκτέλεσης για την λήψη των τελικών αποτελεσμάτων.

Υπάρχουν τέσσερις εφαρμογές κατανεμημένης εξόρυξης δεδομένων κάτω από το menu MetaMining: η εφαρμογή κατανεμημένης ομαδοποίησης MetaClustering , η εφαρμογή κατανεμημένης ταξινόμησης MetaClassification , η εφαρμογή παραγωγής κατανεμημένων κανόνων MetaRules και η εφαρμογή κατανεμημένης συσχέτισης MetaAssociation .

Α.1 Διεπαφή εφαρμογής

Σε αυτό το κεφάλαιο θα περιγραφεί το περιβάλλον της διεπαφής της εφαρμογής diogenis. Το περιβάλλον της διεπαφής είναι απλέ και οι λειτουργίες του περιορίζοντε σε αποθήκευση/άνοιγμα των εκτελέσεων, καθώς και της ειδικής λειτουργίας εύρεσης της ακρίβειας ενός συγκεκριμένου ταξινομητή.

Πιο συγκεκριμένα, στην αρχική οθόνη εμφανίζεται ένας κενός χώρος , ο οποίος χρησιμοποιείται για την εμφάνιση των εκτελέσεων του τρέχοντος project. Η λογική της εφαρμογής είναι η εξής. Κάθε εκτέλεση οποιουδήποτε αλγορίθμου που ΄΄τρέχει΄΄ ο χρήστης πρέπει να είναι μέρος ενός project. Σε αυτό το project μπορεί να αποθηκευτεί. Όταν ο χρήστης επιθυμεί να δει τις λεπτομεριές μιας παλιάς εκτέλεσης ανακαλεί το project στο οποίο είχε αποθηκεύσει την εκτέλεση και έπειτα μπορεί να επιλέξει τη συγκεκριμένη εκτελέσει. Ένα project μπορεί να περιέχει παραπάνω από μια εκτέλεσης διαφορετικού η ίδιου τύπου. Όλες οι εκτελέσεις πρέπει να έχουν διαφορετικό όνομα.

Οι μόνες επιλογές που είναι ενεργοποιημένες στο μενού με την έναρξη της εφαρμογής είναι το File->New Project/Open project και το Tools->Classify . Το

τελευταίο δεν απαιτεί την ύπαρξη ανοικτού project και περιγράφεται λεπτομερώς παρακάτω.

Α.1.1 Δημιουργία νέου project

Για να δημιουργηθεί ένα νέο project επιλέγουμε File->New project η επιλέγουμε το πρώτο κουμπί προς τα αριστερά. Έπειτα ο χρήστης πρέπει να δηλώσει το όνομα του αρχείου στο οποίο θα αποθηκευτεί το project και τη θέση του στο δίσκο. Τα αρχεία που αποθηκεύονται τα project του diogenis έχουν την κατάληξη .dpr (diogenis project).

Αφού ο χρήστης επιλέξει 'Save' θα εμφανιστεί στην οθόνη το όνομα του project .
Πλέον όλες οι επιλογές είναι ενεργοποιημένες και ο χρήστης μπορεί να εκτελέσει οποιοδήποτε αλγόριθμο επιθυμεί. Όλες οι εκτελέσεις θα προστίθενται κάτω από το τρέχον project.

Α.1.2 Άνοιγμα project

Ο χρήστης έχει τη δυνατότητα να ανοίξει ένα παλαιό project που έχει αποθηκεύσει στο δίσκο, και είτε να δει τις παλαιές του εκτελέσεις, είτε να προσθέσει καινούριες σε αυτό. Για να ανοίξει ένα παλιό project ο χρήστης πρέπει να επιλέξει File->Open Project και έπειτα να αναζητήσει το αρχείο .dpr που αντιστοιχεί στο project.

Όταν ο χρήστης επιλέξει 'OPEN' στην οθόνη θα εμφανιστούν όλες οι εκτέλεσης που περιέχονται στο επιλεχθέν project.

Α.1.3 Επιλογή παλαιάς εκτελέσεις

Για να επαναφέρει μια παλιά εκτέλεση ο χρήστης θα πρέπει πρώτα να «ανοίξει» το project όπου είναι αποθηκευμένη και έπειτα να κάνει double-click στην εκτέλεση που επιθυμεί να επαναφέρει .Σημειώνεται ότι όλες οι επιλογές θα είναι απενεργοποιημένες και ο χρήστης δεν μπορεί να αλλάξει κάποιες παραμέτρους και να ξαναεκτελέσει κάποιον αλγόριθμο .Αν επιθυμεί κάτι τέτοιο θα πρέπει να δημιουργήσει μια νέα εκτέλεση εξ αρχής .Επίσης πρέπει να σημειωθεί ότι κατά τη διάρκεια που ο χρήστης έχει ανοιχτό ένα παράθυρο μια παλαιάς (ή και νέας εκτέλεσης) δεν επιτρέπεται να ανοίξει άλλο παράθυρο για εκτέλεση.

Α.1.4 Αποθήκευση project

Όταν ολοκληρώνεται επιτυχώς μια εκτέλεση τότε το όνομα αυτής προστίθεται κάτω από το όνομα του τρέχοντος project. Για να αποθηκευτεί η εκτέλεση στο project ώστε να αποτελεί μέρος του την επομένη φορά που ο χρήστης ΄΄ανοιξεί΄΄ το project πρέπει πριν κλείσει το project να επιλέξει file ->save.αν δε το κάνει η εκτέλεση δεν θα έχει αποθηκευτεί στο project.Κάθε ολοκληρωμένη εκτέλεση δημιουργεί μια σειρά από διαφορετικά αρχεία ,που έχουν κοινό όνομα και διαφορετική κατάληξη(βλ.εκάστοτε αλγόριθμο).Τα αχρεία αυτά αποθηκεύονται στο ίδιο directory με το project το οποίο είναι ανοικτό κατά την εκτέλεση τους .Ο χρήστης μπορεί να προσθέσει και χειρονακτικά εκτελέσεις σε ένα project επεξεργάζοντας το αντίστοιχο .dpr αρχείο ,αλλά αυτό συνίσταται να γίνεται μονό εάν ο χρήστης γνωρίζει τη δομή τους.

