



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Αυτόματος Προσδιορισμός Κανόνων Ασαφούς Λογικής για
Ευφυείς Πράκτορες που Συμμετέχουν σε Ηλεκτρονικές
Αγορές.**

Καλλιρόη Χ. Αράπογλου

**Επιβλέποντες: Ευστάθιος Χατζηευθυμιάδης, Επίκουρος Καθηγητής ΕΚΠΑ
Κωνσταντίνος Κολομβάτσος, Υποψήφιος Διδάκτωρ ΕΚΠΑ**

ΑΘΗΝΑ

Οκτώβριος 2009

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Αυτόματος Προσδιορισμός Κανόνων Ασαφούς Λογικής για Ευφρείς Πράκτορες που Συμμετέχουν σε Ηλεκτρονικές Αγορές.

Καλλιρόη Χ. Αράπογλου

A.M.:1115 2004 00011

ΕΠΙΒΛΕΠΟΝΤΕΣ:

**Ευστάθιος Χατζηευθυμιάδης, Επίκουρος Καθηγητής ΕΚΠΑ
Κωνσταντίνος Κολομβάτσος, Υποψήφιος Διδάκτωρ ΕΚΠΑ**

ΠΕΡΙΕΧΟΜΕΝΑ

| | |
|-------------------------------------------------------------------------|-----------|
| ΠΕΡΙΕΧΟΜΕΝΑ | 3 |
| ΚΕΦΑΛΑΙΟ 1 | 5 |
| ΕΙΣΑΓΩΓΗ | 5 |
| 1.1 ΠΕΡΙΓΡΑΦΗ..... | 5 |
| 1.2 ΚΙΝΗΤΡΟ | 6 |
| 1.3 ΕΠΙΔΙΩΚΟΜΕΝΟΙ ΣΤΟΧΟΙ..... | 6 |
| 1.4 ΕΠΙΛΟΓΟΣ..... | 7 |
| ΚΕΦΑΛΑΙΟ 2 | 8 |
| ΗΛΕΚΤΡΟΝΙΚΕΣ ΑΓΟΡΕΣ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΕΞΥΠΝΟΙ ΠΡΑΚΤΟΡΕΣ..... | 8 |
| 2.1 ΓΕΝΙΚΗ ΠΕΡΙΓΡΑΦΗ | 8 |
| 2.2 ΑΡΧΙΤΕΚΤΟΝΙΚΗ | 12 |
| 2.3 ΕΞΥΠΝΟΙ ΠΡΑΚΤΟΡΕΣ ΛΟΓΙΣΜΙΚΟΥ | 13 |
| 2.5 ΕΠΙΛΟΓΟΣ..... | 15 |
| ΚΕΦΑΛΑΙΟ 3 | 16 |
| ΑΣΑΦΗΣ ΛΟΓΙΚΗ | 16 |
| 3.1 ΓΕΝΙΚΗ ΠΕΡΙΓΡΑΦΗ | 16 |
| 3.2 ΑΣΑΦΕΙΣ ΚΑΝΟΝΕΣ..... | 18 |
| 3.3 ΑΣΑΦΗΣ ΕΞΑΓΩΓΗ ΣΥΜΠΕΡΑΣΜΑΤΩΝ | 19 |
| 3.4 ΔΟΜΗ ΠΡΑΚΤΟΡΑ..... | 22 |
| 3.5 ΕΠΙΛΟΓΟΣ..... | 24 |
| ΚΕΦΑΛΑΙΟ 4 | 25 |
| ΤΕΧΝΙΚΕΣ ΑΥΤΟΜΑΤΗΣ ΕΞΟΡΥΞΗΣ ΒΑΣΗΣ ΚΑΝΟΝΩΝ..... | 25 |
| 4.1 ΤΕΧΝΙΚΕΣ | 25 |
| 4.1.1 ΓΕΝΕΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ..... | 25 |
| 4.1.2 ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ | 26 |
| 4.1.3 ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ | 28 |
| 4.1.4 ΕΞΑΓΩΓΗ ΑΣΑΦΩΝ ΚΑΝΟΝΩΝ ΑΠΟ ΔΕΔΟΜΕΝΑ..... | 28 |
| 4.2 ΣΥΣΤΑΔΟΠΟΙΗΣΗ | 28 |
| 4.3 ΑΛΓΟΡΙΘΜΟΙ ΣΥΣΤΑΔΟΠΟΙΗΣΗΣ..... | 30 |
| 4.3.1 K-means ή HCM (Hard C-means)..... | 31 |
| 4.3.2 FCM (Fuzzy C-means) | 33 |
| 4.3.3 Subtractive clustering..... | 35 |
| 4.3.4 Agglomerative | 38 |
| 4.4 ΠΛΗΘΟΣ CLUSTERS (optimal clustering) | 42 |
| 4.5 ΠΑΡΑΤΗΡΗΣΕΙΣ | 44 |
| ΚΕΦΑΛΑΙΟ 5 | 46 |
| ΕΦΑΡΜΟΓΗ ΜΕΘΟΔΟΛΟΓΙΑΣ ΑΥΤΟΜΑΤΗΣ ΕΞΟΡΥΞΗΣ ΣΕ ΗΛΕΚΤΡΟΝΙΚΕΣ ΑΓΟΡΕΣ..... | 46 |
| 5.1 ΓΕΝΙΚΗ ΠΕΡΙΓΡΑΦΗ | 46 |
| 5.2 ΕΞΑΓΩΓΗ ΑΣΑΦΩΝ ΚΑΝΟΝΩΝ ΑΠΟ CLUSTERS..... | 46 |
| 5.2.1 FCM | 46 |
| 5.2.2 Subtractive | 49 |
| 5.2.3 HCM και Agglomerative..... | 50 |
| 5.2.4 Nearest Neighborhood Clustering (NNC) | 51 |
| 5.3 ΕΞΑΓΩΓΗ ΑΣΑΦΩΝ ΚΑΝΟΝΩΝ ΑΠΟ ΔΕΔΟΜΕΝΑ | 53 |
| 5.3.1 Learning from Examples (LFE) | 53 |

| | |
|---------------------------------------------------|-----------|
| 5.3.2 Modified Learning from Examples (MLFE)..... | 54 |
| 5.4 ΕΠΙΛΟΓΟΣ..... | 55 |
| ΚΕΦΑΛΑΙΟ 6 | 56 |
| ΑΠΟΤΕΛΕΣΜΑΤΑ..... | 56 |
| 6.1 ΓΕΝΙΚΗ ΠΕΡΙΓΡΑΦΗ..... | 56 |
| 6.2 ΣΥΓΚΡΙΤΙΚΗ ΑΠΟΤΙΜΗΣΗ ΑΛΓΟΡΙΘΜΩΝ | 59 |
| 6.3 ΣΥΜΠΕΡΑΣΜΑΤΑ..... | 69 |
| ΑΝΑΦΟΡΕΣ | 71 |
| ΠΑΡΑΡΤΗΜΑ Α..... | 72 |

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

Ξεκινώντας την μελέτη της εργασίας αυτής θα πρέπει να ορίσουμε το πρόβλημα που εξετάζουμε. Στις ενότητες που ακολουθούν θα περιγράψουμε σε γενικές γραμμές το μοντέλο το οποίο μελετήσαμε και αναπτύξαμε. Στη συνέχεια, θα παραθέσουμε τους λόγους για τους οποίους έγινε αυτή η εργασία καθώς και ποιοι είναι οι στόχοι της. Τέλος, θα γίνει μια απαρίθμηση των κεφαλαίων που ακολουθούν ώστε να αποκτήσει ο αναγνώστης μια γενική εικόνα.

1.1 ΠΕΡΙΓΡΑΦΗ

Η σημερινή εποχή χαρακτηρίζεται από πολλούς, και όχι άδικα, ως η εποχή της πληροφορίας. Με το ολοένα και πιο χαμηλό κόστος και τη συνεχή βελτίωση των μέσων αποθήκευσης, τα δεδομένα είναι σε αφθονία. Επιπρόσθετα, η ραγδαία εξάπλωση του Web συμβάλει ώστε κάθε απομακρυσμένος χρήστης να είναι πλέον σε θέση να έχει πρόσβαση σε αυτά. Τα δεδομένα είναι πλέον αγαθό και η τάση της εποχής αυτής έχει επηρεάσει και τις συνήθειες από τα παλιά!

Για παράδειγμα, το Web έχει αλλάξει ακόμη και την συμπεριφορά των χρηστών ακόμη και ως καταναλωτές μια και τώρα ένας τεράστιος αριθμός καταστημάτων μπορούν να προσφέρουν τα προϊόντα τους μέσω του παγκόσμιου ιστού. Μια ηλεκτρονική αγορά αποτελεί ένα σύστημα στο οποίο πολλά πρόσωπα επικοινωνούν και κάνουν συναλλαγές με στόχο την ανταλλαγή ενός αριθμού προϊόντων. Τα πρόσωπα αυτά ανήκουν σε δυο κατηγορίες, τους αγοραστές και τους πωλητές. Αυτά για τα οποία συναλλάσσονται είναι τα αγαθά. Η τιμή ενός αγαθού καθορίζεται μέσα από το νόμο της προσφοράς και της ζήτησης [18].

Τώρα πλέον έχει κάνει την είσοδο της η έννοια της «αγοράς πληροφορίας» (information market). Η έννοια αυτή αποτελεί συνδυασμό τόσο του συστήματος της ηλεκτρονικής αγοράς όσο και της τεχνολογικής εξέλιξης. Τις αγορές πληροφορίας θα μελετήσουμε και εμείς. Οι κύριες οντότητες που εμφανίζονται σε αυτές είναι πάλι οι αγοραστές και οι πωλητές. Στο Κεφάλαιο 2 θα παρουσιάσουμε εκτενώς την αρχιτεκτονική τους. Τα επόμενα κεφάλαια αποτελούν και τα στάδια που θα περάσουμε για να ορίσουμε και να υλοποιήσουμε ένα σενάριο που θα αφορά σε αγορές πληροφορίας.