Α.1.5 Εύρεση ακρίβειας

Το εργαλείο αυτό φτιάχτηκε για να υπολογίζετε η ποιότητα ενός ταξινομητή .Το εργαλείο εφαρμόζει ένα σύνολο κανόνων πάνω σε ένα σύνολο δεδομένων και

προσπαθεί να προβλέψει γνωστές τιμές .Το εργαλείο συγκρίνει τις προβλέψεις με τις σωστές τιμές και υπολογίζει την ακρίβεια της ταξινόμησης.

Πιο συγκεκριμένα ο χρήστης πρέπει να επιλέξει tools ->classify, χωρίς να υπάρχει ανάγκη για ανοικτό project.

Ο χρήστης έχει τη δυνατότητα να προσθέσει στη λίστα 'select miner' όποιες εκτελέσεις επιθυμεί να συγκρίνει. Οι εκτελέσεις μπορεί να ανήκουν και σε διαφορετικά project. Ο χρήστης πρώτα επιλέγει το κουμπί 'browse' για να αναζητήσει ένα project με εκτελέσεις που τον ενδιαφέρουν.

Όταν ο χρήστης επιλέξει 'OPEN', το όνομα του αρχείου θα εμφανιστεί στο πεδίο 'add the project to the list of runs'. Έπειτα ο χρήστης πρέπει να επιλέξει το κουμπί 'add' για να προσθέσει τις εκτελέσεις του συγκεκριμένου project στη λίστα. Η διαδικασία επαναλαμβάνεται για να προστεθούν στη λίστα εκτέλεσης από άλλα projects.

Ο χρήστης θα πρέπει να επιλέξει μόνο μια εκ των εκτελέσεων της λίστας κάθε φορά. Αν έχει επιλεγμένες παραπάνω, το εργαλείο αγνοεί όλες πλην της πρώτης. Η εκτελέσει που θα επιλέγει θα εφαρμοστεί στο σύνολο δεδομένων που επιλέγει ο χρήστης στο επόμενο βήμα.

Το σύνολο δεδομένων ορίζεται, κάνοντας μια σύνδεση με μια βάση. Σημειώνεται ότι ο πίνακας αυτός θα πρέπει να έχει τα ίδια χαρακτηριστικά (related και target) με την εκτέλεση.

Αν ο χρήστης επιθυμεί το εργαλείο να του βγάλει στατιστικά επιτυχίας για κάθε διακριτή τιμή του χαρακτηριστικού-στόχου , τότε θα πρέπει να ενεργοποιήσει 'detailed accuracy'. Αν η επιλογή αυτή είναι ενεργοποιημένη τότε θα δημιουργηθεί ένα αρχείο, με όνομα το όνομα της εκτέλεσης που ελέγχεται και κατάληξη '.acc', το οποίο θα υποθηκευτεί στο ίδιο directory με τα αρχεία της εκτέλεσης.

Τέλος ο χρήστης πρέπει να επιλέξει το κουμπί ‘classify’.

Α.1.6 Δημιουργία κατάστασης κανόνων

Το σύνολο των κανόνων που προήρθαν από μια εκτέλεση, μπορούν να παρουσιαστούν σε κατάλληλη Κατάσταση Κανόνων . Η Κατάσταση Κανόνων παράγεται σε μορφή αρχείου Microsoft Word. Έτσι μπορεί να παρουσιαστεί στον χρήστη για παραπέρα επεξεργασία (εκτύπωση, σχολιασμό, κλπ). Βέβαια, η Κατάσταση Κανόνων μπορεί να αξιοποιηθεί αν υπάρχει πίνακας μετάφρασης.

Η διαδικασία παράγωγης της Κατάστασης Κανόνων έχει ως ακολούθως. Κατ’ αρχήν επιλέγουμε ‘reporting’ και ‘Export Report’,από την βασική οθόνη της εφαρμογής.

Αν δεν υπάρχει πίνακας μετάφρασης, εμφανίζεται διαγνωστικό μήνυμα. Πάντως, σε κάθε περίπτωση δημιουργείται Κατάσταση Κανόνων, η προεπισκόπηση της οποίας μπορεί στο πεδίο ‘report’.

Για μια αποτελεσματική παρουσίαση των αποτελεσμάτων (αστικοποίηση δεδομένων) με δυνατότητα αιτιολόγησης των εξαχθεισών γνώσεων, το σύστημα ΔΙΟΓΕΝΗΣ, έχει σύνδεση με το έμπειρο σύστημα InstantTea.

Όταν έχει καθαρισθεί από τον χρηστή να παραχθεί το αρχείο σύνδεσης (*.exp) με το έμπειρο σύστημα , τότε απλώς ο χρήστης εκκινεί το έμπειρο σύστημα και επιλέγει ως βάρη γνώσεων το συγκεκριμένο αρχείο σύνδεσης. Η παραπέρα διαδικασία περιγράφεται από το εγχειρίδιο χρήσης του έμπειρου συστήματος.

Α.2 Εξόρυξη δεδομένων

Α.2.1 Εφαρμογές ταξινόμησης (classification)

Το submenu classification μας οδηγεί στην επιλογή κάποιων εκ των δυο υλοποιήσιμων αλγορίθμων, CN2 και C4.5. Επιλέγοντας κάποιων εκ των δυο τρέχει η αντίστοιχη εφαρμογή.

Α.2.1.1 CN2

Γενικά ο CN2 είναι ένας αλγόριθμος εξαγωγής κανόνων ταξινόμησης από ένα σύνολο εγγραφών. Ορίζουμε στον αλγόριθμο από ποια πεδία μας, ενδιαφέρει να εξάγουμε κανόνες και αυτοματοποιημένα αυτοί εξάγονται κανόνες με βάση την ακρίβεια και την ποιότητα τους πάνω στα δεδομένα εισόδου. Οι κανόνες έχουν τη μορφή συμπλεγμάτων, πολλαπλών συνθηκών στα πεδία που συνδέονται με τελεστές ΚΑΙ , και μιας τιμής από το πεδίο τάξης. Ένας τέτοιος κανόνας είναι π.χ. :
ΠΕΡΙΟΧΗ=ΑΤΤΙΚΗ ΚΑΙ ΕΠΑΓΓΕΛΜΑ=ΕΜΠΟΡΟΣ ΤΟΤΕ ΠΕΛΑΤΗΣ=ΚΑΛΟΣ
Η υλοποιημένη έκδοση του CN2 αλγορίθμου αποτελείτε , όπως όλοι οι αλγόριθμοι, από μια φόρμα στην οποία υπάρχουν πολλές σελίδες (tabs), 4 για την ακρίβεια.