1.2 ΚΙΝΗΤΡΟ

Το κύριο πρόβλημα που αντιμετωπίζουμε στις αγορές πληροφορίας είναι το γεγονός πως δεν είναι τόσο εύκολη η αναζήτηση και η απόκτηση των κατάλληλων πληροφοριών από τους ίδιους τους χρήστες. Οι χρήστες θα πρέπει να καταναλώσουν πολύ χρόνο στην αναζήτηση και εύρεση του κατάλληλου προϊόντος πληροφορίας μέσα σε εκατομμύρια πηγές. Η χρησιμοποίηση αυτόνομων οντοτήτων για την αναζήτηση και την εύρεση τέτοιου είδους αγαθών φαντάζει να είναι πιο αποδοτική. Όμως, αυτές οι αυτόνομες οντότητες θα πρέπει με κάποιο τρόπο να μπορούν να λειτουργούν σαν να ήταν οι ίδιοι οι χρήστες ώστε να έχουν την υψηλότερη δυνατή απόδοση. Οι έξυπνοι πράκτορες λογισμικού μπορούν να προσφέρουν πολλά πλεονεκτήματα σε ένα τέτοιο σενάριο. Επίσης, μια βάση γνώσης θα μπορούσε να προσφέρει τον κατάλληλο μηχανισμό συμπεριφοράς τέτοιων οντοτήτων. Αυτή η βάση γνώσης μπορεί να περιέχει την αντίδραση που πρέπει η αυτόνομη οντότητα να έχει σε διάφορες καταστάσεις που μπορεί να αντιμετωπίσει σε ένα δυναμικό περιβάλλον μιας ηλεκτρονικής αγοράς. Η θεωρία της ασαφούς λογικής (fuzzy logic) αποτελεί ένα εργαλείο με το οποίο μπορεί κάποια αυτόνομη οντότητα να χτίσει μια βάση γνώσης και να παίρνει αποφάσεις. Το γεγονός ότι η ασαφής λογική μπορεί να δώσει αποτελέσματα μέσα από ελλιπή δεδομένα, αποτελεί έναν ακόμη λόγο για να τη μελετήσουμε και να την χρησιμοποιήσουμε στο σενάριο μας. Η ασαφής λογική και το πώς μπορεί να εφαρμοστεί σε ένα αγοραστή ή πωλητή είναι και το κίνητρο αυτής της εργασίας.

1.3 ΕΠΙΔΙΩΚΟΜΕΝΟΙ ΣΤΟΧΟΙ

Οι στόχοι στους οποίους προσβλέπουμε μέσα από την παρούσα πτυχιακή εργασία μπορούν να συνοψιστούν στα επόμενα:

- Να ορίσουμε ένα δικό μας πρόβλημα-σενάριο όπου θα εμπλέκονται αγοραστές και πωλητές.
- Να μελετήσουμε το θεωρητικό υπόβαθρο για το μοντέλο που θα υλοποιήσουμε.
- Να υλοποιήσουμε αλγορίθμους δημιουργίας βάσης γνώσης για τις οντότητες μας.
- Να προσομοιώσουμε μια εικονική αγορά με τις υλοποιήσεις που κάναμε.
- Να συγκρίνουμε τα αποτελέσματα των διαφόρων αλγορίθμων και να βγάλουμε συμπεράσματα για την απόδοση κάθε αλγορίθμου και τους περιορισμούς που μπορεί να έχει.

1.4 ΕΠΙΛΟΓΟΣ

Σαν επίλογο σε αυτό το εισαγωγικό κεφάλαιο θα κάνουμε μια σύντομη παρουσίαση των όσων θα ακολουθήσουν. Στο Κεφάλαιο 2 θα περιγράψουμε αναλυτικότερα ποια είναι η δομή μιας ηλεκτρονικής αγοράς πληροφορίας και θα εισάγουμε την έννοια του έξυπνου πράκτορα. Στο Κεφάλαιο 3 θα ορίσουμε βασικές έννοιες πάνω στην ασαφή λογική και τα ασαφή συστήματα. Την ασαφή λογική θα χρησιμοποιήσουν και οι πράκτορες μας ώστε να χτίσουν την βάση γνώσης στην οποία θα βασίζονται οι εκάστοτε αποφάσεις. Στο Κεφάλαιο 4 θα γίνει μια παρουσίαση των διαφόρων τεχνικών αυτόματης εξαγωγής ασαφών κανόνων. Θα εστιάσουμε σε συγκεκριμένες τεχνικές και θα κάνουμε εκτενή ανάλυση των αλγορίθμων που συναντάμε εκεί. Η ενότητα αυτή αποτελεί ένα από τα δυο κομμάτια στην υλοποίηση των πρακτόρων μας. Το δεύτερο κομμάτι βρίσκεται στο Κεφάλαιο 5. Εκεί θα δούμε πως χρησιμοποιούνται οι αλγόριθμοι αυτοί για να παράγουμε ασαφής κανόνες. Με την ολοκλήρωση αυτών των δυο κεφαλαίων θα έχουμε προχωρήσει και στην υλοποίηση και δοκιμή του λογισμικού για τον πράκτορα μας. Στο Κεφάλαιο 6 θα ορίσουμε και θα εκτελέσουμε το σενάριο μας με πολλές παραλλαγές. Με διαφορετικές εκτελέσεις του ίδιου σεναρίου, προσπαθούμε να βγάλουμε συμπεράσματα και παρατηρήσεις για τις τεχνικές που μελετήσαμε στα Κεφάλαια 4 και 5. Οι κύριοι άξονες των συμπερασμάτων μας θα αφορούν στην πολυπλοκότητα, την ευελιξία και τα αδύνατα σημεία και τους περιορισμούς των αλγορίθμων.

ΚΕΦΑΛΑΙΟ 2

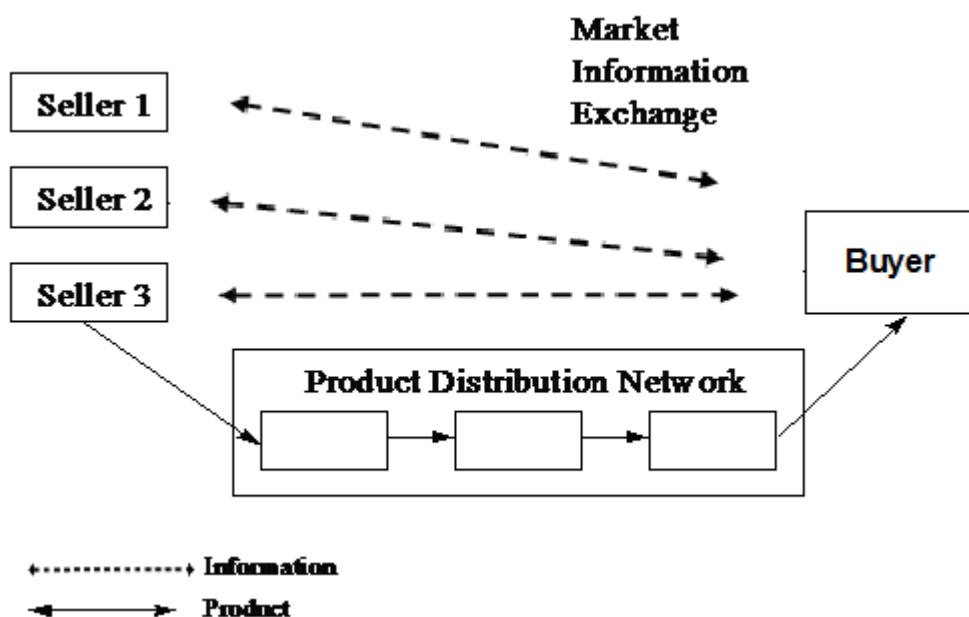
ΗΛΕΚΤΡΟΝΙΚΕΣ ΑΓΟΡΕΣ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΕΞΥΠΝΟΙ ΠΡΑΚΤΟΡΕΣ

Στο κεφάλαιο αυτό θα παρουσιάσουμε αναλυτικότερα την αρχιτεκτονική και τα χαρακτηριστικά των αγορών πληροφορίας, καθώς και των οντοτήτων που συμμετέχουν σε αυτές.

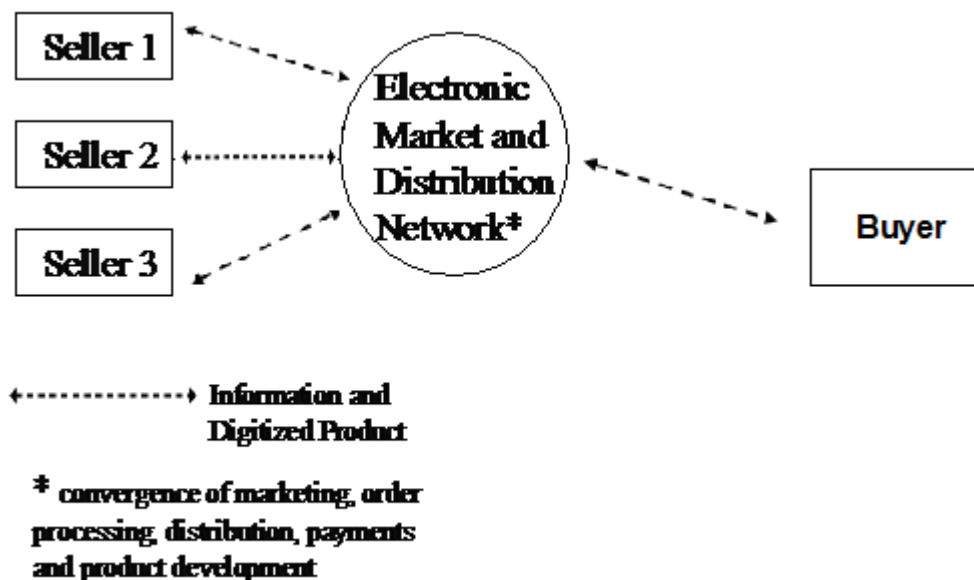
2.1 ΓΕΝΙΚΗ ΠΕΡΙΓΡΑΦΗ

Οι υπηρεσίες του Web εξελίσσονται με ταχείς ρυθμούς. Στο πρόσφατο παρελθόν ήταν δύσκολο για κάποιον χρήστη να χρησιμοποιήσει το διαδίκτυο για να κάνει αγοραπωλησίες. Υπήρχε μεγάλη επιφύλαξη στο να στείλει κάποιος τα προσωπικά του στοιχεία μέσω του Web. Το κλίμα αυτό έχει αλλάξει και το ηλεκτρονικό εμπόριο κατέχει πλέον εξέχουσα θέση και βρίσκει μεγάλη εφαρμογή καθημερινά.