Σελίδα Data

Η πρώτη σελίδα είναι αυτή που επιλέγεται η είσοδος του αλγορίθμου. Συμπληρώνουμε το όνομα της βάσης με την οποία θέλουμε να συνδεθούμε και αφού κάνουμε log in, έχουμε τη δυνατότητα να επιλέξουμε τον πίνακα που μας ενδιαφέρει.
Επιλογή πίνακα

Σε σχέση με τον πίνακα πρέπει να επιλέξουμε επιπλέον αν θέλουμε να δημιουργήσουμε αντίγραφο του πίνακα και αν το αντίγραφο (ή ο αρχικός πίνακας σε περίπτωση που δεν τον αντιγράψουμε) θα σβηστεί στο τέλος από τη βάση ή όχι. Πρέπει να πούμε εδώ ότι ο CN2 σβήνει εγγραφές από τον πίνακα καθώς τρέχει. Συνιστάτε επομένως η αντιγραφή του, είτε μέσα από την εφαρμογή του, με την αντίστοιχη επιλογή, είτε εξωτερικά, από τη βάση δηλαδή, μια μεγαλύτερη ταχύτητα.

Επιλογή πεδίων

Ο CN2 είναι αλγόριθμος ταξινόμησης οπότε πρέπει να επιλέγει ένα σύνολο πεδίων επιλογέων και ένα πεδίο τάξης το οποίο να μην περιέχετε στο σύνολο αυτό. Τα πεδία επιλογέων επιλέγονται από τη λίστα των related attributes και το πεδίο τάξης από τη λίστα επιλογής του target attribute. Αφού όλα γίνουν κανονικά θα ενεργοποιηθεί το κουμπί που θα μας μεταφέρει στην επόμενη σελίδα, στη σελίδα των παραμέτρων.

Σελίδα parameters

Επιλογή παραμέτρων

Οι παράμετροι του CN2ειναι ελάχιστες, 2 μόνο για την ακρίβεια. Ο χρήστης πρέπει να επιλέξει κατώφλι σημαντικότητας και μέγιστο μέγεθος του STAR. Υλοποιητικά οι δυο αυτές παράμετροι έχουν πολύ συγκεκριμένη λειτουργία.

Η σημαντικότητα είναι μια από τις δυο μετρικές που ορίζουν την ποιότητα ενός κανόνα. Η άλλη είναι η εντοπία που μας δείχνει κατά πόσο ο κανόνας αντιστοιχεί σε εγγραφές μια μόνο τάξης ή και περισσοτέρων. Η σημαντικότητα δείχνει κατά πόσο ο κανόνας έχει αξία σε σχέση με τη κατανομή των εγγραφών σε ότι αφορά την τάξη τους. Ας υποθέσουμε π.χ. ότι έχουμε ένα σύνολο εγγράφων όπου το 90% ανήκουν σε μια συγκεκριμένη τάξη. Κανόνες που ταξινομούν σε αυτή τη συγκεκριμένη τάξη είναι στατιστικά λιγότερο σημαντική από αυτούς που ταξινομούν σε άλλη. Η σημαντικότητα μετράει ακριβώς την ποιότητα αυτού του κανόνα. Ακόμα και αν η

εντροπία αυτού του κανόνα είναι μηδενική, με τέτοια κατανομή εγγράφων είναι εύκολο να υπάρξουν πολλοί κανόνες μηδενικής εντροπίας με αυτήν την τάξη. Η σημαντικότητα όμως αυτών των κανόνων θα είναι σίγουρα μικρή. Ορίζοντας λοιπόν κατώφλι στη σημαντικότητα των κανόνων που μας ενδιαφέρουν ευνοούμε τους σημαντικότερους κανόνες.

Το κατώφλι σημαντικότητας μπορεί να είναι οποιοσδήποτε πραγματικός αριθμός μεγαλύτερος ή ίσως με το μηδέν. Στην ουσία ορίζει τι βαθμό σημαντικότητας πρέπει να έχει ο κανόνας για να θεωρηθεί επαρκώς σημαντικός από τον αλγόριθμο. Εμποδίζει έτσι ασήμαντους, σύμφωνα με το συγκεκριμένο κριτήριο, κανόνες από το να προκύψουν σαν έξοδο ακόμα και αν η εντροπία τους είναι πολύ χαμηλή. Δίνει έτσι προτεραιότητα στη δύναμη των κανόνων παρά στην ακρίβεια.

Εδώ όμως πρέπει να εξηγήσουμε ένα λεπτό σημείο. Το κατώφλι σημαντικότητας συγκρίνετε με τη σημαντικότητα όχι μόνο των τελικών κανόνων αλλά και όλων των δυνατών περιπτώσεων κανόνων που εξετάζονται μέχρι να παραχθεί ο τελικός κανόνας. Υπάρχει περίπτωση κάποιος ημιτελείς κανόνας να φαίνετε ασήμαντος και να κοπεί ακόμα και αν μια περεταίρω εξειδίκευση του, αν της επιτρέπονταν να δημιουργηθεί, θα ήταν επαρκώς σημαντική. Το συμπέρασμα είναι, μετά από πολλά πειράματα, ότι δεν πρέπει να υπερβάλουμε με την τιμή του κατωφλιού. Μια σχετικά μικρή, μη μηδενική τιμή γύρο στα 10 αρκεί για να αποφύγουμε τετριμμένους κανόνες.

Περνάμε τώρα στη δεύτερη παράμετρο. Ο αλγόριθμος ξεκάνει με ένα σύνολο επιλογέων, δηλαδή διακριτικές τιμές πεδίων, τους οποίους εξάγει από τα πεδία που ορίζει ο χρήστης. Υπολογίζει την εντροπία και της σημαντικότητας του καθενός και κρατάει τους καλύτερους σε ένα σύνολο συμπλεγμάτων που λέγετε STAR. Στη συνέχεια κάνει όλους τους δυνατούς συνδυασμούς μεταξύ του συνόλου των επιλογέων και των συμπλεγμάτων του STAR παράγοντας ένα νέο σύνολο από συμπλέγματα. Αδειάζουμε το STAR, υπολογίζονται οι δυο μετρικές για τα συμπλέγματα αυτά, τα καλύτερα αποθηκεύονται εκ νέου στο STAR κ.ο.κ.