Οι ηλεκτρονικές αγορές πληροφορίας αποτελούν ένα σύστημα που περιλαμβάνει τις λειτουργίες της διαφήμισης, της παραγγελίας, της παραλαβής και της πληρωμής προϊόντων. Δεν διαφέρουν πολύ από τις παραδοσιακές αγορές τις οποίες γνωρίζουμε. Η έννοια της αγοραπωλησίας μεταξύ δυο μερών (αγοραστή-πωλητή/buyer-seller) υφίσταται και στις δυο. Στις εικόνες που ακολουθούν (Εικόνα 1 και 2) παρουσιάζεται σχηματικά η λειτουργία τόσο μιας παραδοσιακής όσο και μιας ηλεκτρονικής αγοράς.



Εικόνα 1. Σχηματική απεικόνιση μιας παραδοσιακής αγοράς.



Εικόνα 2. Σχηματική απεικόνιση μιας ηλεκτρονικής αγοράς.

Στην παραδοσιακή αγορά (βλέπε Εικόνα 1) ένας αγοραστής (buyer) πρέπει να ψάξει το προϊόν που θέλει. Αποκτά πληροφορίες (information) γι' αυτό όπως η τιμή, η ποιότητα και λοιπά χαρακτηριστικά. Αφού ψάξει ανάμεσα σε αρκετούς πωλητές (sellers) αποφασίζει να αγοράσει. Η αγοραπωλησία και η μεταφορά του προϊόντος είτε γίνεται από τον ίδιο τον αγοραστή ή από ένα δίκτυο εξυπηρέτησης (product distribution network) που συνήθως είναι μια μεταφορική εταιρεία.

Από την άλλη, μια ηλεκτρονική αγορά (βλέπε Εικόνα 2) μπορεί να συσταθεί μέσα στο Web και να αναπτύξει όλες τις λειτουργίες της εκεί (αναζήτηση, πληρωμή, μεταφορά). Ο αγοραστής μπορεί ευκολότερα να βρει και να συγκρίνει περισσότερα προϊόντα, μιας και δεν χρειάζεται να κινηθεί. Ακόμη τις πληρωμές μπορεί να τις κάνει ηλεκτρονικά, για παράδειγμα μέσω πιστωτικής κάρτας, μιας και υπάρχουν μηχανισμοί κρυπτογραφίας που μπορούν να εγγυηθούν την ασφάλεια της συναλλαγής. Μερικά ακόμη χαρακτηριστικά των ηλεκτρονικών αγορών που δεν φαίνονται στο σχήμα είναι ότι σε μια δεδομένη αγορά δεν είναι δεσμευτικό για κάποιον αγοραστή ή πωλητή να μείνει μόνιμα εκεί. Ακόμη, η τιμή ενός προϊόντος σε μια αγοραπωλησία δεν είναι σταθερή αλλά μπορεί να αλλάζει ανάλογα με τις παραδοχές που κάνουμε στο σενάριο μας. Συμπεραίνουμε λοιπόν πως οι αγορές πληροφορίας χαρακτηρίζονται από μια δυναμικότητα.

Στους δυο επόμενους πίνακες (Πίνακας 1 και 2) εξετάζουμε από την σκοπιά του αγοραστή και του πωλητή τα οφέλη και τις αδυναμίες των παραδοσιακών και ηλεκτρονικών αγορών.

Πίνακας 1. Σύγκριση της παραδοσιακής και ηλεκτρονικής αγοράς πληροφορίας από την πλευρά του αγοραστή.

| | Παραδοσιακή αγορά | Ηλεκτρονική αγορά |
|-------------------|-------------------|-------------------|
| Τιμή αγαθού | Υψηλό | Χαμηλό |
| Κόστος αναζήτησης | Υψηλό | Χαμηλό |
| Ρίσκο | Χαμηλό | Υψηλό |
| Κόστος παραλαβής | Χαμηλό | Χαμηλό |
| Φόροι | Υψηλό | Χαμηλό |
| Κόστος της αγοράς | Χαμηλό | Υψηλό |

Παρατηρώντας τον Πίνακα 1 στις ηλεκτρονικές αγορές ένας αγοραστής μπορεί να ψάξει αρκετά για να βρει αυτό που θέλει σε χαμηλή τιμή. Αντίθετα στις παραδοσιακές αγορές δεν θα μπορούσε να αφιερώσει τόσο χρόνο και θα αρκούνταν στην πρώτη ή τη δεύτερη προσφορά. Μια ακόμη παράμετρος που συμβάλει στην αύξηση της τιμής του αγαθού είναι η ύπαρξη μεσάζοντα. Στις παραδοσιακές αγορές αυτό είναι σύνηθες μιας και κανείς δεν αγοράζει χονδρικής. Το κόστος αναζήτησης στις παραδοσιακές αγορές δεν είναι τίποτα άλλο από τα έξοδα μετακίνησης του αγοραστή σε διαφορετικές τοποθεσίες. Στις ηλεκτρονικές αγορές αυτό το κόστος δεν υφίσταται! Το ρίσκο αποτελεί τον κίνδυνο που υπάρχει να παραπλανηθεί ο αγοραστής από αναξιόπιστους πωλητές και να χάσει τα χρήματα του ή να υποκλέψει κάποιος τα προσωπικά του στοιχεία. Το κόστος παραλαβής στις παραδοσιακές αγορές είναι ελάχιστο έως μηδαμινό, όπως και στις ηλεκτρονικές αγορές όπου ο χρήστης απλά κάνει download. Το κόστος της αγοράς αφορά στην πληρωμή συμμετοχής σε μια αγορά, υπάρχει στις ηλεκτρονικές αγορές όταν κάποιος πρέπει να εγγραφεί σε αυτές.

Πίνακας 2. Σύγκριση της παραδοσιακής και ηλεκτρονικής αγοράς πληροφορίας από την πλευρά του πωλητή.

| | Παραδοσιακή αγορά | Ηλεκτρονική αγορά |
|------------------|-------------------|-------------------|
| Κόστος προώθησης | Υψηλό | Χαμηλό |
| Πάγια έξοδα | Υψηλό | Χαμηλό |
| Κόστος παραγωγής | Υψηλό | Χαμηλό |
| Κόστος παραλαβής | Χαμηλό | Χαμηλό |

Από την άλλη, στον Πίνακα 2 βλέπουμε ότι στις παραδοσιακές αγορές ο πωλητής προκειμένου να κάνει γνωστά τα προϊόντα του πρέπει να δαπανήσει αρκετά χρήματα για διαφήμιση. Αντιθέτως, στο Web η διαφήμιση δεν είναι τόσο ακριβή, όπως πχ στην τηλεόραση. Ένας πωλητής που δραστηριοποιείται σε μια παραδοσιακή αγορά, έχει κάποια πάγια έξοδα που αφορούν το κατάστημα του (μισθοί, ενοίκιο, λογαριασμοί). Αντίθετα, στις ηλεκτρονικές αγορές το «κατάστημα» είναι κάποιο site, όπου εκεί γίνονται και οι παραγγελίες των πελατών. Τα προϊόντα των ηλεκτρονικών αγορών είναι ψηφιακά, άρα η παραγωγή τους γίνεται από πολλές αντιγραφές. Αντίθετα, τα προϊόντα των παραδοσιακών αγορών έχουν κόστος παραγωγής όπως πρώτες ύλες, εργαστήρια κλπ. Το κόστος παραλαβής είναι το ίδιο με το «κόστος της αγοράς» που αναφέραμε πιο πάνω στους αγοραστές.

Σε καμία περίπτωση δεν μπορούμε να πούμε πως οι ηλεκτρονικές αγορές πληροφορίας πρέπει να αντικαταστήσουν τις παραδοσιακές. Η κύρια διαφορά μεταξύ τους είναι ότι στις μεν πρώτες τα αγαθά είναι ψηφιακά αρχεία. Συμφέρει πιο πολύ να γίνεται για παράδειγμα η αγορά «ψηφιακών βιβλίων» από ηλεκτρονικές αγορές από ότι να το αγοράσουμε από ένα κατάστημα. Όμως δεν ισχύει το ίδιο και για την αγορά ενός προϊόντος που δεν μπορεί να αποκτήσει ηλεκτρονική μορφή. Εδώ είναι μόνη λύση οι παραδοσιακές αγορές. Στην επόμενη ενότητα θα εξετάσουμε τις ηλεκτρονικές αγορές όσον αφορά τα πρόσωπα τα οποία συμμετέχουν σε αυτές και το τι λειτουργίες επιτελούν.

2.2 ΑΡΧΙΤΕΚΤΟΝΙΚΗ

Στις ηλεκτρονικές αγορές συναντάμε τρεις βασικές οντότητες. Τους αγοραστές, τους πωλητές και κάποιους μεσολαβητές. Για κάθε μια από αυτές τις οντότητες μπορεί να υπάρχει ένας ή περισσότεροι πράκτορες που υλοποιούν τις προδιαγραφές που αναφέρουμε παρακάτω. Στην επόμενη ενότητα θα ορίσουμε την έννοια του πράκτορα.

Οι αγοραστές

Μπαίνουν σε μια αγορά με στόχο να αποκτήσουν ένα προϊόν σε συμφέρουσα τιμή. Με βάση τη γνώση και τις απαιτήσεις που τους έχει δώσει ο χρήστης, λαμβάνουν απόφαση για το αν θα κλείσει μια συμφωνία. Πολλές φορές καθοριστικό ρόλο παίζουν και χρονικοί περιορισμοί για την λήψη αποφάσεων. Το δικό μας πρότυπο αγοραστή θα περιγράψουμε και θα υλοποιήσουμε στα επόμενα κεφάλαια.