Το μέγιστο μέγεθος του STAR είναι ένας ακέραιος που αριθμεί πόσα καλύτερα συμπλέγματα θα αποθηκεύονται για περεταίρω εξειδίκευση στον επόμενο κύκλο. Για εξαντλητικό ψάξιμο χρειάζονται μόνο μεγάλες τιμές. Η πολυπλοκότητα πάντως του αλγορίθμου αυξάνετε εκθετικά με αυτήν την παράμετρο. Έχει παρατηρηθεί ότι μια τιμή από τρία έως πέντε οδηγεί σε ικανοποιητικά αποτελέσματα.

Υπάρχει και ένα στοιχείο που πρέπει να συμπληρώσουμε, το Translate Rules που ενεργοποιείτε ή απενεργοποιείτε. Αυτό έχει σχέση απλώς με την έξοδο των κανόνων, αν θα βγουν σε ποιο αναγνώσιμη μορφή και μεταφρασμένη ή όχι. Για να λειτουργήσει αυτή η επιλογή πρέπει να υπάρχει πίνακας μετάφρασης για τον πίνακα που έχει επιλεγεί στην πρώτη σελίδα. Είναι θέμα του administrator να κατασκευάσει πίνακες μετάφρασης για κάθε πίνακα υποψήφιο για εξόρυξη δεδομένων.

Σελίδα Results

Αφού συμπληρώσουμε τις παραμέτρους μπορούμε μετά να εκτελέσουμε τον αλγόριθμο με τα αντίστοιχο πλήκτρο. Η έξοδος θα παρουσιαστεί στην επόμενη σελίδα με την ολοκληρώσει του τρεξίματος και θα αποθηκευθεί στο αντίστοιχο αρχείο του έργου.

Σελίδα Classify Table

Μαζί με τη σελίδα Results όπου παρουσιάζονται οι κανόνες, εμφανίζετε και η τέταρτη σελίδα της φόρμας η Classify Table. Η χρήση αυτής της σελίδας είναι προαιρετική και εξαρτάτε απ’το αν μας ενδιαφέρει να χρησιμοποιήσουμε τους κανόνες που μόλις πήραμε για να ταξινομήσουμε έναν άλλο πίνακα.

Στην περίπτωση που μας ενδιαφέρει κάτι τέτοιο με την βάση όπου βρίσκετε ο πίνακας προς ταξινόμηση, επιλέγουμε τον πίνακα και το πεδίο που πρόκειται να ταξινομηθεί. Ο πίνακας πρέπει να έχει τα πεδία που χρησιμοποιούνται στους κανόνες. Το πεδίο που έχει επιλογή για ταξινόμηση θα πρέπει προφανώς να μπορεί να

αποθηκεύσει αλφαριθμητικά δεδομένα. Αν περιέχει ήδη δεδομένα, αυτά θα καλυφθούν από τις νέες τιμές τάξεις.

Τελειώνοντας τις οδηγίες χρήσης αυτού του αλγόριθμου πρέπει να πούμε ότι παράγει πολύ χρήσιμους κανόνες αλλά πρέπει τα δεδομένα εισόδου να είναι τέτοια ώστε να μην υπάρχουν πολλές διακριτές τιμές, διαφορετικά η πολυπλοκότητα οδηγεί σε απαγορευτικούς χρόνους εκτέλεσης. Επίσης τα δεδομένα δεν πρέπει να περιέχουν κενές τιμές.

A. 2.1.2 C4.5

Γενικά ο C4.5 είναι ένας αλγόριθμος εξαγωγής δέντρων αποφάσεων. Παίρνει σαν είσοδο ένα σύνολο εγγραφών όπως και ο CN2 και παράγει ένα δέντρο απόφασης. Το δέντρο αυτό χρησιμοποιείτε τόσο σαν μοντέλο γνώσης για μελέτη και εκτίμηση όσο και σαν εργαλείο ταξινόμησης νέων εγγραφών. Οι κόμβοι του δέντρου είναι έλεγχοι που γίνονται σε κάποιο πεδίο. Οι διακριτές τιμές του πεδίου αυτού οδηγούν σε παιδιά όπου γίνονται έλεγχοι σε άλλα πεδία κ.ο.κ έως ότου καταλήξουμε σε φύλλα όπου υπάρχει και η κατηγορία των εγγραφών που οδηγούνται ως εδώ.

Η καρδιά του αλγορίθμου στη ουσία είναι ο τρόπος επιλογής των πεδίων. Ο τρόπος αυτός βασίζετε στο κέρδος της πληροφορίας. Όπως περιγράφετε και μαθηματικά στη θεωρητική παρουσίαση του αλγορίθμου, η ερώτηση << σε πια τάξη ανήκει μια εγγραφή του συνόλου ;>> απαιτεί ένα ποσό πληροφορίας για να απαντηθεί. Το ποσό αυτό μειώνετε όσο περισσότερες εγγραφές από το σύνολο ανήκουν σε μια συγκεκριμένη κατηγορία. Εφόσον λοιπόν επιλέγει ένα πεδίο που χωρίζει τις εγγραφές σε υποσύνολα όπου η τάξη που ανήκει μια εγγραφή αποσαφηνίζετε κάπως πιθανοτικά, η πληροφορία που απαιτείτε για να απαντηθεί η παραπάνω ερώτηση είναι μικρότερη αθροιστικά σε σχέση με αυτή που απαιτείτε στο αρχικό σύνολο. Επομένως ο έλεγχος με βάση αυτό το πεδίο μας παρέχει κέρδος πληροφορίας. Ο αλγόριθμος επιλέγει το πεδίο που δίνει το μεγαλύτερο κέρδος που χωρίζει καλύτερα τις εγγραφές.

Σελίδα Data

Η υλοποίηση του C4.5 έχει αντίστοιχο παρουσιαστικό με αυτή του CN2. Η πρώτη σελίδα είναι ακριβώς ίδια οπότε θα προχωρήσουμε κατευθείαν στην εξήγηση των παραμέτρων που είναι αρκετές.

Σελίδα Parameters

Επιλογή Criteria

Η πρώτη παράμετρος είναι το Criteria όπου ορίζουμε πιο κριτήριο θα χρησιμοποιηθεί κατά την εξαγωγή του δέντρου απόφασης. Αν επιλέξουμε το Ratio θα χρησιμοποιηθεί το κριτήριο αναλογίας κέρδους, διαφορετικά το απλό κριτήριο κέρδους πληροφορίας. Το πρώτο είναι ουσιαστικά μια εξέλιξη του δεύτερου όπου λαμβάνετε υπόψη όχι απλώς πόσο καλά <<χωρίζονται>> οι εγγραφές σε σχέση με την τιμή τάξης τους, αλλά και πια είναι η μορφή των δεδομένων σε σχέση με το <<σπάσιμο>> αυτό. Έχει παρατηρηθεί ότι το απλό κριτήριο κέρδους ευνοεί ελέγχους σε πεδία με πολλές διακριτές τιμές. Το πρόβλημα αυτό λύνετε με το κριτήριο αναλογίας κέρδους ως εξής. Μελετάει την πληροφορία όχι μόνο σε σχέση με την τιμή της τάξης αλλά και σε σχέση με την τιμή του πεδίου ελέγχου. Η πληροφορία που χρειαζόμαστε για να μάθουμε την τιμή του πεδίου αυξάνει προφανώς όσο περισσότερες διακριτές τιμές υπάρχουν. Αν λοιπόν διαιρέσουμε το κέρδος με την πληροφορία αυτή παίρνουμε μια αρκετά καλύτερα μετρική, που εμποδίζει τα πολύ σύνθετα <<σπασίματα>>. Γενικά είναι καλύτερα να χρησιμοποιείτε το κριτήριο αναλογίας κέρδους και έχει παραμετροποιηθεί απλός για λογούς πειραματισμών.

Επιλογή Κλαδέματος

Η επόμενη παράμετρος αφορά το <<κλάδεμα>> ή όχι του δέντρου. Εφόσον επιλεγεί η επιλογή Prune Check θα βγει σαν έξοδος μια <<κλαδεμένη>> έκδοση του δέντρου. Η έκδοση αυτή θα έχει λιγότερους κόμβους και, σύμφωνα με τον αλγόριθμο, θα είναι

απαλλαγμένη από εξειδικεύσεις που μειώνουν την αξία του δέντρου σαν μοντέλο πρόβλεψης και ταξινόμησης.

Ο τρόπος που γίνετε το κλάδεμα αφορά στην εκτίμηση του λάθους ταξινόμησης σε ένα κόμβο σε σχέση με την εκτίμηση του λάθους αθροιστικά στα παιδιά του. Άμα το λάθος στον κόμβο εκτιμηθεί μικρότερο απ' ότι στα παιδιά του τότε το δέντρο σε εκείνο το σημείο <<κλαδεύετε>> και ο κόμβος μετατρέπετε σε φύλλο. Για την εκτίμηση του λάθους υπάρχουν προφανώς μόνο στατιστικά στοιχεία και όχι πραγματικά. Χρησιμοποιούμε την δυονιμική καταγωγή για να μας δώσει μια ''απαισιόδοξη'' στατιστική εκτίμηση του λάθους με βάση την κατανομή των εγγραφών του συνόλου εκπαίδευσης. Η εκτίμηση λάθους είναι μεγάλη για εξειδίκευσης κόμβων που φαίνονται ότι μάλλον δεν στέκουν γενικά αλλά αποτελούν ειδικές περιπτώσεις του συνόλου εκπαίδευσης.

Ο βαθμός κλαδέματος εξαρτάτε από δυο παραμέτρους. Το επίπεδο εμπιστοσύνης (confidence) και το επιπρόσθετο λάθος (error overhead). Το επίπεδο εμπιστοσύνης δηλώνει ουσιαστικά πόσο πολύ εμπιστευόμαστε τα δεδομένα που έχουμε σα δείγμα από ένα άπειρο σύνολο δεδομένων. Η εκτίμηση λάθους που καθοδηγεί και το κλάδεμα, αυξάνει όσο περισσότερο επίπεδο εμπιστοσύνης έχουμε. Οπότε μικρό επίπεδο εμπιστοσύνης σημαίνει πιο δραστικό κλάδεμα. Η συνηθισμένη τιμή γι' αυτή την παράμετρο είναι 0.25 (25%).

Το επιπρόσθετο λάθος είναι ένας πιο όμως τρόπος να αυξηθεί η δραστικότητα του κλαδέματος. Το λάθος αυτό ενισχύει αθροιστικά την εκτίμηση λάθους για ένα κόμβο και έτσι το κλάδεμα γίνεται σε περισσότερους κόμβους. Η επίδραση της παραμέτρου αυτής είναι αρκετά μεγάλη και για μεγάλες τιμές συχνά καταλήγουμε σε δέντρο ενός φύλλου. Συνήθως λοιπόν το αφήνουμε 0 ή 0,1.

Επιλογή κανόνων

Επόμενη παράμετρος είναι το ExportRules όπου δηλώνουμε αν θέλουμε να εξάγουμε και σύνολα κανόνων ταξινόμησης εκτός από δέντρα. Ένα σύνολο κανόνων

προκύπτει από ένα δέντρο αν θεωρήσουμε κάθε μονοπάτι του δέντρου σαν έναν κανόνα. Οι κανόνες αυτοί έχουν την ίδια ακριβώς μορφή με αυτούς του CN2 με την διαφορά ότι αντί για εντροπία και σημαντικότητα αναφέρεται ο αριθμός των εγγραφών που καλύπτονται από τον κανόνων αυτόν και πόσες από αυτές ταξινομούνται λάθος. Υπάρχει επίσης επιλογή να ταξινομηθεί το σύνολο των κανόνων επιλέγοντας την επιλογή SortRules.Αυτή χρησιμοποιεί μια μετρική τιμή της οποίας αναφέρεται, που μετράει το πόσο σημαντικός είναι ο κανόνας και τον ταξινομεί ανάλογα.

Μετάφραση κανόνων

Μπορούμε ,εφόσον υπάρχει δυνατότητα, να μεταφράσουμε την έξοδο των κανόνων σε πιο κατανοήσιμη μορφή .Οι κανόνες ,εφόσον μεταφραστούν ,θα βγουν σε if-then μορφή και θα έχουν αντικατασταθεί οι τιμές των πεδίων με πιο ευκολονόητες τιμές από αυτές που υπάρχουν στα δεδομένα. Αυτό γίνεται επιλέγοντας το TranslateRules.Εννοείται ότι πρέπει να έχει φροντίσει ο administrator της βάσης να κατασκευάσει πίνακα μετάφρασης για το σύνολο των δεδομένων που χρησιμοποιήσαμε.

Παραγωγή Expert Output

Επιλέγοντας να παράγουμε έξοδο για expert system απλώς παράγουμε ένα επιπλέον αρχείο. Οι κανόνες μετασχηματίζονται και σε μια άλλη μορφή κατάλληλη για επεξεργασία από ένα system.