Οι πωλητές

Παρέχουν προϊόντα πληροφορίας στους επίδοξους αγοραστές. Η τιμή που πωλούν δεν είναι σταθερή, αλλά προσαρμόζεται ανάλογα με μια στρατηγική που ακολουθεί ο χρήστης στον οποίο ανήκουν ή από την διαμόρφωση του περιβάλλοντος όπου δραστηριοποιούνται. Για παράδειγμα, ένας πωλητής μπορεί να αποφασίσει να πουλήσει ένα προϊόν-πληροφορία σε χαμηλότερη τιμή εάν διαπιστώσει ότι υπάρχει αυξημένη ζήτηση για αυτό το προϊόν.

Οι μεσολαβητές

Κύριος ρόλος τους είναι να φέρουν σε επαφή τις δύο «πλευρές». Ανάλογα με την υλοποίηση και το περιβάλλον ανάπτυξης λειτουργούν και αυτοί σε μια αγορά. Οι κύριες μορφές μεσολαβητών είναι οι εξής:

- **blackboard agents:** σε αυτούς οι αγοραστές δηλώνουν την παρουσία τους στην αγορά με στόχο την απόκτηση αγαθών. Ακόμη, τους συμβουλεύονται οι πωλητές για να πουλήσουν τα προϊόντα τους σε αγοραστές.
- **brokers:** σε αυτούς δηλώνουν το ενδιαφέρον τους τόσο οι αγοραστές όσο και οι πωλητές. Οι brokers κάνουν το «ταίριασμα» για την επίτευξη μιας αγοραπωλησίας και τη παράδοση του προϊόντος στους αγοραστές, κρατώντας μυστική την ταυτότητα των δύο μερών.
- **matchmakers:** ενημερώνουν τους αγοραστές για τις προσφορές που υπάρχουν. Οι αγοραστές και οι πωλητές από μόνοι τους κλείνουν μετά μια συμφωνία.

2.3 ΕΞΥΠΝΟΙ ΠΡΑΚΤΟΡΕΣ ΛΟΓΙΣΜΙΚΟΥ

Σε γενικές γραμμές ένας πράκτορας είναι ένα πρόγραμμα. Γιατί όμως το λέμε πράκτορα και όχι πρόγραμμα; Ο λόγος είναι ότι ένας πράκτορας δεν θέλουμε να κάνει απλά υπολογισμούς ή να τυπώνει μηνύματα της μορφής “Hello World!”, αλλά θέλουμε να συμπεριφέρεται όπως εμείς οι άνθρωποι. Στην περίπτωση των ηλεκτρονικών αγορών, θέλουμε να κάνει αγοραπωλησίες με άλλους πράκτορες.

Ένας έξυπνος πράκτορας μπορεί να οριστεί σαν μια αυτόνομη οντότητα που καθορίζει τους στόχους της και μέσα από την παρατήρηση και δράση σε ένα περιβάλλον προσπαθεί να τους επιτύχει[1]. Μια γενική περιγραφή των δυνατοτήτων του, μας κάνει να αντιληφθούμε τι είναι αυτό το επιπλέον που έχει πέρα του ότι είναι πρόγραμμα. Έτσι λοιπόν, ένας πράκτορας:

- Είναι αυτόνομος. Ξεκινά η εκτέλεση του όπως ένα συνηθισμένο πρόγραμμα και τρέχει από κει και πέρα χωρίς την παρέμβαση του χρήστη - προγραμματιστή.
- Έχει την δυνατότητα να ανιχνεύει το περιβάλλον του και να αντιδρά στα μηνύματα που λαμβάνει.
- Έχει την ικανότητα να προσαρμόζεται η συμπεριφορά του σε διαφορετικές καταστάσεις του περιβάλλοντος που επιδρά.
- Είναι φτιαγμένος έτσι ώστε να επικοινωνεί με άλλους πράκτορες.
- Μέσω διαφόρων τεχνικών μάθησης, υπό επίβλεψη ή όχι (supervised, unsupervised), μπορεί να προσδιορίσει τη συμπεριφορά του.

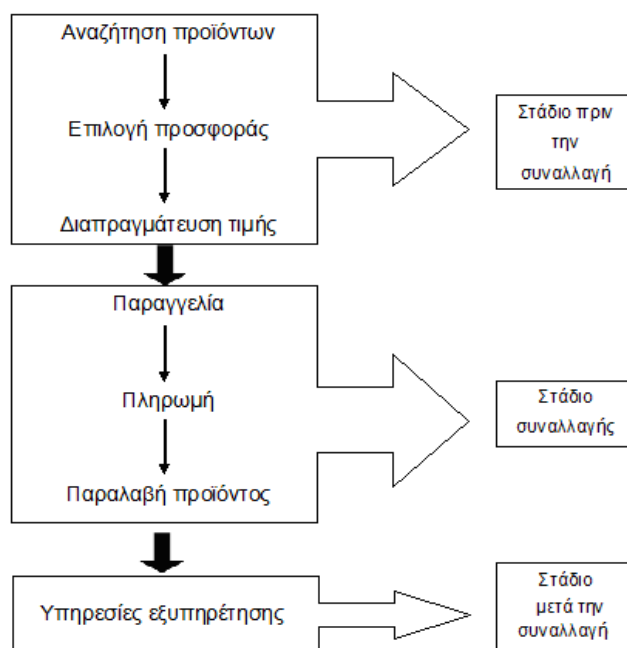
Ανάλογα με τα παραπάνω χαρακτηριστικά, οι πράκτορες μπορούν να χωριστούν σε επιμέρους κατηγορίες [1]. Ο ακόλουθος πίνακας (Πίνακας 3) παρουσιάζει όλες τις κατηγορίες πρακτόρων και περιγράφει συνοπτικά τη λειτουργικότητα τους.

Πίνακας 3. Κατηγοριοποίηση των έξυπνων πρακτόρων λογισμικού.

| ΚΑΤΗΓΟΡΙΑ | ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ | ΚΙΝΗΤΡΑ |
|------------------------|-------------------------------------|-------------------------------------------------------------------------------------------------|
| Συνεργατικοί πράκτορες | Αυτονομία, κοινωνικότητα | Όταν η χρήση ενός πράκτορα δεν είναι αρκετή. Χρήσιμος σε κατανεμημένα περιβάλλοντα. |
| Πράκτορες διεπαφής | Αυτονομία, μάθηση | Κάνει όλες τις κοπιαστικές και «ανιαρές» εργασίες. Παίζει ρόλο βοηθού για αρχάριους χρήστες. |
| Κινητοί πράκτορες | Αυτονομία, συνεργασία, κινητικότητα | Φθηνός τρόπος για να μεταφέρει μεγάλο όγκο δεδομένων. Υπερτερεί σε κατανεμημένες |

| | | |
|-------------------------|------------------------------------------------------------|-----------------------------------------------------------------------|
| | | αρχιτεκτονικές έναντι στατικών πρακτόρων. |
| Πράκτορες πληροφορίας | Προαιρετικά: συνεργασία, κοινωνικότητα, μάθηση | Αναζητά πηγές στο internet με βάση ένα προφίλ προτιμήσεων του χρήστη. |
| Αντιδραστικοί πράκτορες | Προσαρμοστικότητα, ανίχνευση περιβάλλοντος | Δείχνει ανοχή σε λάθη από άλλους πράκτορες. |
| Υβριδικοί πράκτορες | Διασταύρωση δύο ή περισσότερων από τις παραπάνω κατηγορίες | -- |
| Νοήμονες πράκτορες | Όλα | Ένας πράκτορας που τα κάνει όλα τέλεια! |

Πέρα από την ύπαρξη των ηλεκτρονικών αγορών, μια ακόμη εξέλιξη είναι η χρήση έξυπνων πρακτόρων σε αυτές. Μπορούν πλέον τόσο οι αγοραστές όσο και οι πωλητές, που αποτελούν φυσικά πρόσωπα, να αφήνουν ένα πράκτορα να συμμετέχει σε μια αγοραπωλησία και να τους εκπροσωπεί. Οι συναλλαγές αυτοματοποιούνται χωρίς την συμμετοχή ή την επίβλεψη ενός φυσικού προσώπου. Στην εικόνα παρακάτω (Εικόνα 3) βλέπουμε ένα διάγραμμα που παρουσιάζει τα στάδια που περνά ένας αγοραστής όταν κάνει μια συναλλαγή σε ηλεκτρονική αγορά. Όλες οι επιμέρους ενέργειες μπορούν να εκτελεστούν πλήρως από ένα έξυπνο πράκτορα.



Εικόνα 3. Στάδια διεξαγωγής μιας συναλλαγής από την πλευρά του αγοραστή.

Θεωρούμε πως ο χρήστης αγοραστής έχει κατασκευάσει έναν έξυπνο πράκτορα για να αναλάβει αυτός να κάνει μια αγορά. Επικεντρωνόμαστε στο στάδιο πριν την συναλλαγή. Στο στάδιο αυτό ο χρήστης δίνει στον πράκτορα τη γνώση για το ποιες είναι οι ανάγκες του. Ο πράκτορας περιηγείται αυτόνομα μέσα στο διαδίκτυο και αναζητά προϊόντα με συγκεκριμένες προδιαγραφές. Επιλέγει ανάμεσα σε αυτά εκείνο που ικανοποιεί τις επιθυμίες του και στη συνέχεια διαπραγματεύεται την τιμή με τον πωλητή. Και τα επόμενα στάδια της αγοραπωλησίας μπορούν να υλοποιηθούν από τον ίδιο πράκτορα, η εξέταση τους όμως δεν είναι αναγκαία για το δικό μας σενάριο.

Οι έξυπνοι πράκτορες μπορούν να εκπροσωπήσουν και να αντικαταστήσουν πλήρως έναν χρήστη. Πολλές φορές είναι πιο αποδοτικοί, μιας και έχουν την δυνατότητα:

- Να κάνουν εκτενή αναζήτηση μέσα στο web και να βρουν έτσι καλύτερες προσφορές.
- Να επικοινωνούν εύκολα με άλλους πράκτορες.
- Να συναλλάσσονται με πολλές πλευρές ταυτόχρονα.
- Να ανιχνεύσουν επιθέσεις από κακόβουλους πράκτορες.
- Να παρέχουν ασφάλεια και προστασία ευαίσθητων δεδομένων.