Σελίδες Results- Tree/Results- Rules

Η έξοδος βγαίνει λοιπόν σε δυο μορφές, σε δέντρα και σε σύνολα κανόνων. Η μορφή των κανόνων είναι αντιστοιχεί με αυτής των κανόνων του CN2 και αρκετά κατανοήσιμη. Πρέπει όμως να αναλύσουμε τον τρόπο με τον οποίο παρουσιάζονται τα δέντρα του C4.5 καθώς έχουν γίνει κάποιες συμβάσεις στην απεικόνιση τους.

Κάθε κόμβος περιγράφεται από τα στοιχεία του που χωρίζονται με το σύμβολο <<|>>. Το πρώτο στοιχείο είναι η διακριτή τιμή από το πεδίο εξειδίκευσης του πατέρα που οδηγεί σε αυτόν τον κόμβο. Προφανώς για την ρίζα του δέντρου το στοιχείο αυτό είναι κενό. Ακλουθούν τα στοιχεία Num και Err. Το Num είναι ο αριθμός των εγγραφών του συνόλου εκπαίδευσης που <<καλύπτονται>> από αυτόν τον κόμβο. Για την ρίζα π.χ., το Num είναι ο συνολικός αριθμός εγγραφών του συνόλου εκπαίδευσης. Το Err είναι ο αριθμός των εγγραφών που ταξινομούνται λάθος στον κόμβο σε σχέση με την επικρατούσα τιμή τάξης Class: που ακολουθεί. Το Err φυσικά έχει μεγαλύτερη σημασία για τα φύλλα του δέντρου, όπου καταλήγει η διαδικασία ταξινόμησης, για να δούμε την ακρίβεια με την οποία χωρίζονται τελικά οι εγγραφές. Το τελευταίο στοιχείο του κόμβου είναι το NextAttr όπου δηλώνετε το όνομα του πεδίου που χρησιμοποιείται για έλεγχο στον κόμβο αυτόν. Στα φύλλα όπου δεν γίνονται έλεγχοι η τιμή που δίνεται είναι <<_leaf>>.

Ας πάρουμε για παράδειγμα μια αταξινόμητη εγγραφή που θα ταξινομηθεί με χρήση του παραπάνω δέντρου. Ξεκινάμε από τη ρίζα και κοιτάμε την τιμή του NextAttr. Η τιμή είναι <<MHTRWO>> οπότε και ελέγχουμε την τιμή του πεδίου <<MHTRWO>> στην εγγραφή μας, π.χ. <<yes>>. Ψάχνουμε τα παιδιά της ρίζας μέχρι να βρούμε τον κόμβο με τιμή εξειδίκευσης <<=yes>>. Αυτός ο κόμβος είναι ο δεύτερος και εκεί οδηγούμαστε. Ελέγχουμε πάλι το στοιχείο NextAttr. Εφόσον είναι <<_leaf>> έχουμε φτάσει σε φύλλο οπότε και ταξινομούμε την εγγραφή στην τάξη που δηλώνει το στοιχείο Class του φύλλου (no).

Σελίδα ClassifyTable

Όπως και στον CN2, έτσι και στον C4.5 υπάρχει η σελίδα ClassifyTable για ταξινόμηση ενός άλλου πίνακα. Η λειτουργία της σελίδας είναι πανομοιότυπη με αυτή του CN2 (Μουστάκας, 2001).

Διεθνής βιβλιογραφία

[1] Acid S, De Campos LM. Searching for Bayesian Network Structures in the Space of Restricted Acyclic Partially Directed Graphs. Journal of Artificial Intelligence Research 2003; 18; 445-490.

[2] Aggarwal CC, Yu PS. Data Mining Techniques for Associations, Clustering and Classification. In: Proceedings of the 3rd Pacific-Asia Conference on Methodologies for Knowledge Discovery and Data Mining. 1999. p. 13-23.

[3] Agrawal R, Arning A, Bollinger T, Mehta M, Shafer J, Srikant R. The Quest Data Mining System In: Proceedings of the 2nd International Conference on Knowledge Discovery in Databases and Data Mining, Portland, Oregon, 1996.

[4] Aha DW. Lazy Learning. Dordrecht, Kluwer Academic Publishers, 1997.

[5] An A, Cercone N. Rule Quality Measures improve the Accuracy of Rule Induction: An experimental approach. Lecture Notes in Computer Science 2000, volume 1932, 119-129.

[6] Andrews R, Diederich J, Tickle AB. Survey and critique of techniques for extracting rules from trained artificial neural networks. Knowledge-Based Systems 1995; 8; 373-389.

[7] Bandyopadhyay S, Pal SK. Classification and Learning using Genetic Algorithms-Applications in bioinformatics and web intelligence. Springer, 2007.

[8] Batcher KE. Sorting networks and their applications. In: Proceedings of the AFIPS Spring Joint Computer Conference 32, 1968, p. 307–314.

[9] Boutsinas B, Antzoulatos G, Alevizos P. A novel classification algorithm based on clustering. In: 1st International Conference “From Scientific Computing to Computational Engineering”. Athens, Greece, 2004.

[10] Boutsinas B, Vrahatis MN. Artificial Nonmonotonic Neural Networks. Artificial Intelligence 2001; 132 (1); 1-38.

[11] Breinman L, Friedman JH, Olshen RA, Stone CJ. Classification and Regression Trees. Wadsworth and Brooks, Calif, 1984.

[12] Burges C. A tutorial on support vector machines for pattern recognition. Data

Mining and Knowledge Discovery 1998; 2(2); 1-47.
[13] Cabena P, Hadjinian P, Stadler R, Verchees J, Zanasi A. Discovering Data Mining: From Concept to Implementation. Prentice Hall, Upper Saddle River, NJ, 1998.
[14] Cheeseman P, Stutz J. Bayesian Classification (Auto Class): Theory and Results. In: Fayyad et al (Eds.). Advances in Knowledge Discovery and Data Mining. AAAI Press/MIT Press, 1996. p. 153-180.
[15] Cheng J, Greiner R, Kelly J, Bell D, Liu W. Learning Bayesian networks from data: An information-theory based approach. Artificial Intelligence 2002; 137; 43-90.
[16] Cheng J, Greiner R. Learning Bayesian Belief Network Classifiers: Algorithms and System. In: Stroulia E, Matwin S (Eds.), Advances in Artificial Intelligence 2001; volume 2056; 141-151.
[17] Chickering DM. Optimal Structure Identification with Greedy Search. Journal of Machine Learning Research 2002; 3; 507-554.
[18] Clark P, Niblett T. The CN2 Induction Algorithm. Machine Learning 1989; 3(4); 261-283.
[19] Cohen W. Fast Effective Rule Induction. In: Proceedings of 12th International Conference on Machine Learning, Tahoe City, California, USA. 1995. p. 115-123.
[20] Cost S, Salzberg S. A Weighted Nearest Neighbour Algorithm for Learning with Symbolic Features. Machine Learning 1993; 10; 57-78.
[21] Cover T, Hart P. Nearest neighbor pattern classification. IEEE Transactions on Information Theory 1967; 13(1); 21–27.
[22] Cristianini N, Shawe-Taylor J. An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods, Cambridge University Press, Cambridge, 2000.
[23] De Mantaras & Armengol E. Machine learning from examples: Inductive and Lazy methods. Data & Knowledge Engineering 1998; 25; 99-123.
[24] Domingos P, Pazzani M. On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning 1997; 29; 103-130.

[25] Dzeroski S. Inductive Logic Programming and Knowledge Discovery in Databases. In: Fayyad UM, Piatetsky-Shapiro G, Smyth P (Eds.). Advances in Knowledge Discovery and Data Mining, 1996. p. 117-152.
[26] Fellbaum C. WordNet: An Electronic Lexical Database. The MIT Press, 1998.

[27] Flach PA, Lavrac N. Rule Induction. In: Berthold M, Hand DJ (Eds.). Intelligent Data Analysis: An Introduction. Springer, 2003. p. 229-267.
[28] Floyd RW. Algorithm 245 - Treesort 3. Communications of the ACM 1964; 7(12); 701.
[29] Frank E, Witten I. Generating Accurate Rule Sets Without Global Optimization. In: Proceedings of the 15th International Conference on Machine Learning. San Francisco, California, USA. 1998. p. 144-151.
[30] Freitas AA. A survey of Evolutionary Algorithms for Data Mining and Knowledge Discovery. In: Ghosh A, Tsutsui S (Eds.). Advances in Evolutionary Computation. Springer Verlag, 2003. p. 819-845.
[31] Freund Y, Schapire RE. A short introduction to boosting. Journal of Japanese Society for Artificial Intelligence 1999; 14(5); 771-780.
[32] Friedman N, Geiger D, Goldsmidt M. Bayesian network classifiers. Machine Learning 1997; 29(2); 131-163.
[33] Furnkranz J. Separate-and-Conquer Rule Learning. Artificial Intelligence Review 1999; 13; 3-54.
[34] Furnkranz J. Pruning algorithms for rule learning. Machine Learning 1997; 27; 139-171.
[35] Gastwirth JL. The Estimation of the Lorenz Curve and Gini Index. The Review of Economics and Statistics 1972, 54, 306-316.
[36] Gehrke J, Ramakrishnan R, Ganti V. Rainforest – A framework for fast decision tree construction of large datasets. Journal of Data Mining and Knowledge Discovery 2000, 4(2), 127-162.
[37] Hand D, Mannila H, Smyth P. Principles of Data Mining, MIT Press, Cambridge, MA, 2001.
[38] Heckerman D, Meek C, Cooper G. A Bayesian Approach to Causal

Discovery. In: Glymour C, Cooper G (Eds.). Computation, Causation, and Discovery. MIT Press, 1999. p. 141-165.

[39] Hunt EB, Martin J, Stone PJ. Experiments in Induction. Academic Press, New York, 1966.

[40] Jain AK, Dubes RC. Algorithms for Clustering Data. Prentice-Hall, New Jersey, 1988.

[41] Jensen FV. Introduction to Bayesian networks. Springer Verlag, New York, 1996.

[42] Kalbfleish J. Probability and statistical inference (volume 2), Springer Verlag, New York, 1979.

[43] Kotsiantis SB. Supervised Machine Learning: A Review of Classification Techniques. Informatica 2007; 31; 249-268.

[44] Kubat M, Cooperson M. A reduction technique for nearest-neighbor classification: Small groups of examples. Intelligent Data Analysis 2001; 5(6); 463-476.

[45] Lange K. Computational Statistics and Optimization Theory at UCLA. The American Statistician 2004; 58; 9–11.

[46] Larose DT. Data Mining: Methods and Models. Wiley, New York, 2006.

[47] Larose DT. Discovering Knowledge in Data: An Introduction to Data Mining. Wiley, New York, 2004.

[48] Lavrac N, Flach P, Zupan B. Rule Evaluation Measures: A Unifying View. In: Proceedings of the 9th International Workshop on Inductive Logic Programming ILP'99, Bled, Slovenia, 1999. p. 174-185.

[49] Mehta M, Agrawal R, Rissanen J. SLIQ: A fast scalable classifier for data mining. In: Proceedings of the 5th International Conference of Extending Database Technology, Avignon, France, 1996.

[50] Michalski RS, Chilausky RL. Learning by being told and learning from examples: an experimental comparison of the two methods of knowledge acquisition in the context of developing and expert system for soybean disease diagnosis. Policy Analysis and Information Systems 1980; 4(2); 125-160.

[51] Michie D, Spiegelhalter DJ, Taylor CC, Campbell J. Machine Learning, neural

and statistical classification. Ellis Horwood Upper Saddle River, NJ, USA, 1995.

[52] Mitchell T. Machine Learning, McGraw-Hill, 1997.

[53] Muggleton S. Inductive Logic Programming, 38 of A.P.I.C. series. Academic Press, London, 1992.

[54] Murthy T. Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey. Data Mining and Knowledge Discovery 1998; 2; 345–389.

[55] Neocleous C, Schizas C. Artificial Neural Network Learning: A Comparative Review, Lecture Notes in Artificial Intelligence 2308, Springer-Verlag Berlin Heidelberg. 2002. p. 300-313.

[56] Okamoto S, Yugami N. Effects of domain characteristics on instance-based learning algorithms. Theoretical Computer Science 2003; 298; 207-233.

[57] Parlante N. Binary trees. Stanford CS Education Library, 2000. Ανάκτηση από την ιστοσελίδα http://cslibrary.stanford.edu/110/BinaryTrees.pdf στις 3/1/2010.

[58] Piatetsky-Shapiro G, Frawley WJ. Knowledge Discovery in Databases, AAAI/MIT Press, 1991.