Βλέπουμε πως οι ίδιες οι προδιαγραφές και τα χαρακτηριστικά του πράκτορα μπορούν να «κολλήσουν» μέσα σε περιβάλλοντα ηλεκτρονικών αγορών. Το Web αποτελεί φυσικό χώρο γι' αυτούς και οι προδιαγραφές τους είναι κατάλληλες για να εξυπηρετούν τους στόχους τόσο των αγοραστών όσο και των πωλητών.

2.5 ΕΠΙΛΟΓΟΣ

Στο επόμενο κεφάλαιο γίνεται παρουσίαση της ασαφούς λογικής και σημαντικών εννοιών που αυτή περιλαμβάνει. Σε επόμενα κεφάλαια θα δούμε πως μπορεί αυτή να χρησιμοποιηθεί από έξυπνους πράκτορες με στόχο να μπορούν να ενταχθούν και να αποδώσουν σε περιβάλλοντα ηλεκτρονικών αγορών.

ΚΕΦΑΛΑΙΟ 3

ΑΣΑΦΗΣ ΛΟΓΙΚΗ

Στην λογική (binary logic) μια πρόταση μπορεί να είναι είτε «αληθής» ή «ψευδής». Από την άλλη, η ασαφής λογική (fuzzy logic - 1965 by Lotfi A. Zadeh [4]) που προέκυψε από τα ασαφή σύνολα, επιτρέπει περισσότερες από μια τιμές αλήθειας. Συγκεκριμένα, ο βαθμός αλήθειας μιας πρότασης (degree of truth) μπορεί να πάρει τιμές από 0 έως 1. Έτσι, παρατηρούμε πως δεν υπάρχει ακρίβεια στο να ορίσουμε τι είναι αληθές.

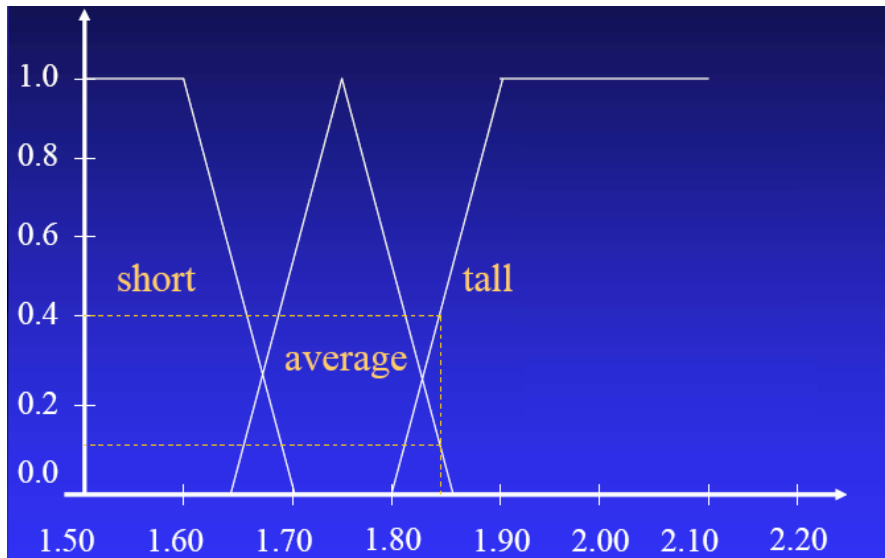
Αυτό τον τρόπο σκέψης συναντάμε καθημερινά, όταν δεν μπορούμε να περιγράψουμε με σαφήνεια ένα γεγονός. Για παράδειγμα ένα μισογεμάτο ποτήρι δεν μπορούμε να πούμε ούτε ότι είναι γεμάτο αλλά ούτε και άδειο. Έτσι, σε πολλές εφαρμογές δεν υπάρχει αυστηρότητα στο να ορίσουμε ένα γεγονός. Εδώ έρχονται λοιπόν τα ασαφή συστήματα (fuzzy logic systems). Παρακάτω, δίνεται μια περιγραφή της ασαφούς λογικής και της δομής που έχουν τα ασαφή συστήματα.

3.1 ΓΕΝΙΚΗ ΠΕΡΙΓΡΑΦΗ

Ξεκινώντας την μελέτη της ασαφούς λογικής, συναντάμε πρώτα τα ασαφή σύνολα (fuzzy sets). Έστω ένα ασαφές σύνολο A που περιέχει στοιχεία, όπως και όλα τα σύνολα. Κάθε στοιχείο του συνόλου A δεν ανήκει αποκλειστικά σε αυτό. Ο βαθμός συμμετοχής (membership degree) ενός στοιχείου x στο A είναι ένας πραγματικός αριθμός με τιμή στο $[0, 1.0]$ που υπολογίζεται μέσα από την συνάρτηση συμμετοχής (membership function) (βλέπε τύπο (1)):

$$\mu_A(x) = \begin{cases} 1 & \text{αν } x \text{ ολικά στο } A \\ 0 & \text{αν } x \text{ καθόλου στο } A \\ (0,1) & \text{αν } x \text{ μερικώς στο } A \end{cases} \quad (1)$$

Στην ασαφή λογική συναντάμε τους όρους ασαφής μεταβλητή (fuzzy variable) και ασαφής τιμή (fuzzy value). Μέσα από το ακόλουθο παράδειγμα (Εικόνα 4) θα κατανοήσουμε τον ορισμό τους. Έστω η ασαφής μεταβλητή “height”. Οι ασαφείς τιμές που μπορεί να πάρει είναι “short”, “average” και “tall”. Παρακάτω φαίνεται η γραφική παράσταση των συναρτήσεων συμμετοχής για τις τρεις ασαφείς τιμές που είναι και ασαφή σύνολα.



Εικόνα 4. Συναρτήσεις συμμετοχής.

Παρατηρούμε πως κάθε ασαφές σύνολο παριστάνει μια από τις ασαφείς τιμές που παίρνει η ασαφής μεταβλητή “height”. Τα ασαφή σύνολα καλύπτουν το ένα το άλλο σε κάποιες περιοχές τιμών. Η ένωση όλων αυτών των συνόλων μας δίνει το «σύμπαν». Το ύψος ενός ατόμου αποτελεί στοιχείο που ανήκει και στα τρία σύνολα (“short”, “average”, “tall”) αλλά με διαφορετικό βαθμό. Το άθροισμα των βαθμών θα μας κάνει πάντα 1.0.

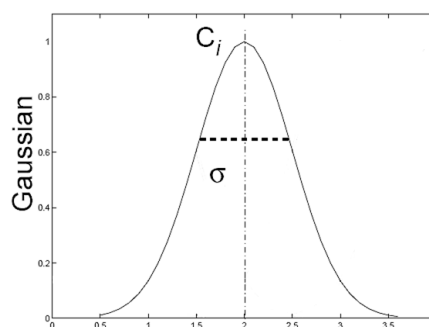
Μπορεί η γραφική παράσταση μιας συνάρτησης συμμετοχής να πάρει οποιαδήποτε μορφή, αρκεί:

- Οι τιμές της να βρίσκονται στο διάστημα [0, 1.0].
- Ο βαθμός συμμετοχής ενός σημείου σε ένα σύνολο να είναι μοναδικός.

Μια από τις πιο συνηθισμένες συναρτήσεις συμμετοχής, που θα συναντήσουμε και εμείς πολύ στο σενάριο μας, είναι η Gaussian με τύπο (τύπος (2)):

$$f(x) = a \cdot e^{-\frac{(x-b)^2}{c}} \quad (2)$$

και έχει σχηματικά τη μορφή:



Εικόνα 5. Η γραφική παράσταση μιας Gaussian κατανομής.

Στην Εικόνα 5 το c_i είναι το σημείο που έχουμε membership degree 1.0. Στο [3] μπορούμε να βρούμε και άλλους τύπους συναρτήσεων συμμετοχής.

3.2 ΑΣΑΦΕΙΣ ΚΑΝΟΝΕΣ

Αφού κάναμε μια αναφορά σε βασικές έννοιες της ασαφούς λογικής, πάμε τώρα να ορίσουμε τι είναι ασαφείς κανόνες. Ασαφής κανόνας (fuzzy rule) είναι μια πρόταση της μορφής

IF x is A THEN y is B

Όπου x, y ασαφείς μεταβλητές και A, B ασαφείς τιμές. Για παράδειγμα

IF height is tall THEN weight is heavy

Το κομμάτι “IF” του κανόνα λέγεται antecedent part και το “THEN” consequent part. Μπορεί στο antecedent part να κάνουμε χρήση των τελεστών AND, OR για να παράγουμε πιο σύνθετους κανόνες. Για παράδειγμα, αν θέλουμε να κάνουμε μια μέση έκπτωση σε ανήλικους ή φοιτητές τότε ένας κανόνας θα ήταν

IF age is teenager OR job is student THEN discount is middle

Παρατηρούμε πως οι ασαφείς κανόνες μοιάζουν πολύ με τον ανθρώπινο τρόπο σκέψης. Πολλές φορές για να λάβουμε μια απόφαση λέμε «αν ισχύει αυτό, τότε...». Η ίδια φόρμουλα υπάρχει και εδώ και μπορεί να εφαρμοστεί σε υπολογιστικά συστήματα.

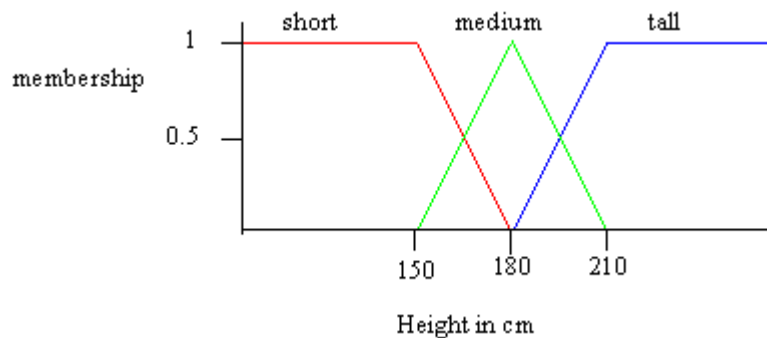
Οι ασαφείς κανόνες δέχονται τιμές για τις ασαφείς μεταβλητές στην είσοδο και βγάζουν αποτελέσματα. Οι τιμές αυτές δεν χρειάζεται να είναι ακριβείς ή διακριτές. Η ύπαρξη των ασαφών συνόλων μέσα στους κανόνες μας επιτρέπει να δίνουμε ένα εύρος τιμών μεταξύ του 0/1 ή αληθές/ψευδές και να βγάζουμε πάλι αποτελέσματα.

Συμπεραίνουμε πως η ιδέα γύρω από την ασαφή λογική και τους ασαφείς κανόνες ταιριάζει πολύ με τις προδιαγραφές που έχουμε ορίσει για τους έξυπνους πράκτορες σε ηλεκτρονικές αγορές. Ένας έξυπνος πράκτορας θέλουμε να είναι ένα πρόγραμμα με ανθρώπινο τρόπο σκέψης. Ακόμη, θέλουμε να λαμβάνει στηριζόμενος σε οτιδήποτε δεδομένα λαμβάνει από το περιβάλλον του. Και τα δυο αυτά χαρακτηριστικά τα εντοπίζουμε στους ασαφείς κανόνες. Σε επόμενα κεφάλαια θα δούμε πως μπορούμε να υλοποιήσουμε πράκτορες σε ηλεκτρονικές αγορές που με τη βοήθεια ασαφών κανόνων να λαμβάνουν αποφάσεις. Στην επόμενη ενότητα θα παρουσιάσουμε εκτενώς τη δομή των ασαφών κανόνων και το πώς λειτουργούν.

3.3 ΑΣΑΦΗΣ ΕΞΑΓΩΓΗ ΣΥΜΠΕΡΑΣΜΑΤΩΝ

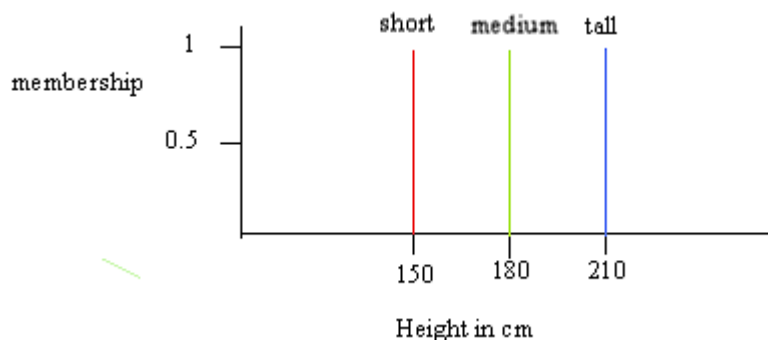
Ασαφή συστήματα (Fuzzy Systems) ονομάζονται τα συστήματα που αλληλεπιδρούν με το περιβάλλον τους και χρησιμοποιούν ασαφείς κανόνες για την λήψη αποφάσεων. Σε πραγματικές εφαρμογές γίνεται χρήση πολλών ασαφών κανόνων, που συνιστούν μια ασαφή βάση (fuzzy rule base). Ένα ασαφές σύστημα λαμβάνει κάποια είσοδο, την εφαρμόζει στους ασαφείς κανόνες και βγάζει μια έξοδο.

Υπάρχουν δύο μέθοδοι (και άρα δύο κατηγορίες συστημάτων) για εξαγωγή συμπερασμάτων. Η διαφορά τους εντοπίζεται στην μορφή που έχει ένας ασαφής κανόνας. Η πρώτη ονομάζεται Mamdani [4]. Στην κατηγορία αυτή θεωρούμε πως στο consequent part ενός κανόνα οι ασαφείς μεταβλητές παίρνουν ως τιμές συνεχή ασαφή σύνολα. Για παράδειγμα την εικόνα που ακολουθεί (Εικόνα 6) θεωρούμε την ασαφή μεταβλητή height που παίρνει ως τιμές τα ασαφή σύνολα “short”, “medium”, “tall” με συναρτήσεις συμμετοχής:



Εικόνα 6. Συνεχείς συναρτήσεις συμμετοχής.

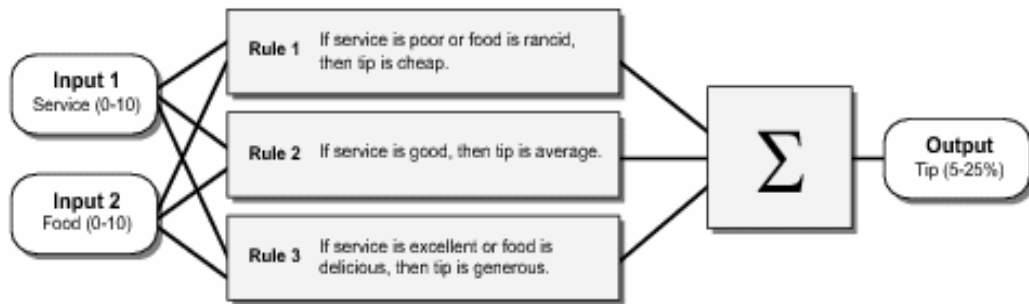
Ανάλογα με το ύψος ενός ανθρώπου, ανήκει και στα τρία ασαφή σύνολα, με διαφορετικό βαθμό συμμετοχής. Η δεύτερη κατηγορία ονομάζεται Takagi-Sugeno [4], εδώ οι membership functions του consequent part έχουν μορφή διακριτή. Έτσι αντίστοιχα για το παράδειγμα μας θα οι συναρτήσεις συμμετοχής θα είχαν την μορφή:



Εικόνα 7. Διακριτές συναρτήσεις συμμετοχής.

Μέσα από το επόμενο παράδειγμα θα μάθουμε πως λειτουργούν οι δύο μέθοδοι.

Έστω ότι έχουμε ένα σύστημα που αποτελείται από τρεις ασαφής κανόνες, όπως φαίνεται και στην Εικόνα 8:



Εικόνα 8. Πραγματική εφαρμογή ασαφούς συστήματος.

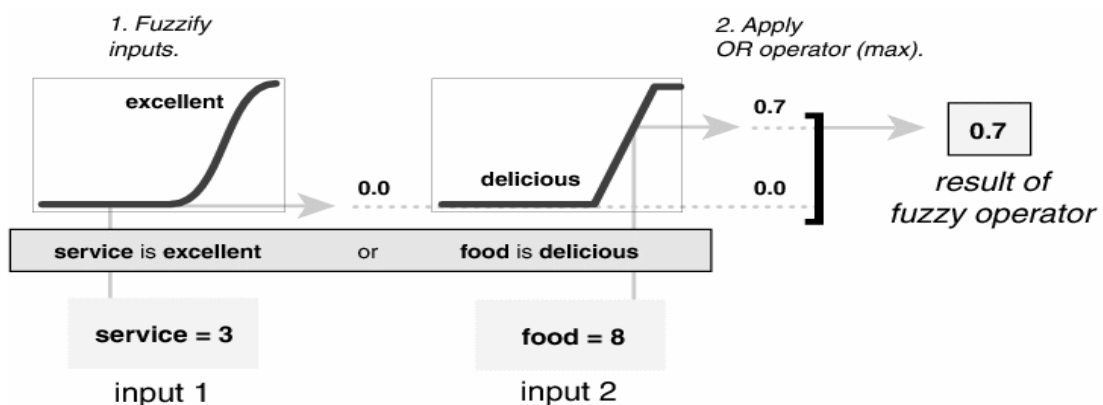
Βάζοντας ως είσοδο τιμές για το service και το food, θέλουμε να εξαγάγουμε συμπέρασμα για το tip. Θεωρούμε πως η fuzzy variable “service” παίρνει τιμές poor, good, excellent. Η fuzzy variable “food” παίρνει τιμές rancid και delicious. Το “tip” παίρνει τιμές cheap, average και generous. Τα επόμενα βήματα μας δείχνουν πως λειτουργεί το σύστημα μας.

Βήμα 1^ο (fuzzification)

Όταν μας έρθουν βαθμολογίες από το 0 έως το 10 για το food και για το service, εμείς θα τις μετατρέψουμε σε ασαφείς τιμές. Έστω ότι food = 8 και service = 3. Πρέπει να τις περάσουμε από τις αντίστοιχες membership functions των κανόνων.

Βήμα 2^ο (rule evaluation)

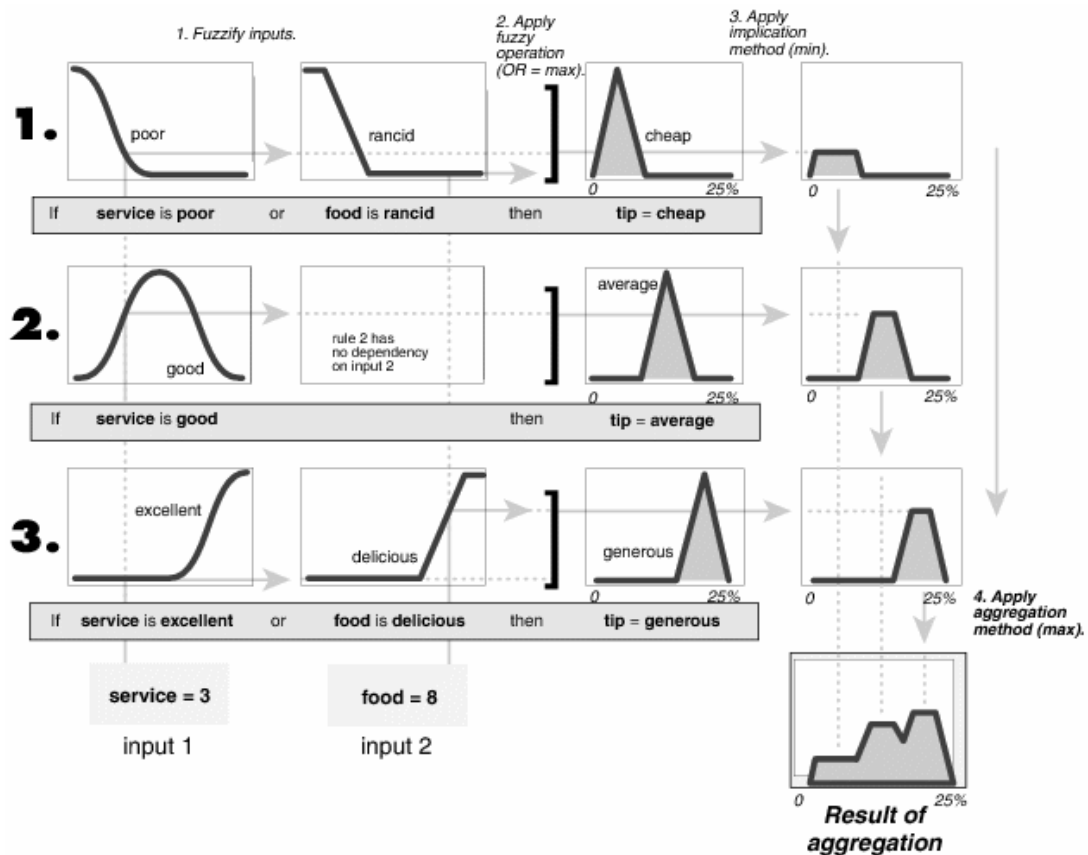
Οι κανόνες μας είναι σύνθετοι. Για τον AND τελεστή αντιστοιχεί το minimum και για τον OR η πράξη maximum. Κάνουμε τις πράξεις και ο κάθε κανόνας θα έχει τελικά μια τιμή. Για παράδειγμα ο τρίτος κανόνας θα έχει ως τιμή:



Εικόνα 9. Ο βαθμός του κανόνα.