[59] Piatetsky-Shapiro G. Knowledge discovery in real databases: a report on the IJCAI-89 workshop. AI Magazine 1990; 11(5); 68-70.

[60] Quinlan JR. C4.5: Programs for Machine Learning. Morgan Kaufmann, CA, 1993.

[61] Quinlan JR. Generating Production Rules from Decision Trees. In: Proceedings of the 10th International Joint Conference on Artificial Intelligence, Milan, Italy. 1987. p. 304-307.

[62] Quinlan JR. Induction of Decision Trees. Machine Learning 1986; 1; 85-106.

[63] Rivest RL. Learning decision lists. Machine Learning 1987; 2; 229-246.

[64] Rumelhart DE, Hinton GE, Williams RJ. Learning internal representations by error propagation. In: Rumelhart DE, McClelland JL (Eds.). Parallel Distributed Processing: Explorations in the Microstructure of Cognition. MIT Press, 1986. p. 318-363.

[65] Sanchez J, Barandela R, Ferri F. On Filtering the Training Prototypes in

Nearest Neighbor Classification, Lecture Notes in Computer Science, 2002; 2504; 239-248.

[66] Sedgewick R. Algorithms in C++, Parts 1-4: Fundamentals, Data Structure, Sorting, Searching, 3rd edition. Addison-Wesley Longman, 1998, p. 273–274.

[67] Shafer J, Agrawal R, Mehta M. SPRINT: A scalable parallel classifier for data mining. In: Proceedings of the 22th International Conference of Very Large Databases, Bombay, India, 1996. p. 544-555.

[68] Shin KS, Lee TS, Kim HJ. An application of support vector machines in bankruptcy prediction model. Expert Systems with applications 2005; 28(1); 127-135.

[69] Siddique MNH, Tokhi MO. Training Neural Networks: Back-propagation vs. Genetic Algorithms, IEEE International Joint Conference on Neural Networks 2001; 4; 2673-2678.

[70] Siegler RS. Three aspects of cognitive development. Cognitive Psychology 1976; 8; 481-520.

[71] Titterington DM, Smith AFM, Makov UE. Statistical Analysis of finite mixture distributions. Wiley, New York, 1985.

[72] Towell GG, Shavlik JW. Knowledge based artificial neural networks. Artificial Intelligence 1994; 40; 119-165.

[73] Tsumoto S. Characteristics of Accuracy and Coverage in Rule Induction. Lecture Notes in Computer Science 2003; 237-244.

[74] Vapnik VN. Statistical Learning Theory, Wiley, New York, 1998.

[75] Vapnik VN. The Nature of Statistical Learning Theory. Springer, New York, 1995.

[76] Vivarelli F, Williams C. Comparing Bayesian neural network algorithms for classifying segmented outdoor images. Neural Networks 2001; 14; 427-437.

[77] Vovk V, Gammerman A, Shafer G. Algorithmic Learning in a Random World. Springer, New York, 2005.

[78] Weigend AS, Rumelhart DE, Huberman BA. Generalization by weightelimination with application to forecasting. In: Proceedings of the Conference on Neural Information Processing Systems 3, San Mateo, California, USA.

1990. p. 875-882.
[79] Wettschereck D, Aha DW, Mohri T. A Review and Empirical Evaluation of Feature Weighting Methods for a Class of Lazy Learning Algorithms. Artificial Intelligence Review 1997; 10; 1–37.
[80] Wirth N. Algorithms and Data Structures. Prentice-Hall, Inc., 1985.
[81] Wu X, Kumar V, Quinlan R, Ghosh J, Yang Q, Motoda H, McLachlan G, Ng A, Liu B, Yu P et al. Top 10 algorithms in data mining. Knowledge Information Systems 2008; 14(1); 1-137.
[82] Yam J, Chow W. Feed-forward Networks Training Speed Enhancement by Optimal Initialization of the Synaptic Coefficients. IEEE Transactions on Neural Networks 2001; 12; 430-434.
[83] Zhang H. The optimality of Naïve Bayes. In: Proceedings of the 17th International Florida Artificial Intelligence Research Society Conference, Miami Beach, Florida, USA, 2004. p. 562-567.
[84] Zhang G. Neural networks for classification: a survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C 2000; 30(4); 451-462.
[85] Zhou Z. Rule Extraction: Using Neural Networks or For Neural Networks Journal of Computer Science and Technology 2004; 19(2); 249-253.

Ελληνική βιβλιογραφία

[86] Rawlings G. Αλγόριθμοι - Ανάλυση και Σύγκριση. Εκδόσεις Κριτική, 2004.
[87] Αδαμίδης Π. Εισαγωγή στη πολυπλοκότητα - Αλγόριθμοι ταξινόμησης, Πολυπλοκότητα - Εισαγωγή στους Αλγόριθμους ταξινόμησης. Σημειώσεις μαθήματος: Προγραμματισμός Η/Υ ΙΙ. Τμήμα Πληροφορικής, ΑΤΕΙ Θεσσαλονίκης, 2009.
[88] Βαζιργιάννης Μ, Χαλκίδη Μ. Εξόρυξη Γνώσης από Βάσεις Δεδομένων. Τυπωθήτω – Γιώργος Δάρδανος, Αθήνα, 2003.

[89] Μανωλόπουλος Γ. Ανάλυση αλγορίθμων, Σημειώσεις μαθήματος. Τμήμα Πληροφορικής, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης. Ανάκτηση από την ιστοσελίδα http://delab.csd.auth.gr/~manolopo/Analysis/cs_algorithms.htm στις 10/2/2009.

[90] Μαστρογιάννης Ν. Μεθοδολογικό Πλαίσιο Υποστήριξης της Εξόρυξης Γνώσης από Δεδομένα με τη χρήση αρχών της Πολυκριτήριας Ανάλυσης Αποφάσεων. Τμήμα Διοίκησης Επιχειρήσεων, Πανεπιστήμιο Πατρών, 2009.

[91] Μουστάκας Γ. ΔΙΟΓΕΝΗΣ: Σύστημα Ιεραρχικά Κατανεμημένης Εξόρυξης Λειτουργικών Επιχειρηματικών Δεδομένων. Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Πανεπιστήμιο Πατρών, 2001.

[92] Ρίζος Γ. Τεχνητά Νευρωνικά δίκτυα: Θεωρία και Εφαρμογές. Εκδόσεις Νέων Τεχνολογιών, Αθήνα, 1996.