Βήμα 3^ο (rule aggregation)

Εφαρμόζουμε το 2^ο βήμα σε όλους τους κανόνες και συναθροίζουμε το αποτέλεσμα όπως φαίνεται και στην Εικόνα 10 που ακολουθεί:

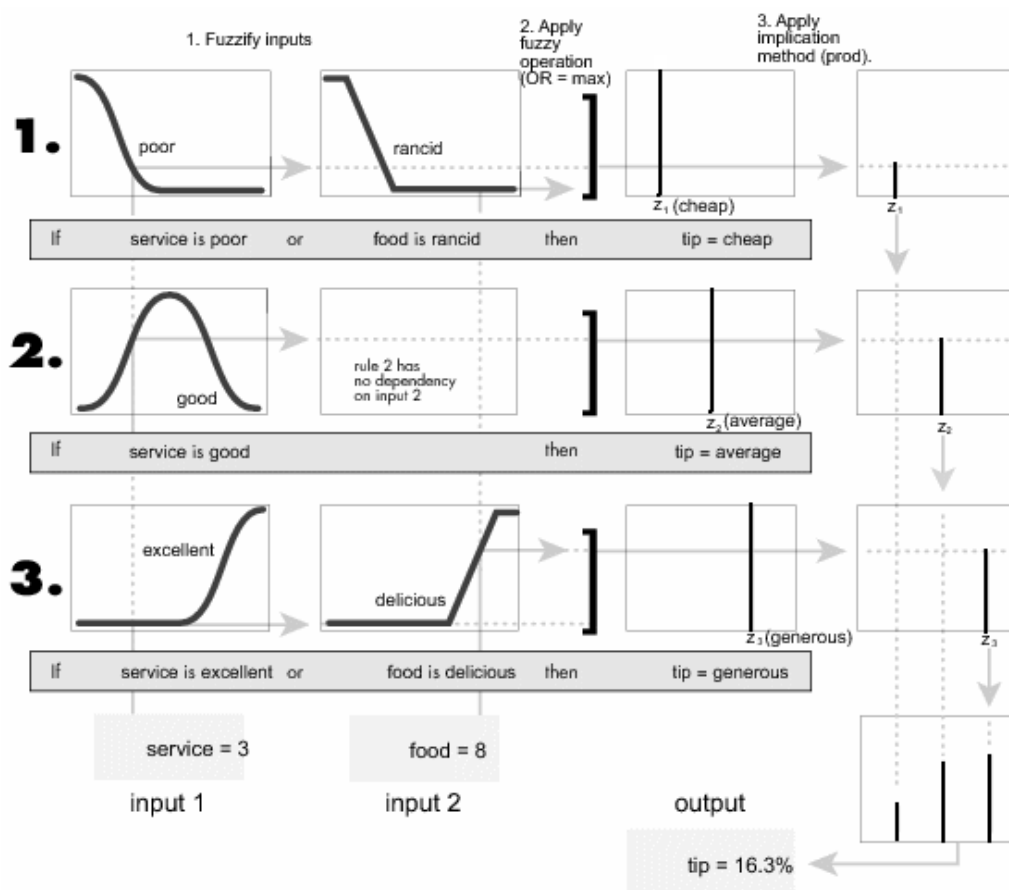


Εικόνα 10. Το τελικό αποτέλεσμα με Mamdani fuzzy rules.

Βήμα 4^ο(defuzzification)

Από το αποτέλεσμα της συνάθροισης πάμε να βγάλουμε μια τιμή για το tip από 0 έως 10. Μια συνηθισμένη μέθοδος είναι να υπολογίσουμε το κέντρο [4].

Το παραπάνω σύστημα ανήκει στην κατηγορία Mamdani, μιας και η μεταβλητή tip είχε ως τιμές ασαφή σύνολα (cheap, average, generous) με συνεχείς τιμές. Αν ακολουθούσαμε την μέθοδο Takagi-Sugeno τότε θα είχαμε την εξής μορφή:

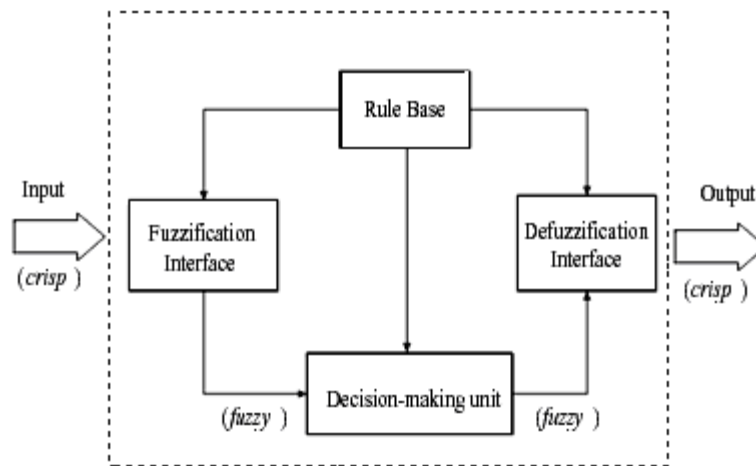


Εικόνα 11. Το τελικό αποτέλεσμα με Takagi-Sugeno fuzzy rules.

Παρατηρούμε πως η όλη διαδικασία είναι ίδια και για τις δυο μεθόδους. Η διαφορά έγκειται στο ότι στο consequent part των fuzzy rules τύπου Takagi-Sugeno έχουμε διακριτές συναρτήσεις. Το θετικό με την μέθοδο Takagi-Sugeno είναι ότι το στάδιο του defuzzification γίνεται με λιγότερες πράξεις. Όμως η μέθοδος Mamdani πλεονεκτεί στην ακρίβεια των αποτελεσμάτων που δίνει.

3.4 ΔΟΜΗ ΠΡΑΚΤΟΡΑ

Οι πράκτορες που θα υλοποιήσουμε στο μοντέλο μας ανήκουν στην κατηγορία των ασαφών πρακτόρων (fuzzy agents). Είναι και αυτοί έξυπνοι πράκτορες που αλληλεπιδρούν με το περιβάλλον της αγοράς και λαμβάνουν αποφάσεις μέσω της fuzzy rule base που έχουν οι ίδιοι κατασκευάσει. Το κομμάτι εκείνο του πράκτορα που αφορά στη χρήση της rule base έχει την ακόλουθη μορφή (βλέπε Εικόνα 12):



Εικόνα 12. Απεικόνιση της δομής ενός πράκτορα.

Η (fuzzy) rule base, γνωστή και ως knowledge base, περιέχει fuzzy if-then rules. Το ζητούμενο, λοιπόν, είναι να μελετήσουμε το πώς πρέπει να χτιστεί ένα τέτοιο σύστημα.

Η κατασκευή του τμήματος αυτού απαιτεί τη μελέτη των εξής ζητημάτων:

- Κατανόηση του προβλήματος. Πρέπει να εξετάσουμε το τι θα κάνει ο πράκτορας μας καθώς επίσης και να αναγνωρίσουμε από πόσα μέρη απαρτίζεται.
- Ορισμός των ασαφών μεταβλητών, ασαφών τιμών και των membership functions τόσο για την είσοδο όσο και για την έξοδο του fuzzy agent.
- Κατασκευή της fuzzy rule base.
- Εύρεση fuzzification / defuzzification μεθόδων.

Οι λόγοι που ένας πράκτορας κάνει χρήση της ασαφούς λογικής:

- Η ασαφής λογική είναι κάτι που το συναντάμε καθημερινά και έτσι οι ασαφείς κανόνες είναι εύκολα κατανοητοί από τον άνθρωπο.
- Ο πράκτορας μπορεί να είναι ανεκτικός σε ελλιπή δεδομένα εισόδου για να λάβει αποφάσεις.
- Μπορούν να γίνουν πράξεις με linguistic values, και όχι με αριθμητικά δεδομένα όπως έχουμε συνηθίσει.
- Μπορεί ο πράκτορας να φτιάξει ασαφείς κανόνες χωρίς πάρα πολλά αρχικά δεδομένα.
- Η βάση γνώσης μπορεί να προσαρμοστεί εύκολα σε νέα δεδομένα, επεκτείνοντας ή αλλάζοντας την δομή της.
- Επιτρέπεται να εκφραστούν με μορφή fuzzy rules input output data που έχουν nonlinear σχέση (δηλαδή δεν υπάρχει κάποια συνάρτηση που να τα συνδέει).

- Δεν χρειάζεται να προϋπάρχει κάποιο μοντέλο ώστε ο πράκτορας μας να φτιάξει τη βάση γνώσης.

3.5 ΕΠΙΛΟΓΟΣ

Ίσως, το πιο απαιτητικό στάδιο στην ανάπτυξη ενός fuzzy agent είναι το «χτίσιμο» της rule base, μιας και το κομμάτι αυτό είναι που δίνει το αποτύπωμα ολόκληρου του πράκτορα. Σε ασαφή και δυναμικά περιβάλλοντα είναι δύσκολο για ένα προγραμματιστή να συντάξει fuzzy if-then rules από μόνος του. Ο λόγος είναι ότι δεν μπορεί να λάβει υπόψη του όλους τους περιορισμούς που πρέπει να ικανοποιεί ένα fuzzy expert system, που αντικατοπτρίζονται σε fuzzy rules. Ακόμη μπορεί να μην είναι σε θέση να καταλάβει τι κάνει το σύστημα αυτό, πόσο μάλλον να μπορεί να λάβει υπόψη και περιορισμούς!

Η φιλοσοφία που ακολουθείται πια είναι να μαθαίνουμε fuzzy rules από δεδομένα. Μέσω των neural networks, των genetic algorithms ή των machine learning μεθόδων μπορούμε έχοντας μια βάση από δεδομένα, που θα μπορούσαν να είναι είσοδοι στο σύστημα μας, να φτιάξουμε αυτόματα fuzzy rules. Τα δεδομένα αυτά δεν είναι τυχαίες τιμές, αλλά είναι επιλεγμένα έτσι ώστε να ορίζουν το πώς θα πρέπει να λειτουργεί το σύστημα και τι αποφάσεις θα λαμβάνει. Στην βιβλιογραφία αναφέρονται και ως predefined labeled data. Στο κεφάλαιο που ακολουθεί γίνεται μελέτη αυτών των τεχνικών. Στην ουσία ξεκινά η μελέτη του θεωρητικού τμήματος για το σενάριο μας.

ΚΕΦΑΛΑΙΟ 4

ΤΕΧΝΙΚΕΣ ΑΥΤΟΜΑΤΗΣ ΕΞΟΡΥΞΗΣ ΒΑΣΗΣ ΚΑΝΟΝΩΝ

Στο κεφάλαιο αυτό θα μελετήσουμε τις διάφορες μεθόδους που υπάρχουν για την κατασκευή βάσης κανόνων. Θα κάνουμε μια γενική αναφορά στις κατηγορίες που υπάρχουν και έπειτα θα εστιάσουμε σε αλγόριθμους και υλοποιήσεις που αφορούν την μηχανική μάθηση.

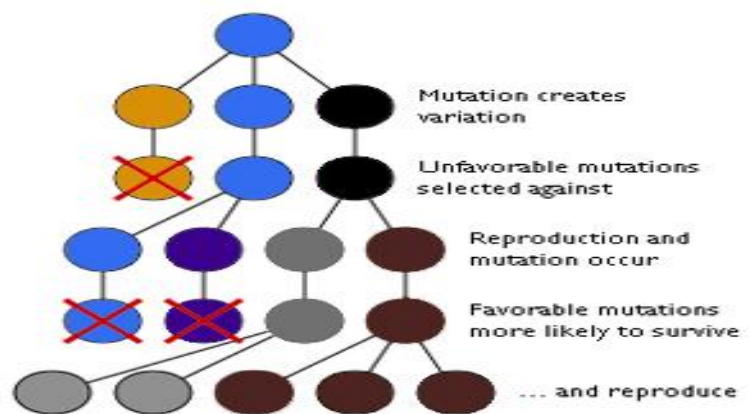
4.1 ΤΕΧΝΙΚΕΣ

4.1.1 ΓΕΝΕΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ

Χρησιμοποιούνται σε προβλήματα αναζήτησης για να βρουν ακριβείς ή βέλτιστες λύσεις. Ο τρόπος που λειτουργεί αυτή η κατηγορία αλγορίθμων μιμείται τη βιολογία [5]. Η λύση στο πρόβλημα αναπαρίσταται μέσω μιας μεταβλητής που παίρνει δυαδικές τιμές και ο αλγόριθμος ακολουθεί την εξής διαδικασία:

1. βρίσκουμε όλες τις δυνατές τιμές που μπορεί να έχει η μεταβλητή (όλους τους δυνατούς συνδυασμούς).
2. εφαρμόζουμε σε κάθε τιμή μια συνάρτηση (fitness function) που υπολογίζει το πόσο βέλτιστη είναι σαν λύση του προβλήματος.
3. επαναλαμβάνουμε τα ακόλουθα βήματα μέχρι να επιτύχουμε βέλτιστη λύση ή όταν ο χρόνος τελειώσει:
 - a. επιλέγουμε τις τιμές που δίνουν καλύτερα αποτελέσματα.
 - b. από αυτές μέσα από διαδικασίες διασταύρωσης και μετάλλαξης παράγουμε νέες τιμές.
 - c. εφαρμόζουμε στις νέες τιμές την fitness function.
 - d. για όλες τις νέες τιμές πάμε ξανά στο βήμα α.

Η εικόνα παρακάτω (Εικόνα 13) δείχνει την εξέλιξη της μεταβλητής:



Εικόνα 13. Παραγωγή όλων των δυνατών συνδυασμών τιμών.

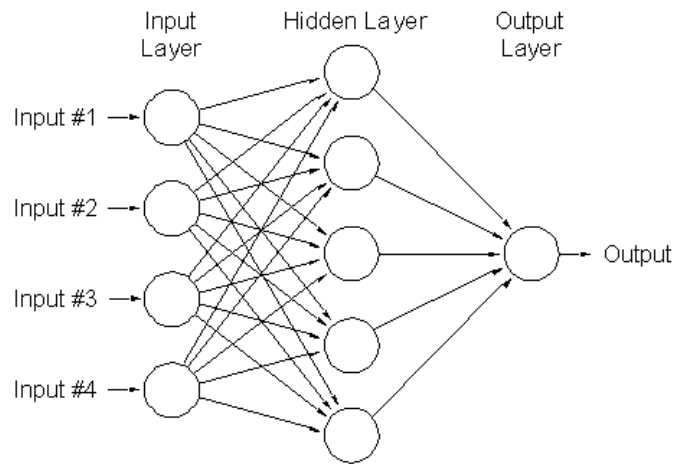
Οι αλγόριθμοι της κατηγορίας αυτής είναι εύκολο να υλοποιηθούν. Είναι πιο ισχυροί από τα συστήματα τεχνητής νοημοσύνης. Είναι ανθεκτικοί στις αλλαγές και στα noise data. Έχουν εφαρμογή και στην κατασκευή fuzzy rule bases.

4.1.2 ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

Τα νευρωνικά δίκτυα αποτελούν ένα υπολογιστικό μοντέλο που βρίσκει πληροφορία μέσα από ανακριβή ή περίπλοκα δεδομένα εισόδου. Αποτελούνται από χιλιάδες στοιχεία (νευρώνες) που είναι συνδεδεμένα μεταξύ τους. Τα νευρωνικά δίκτυα θυμίζουν πολύ την λειτουργία του νευρικού συστήματος του ανθρώπου μιας και:

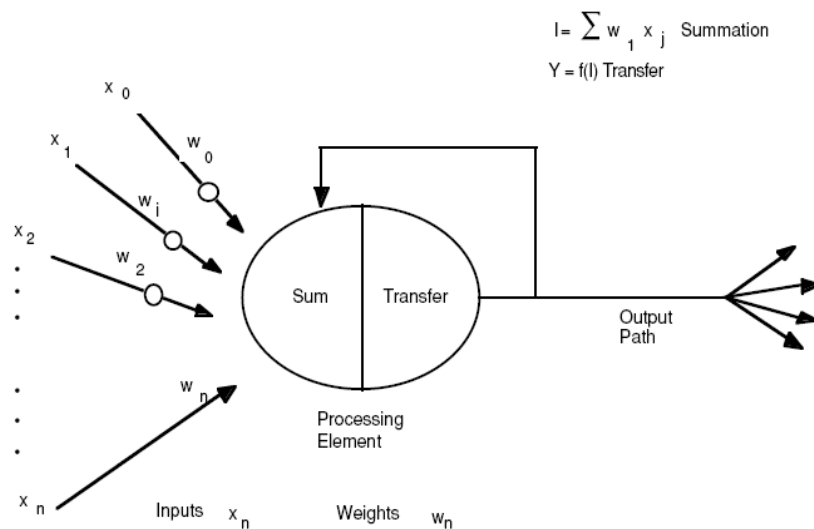
- οι νευρώνες αποθηκεύουν δεδομένα.
- μπορούν να προσαρμόζονται στις αλλαγές και να δημιουργούν ή να καταστρέφουν νευρώνες ή να αλλάζουν σχηματισμούς (συνδέσεις).
- αποκτούν γνώση για το πώς να λειτουργούν μέσα από τα δεδομένα που τους δίνονται.

Ακόμη, τα νευρωνικά δίκτυα έχουν μεγαλύτερες δυνατότητες από τα απλά προγράμματα.



Εικόνα 14. Σχηματική παράσταση Neural network τριών επιπέδων.

Όπως βλέπουμε στην Εικόνα 14, οι νευρώνες εισόδου αποτελούν το input layer. Το μόνο που κάνουν είναι να λαμβάνουν τα δεδομένα εισόδου και να τα στέλνουν στους υπολογιστικούς νευρώνες στο hidden layer. Σύμφωνα με την Εικόνα 15, κάθε υπολογιστικός νευρώνας δέχεται πολλά δεδομένα εισόδου που τα πολλαπλασιάζει με τα αντίστοιχα βάρη τους και παράγει ένα άθροισμα που το στέλνει στην συνάρτηση μεταφοράς (transfer function) για να παραχθεί ένα αποτέλεσμα.



Εικόνα 15. Παράδειγμα δομής ενός νευρώνα.

Τα νευρωνικά δίκτυα βρίσκουν εφαρμογή σε πεδία όπως function approximation, classification, data processing. Περισσότερες λεπτομέρειες για την λειτουργία τους μπορούμε να βρούμε στο [6].

4.1.3 ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Η μηχανική μάθηση αποτελεί κομμάτι της τεχνητής νοημοσύνης. Περιλαμβάνει αλγόριθμους μέσω των οποίων οι υπολογιστές «μαθαίνουν» από δεδομένα εισόδου. Στον πίνακα (Πίνακας 4) που ακολουθεί φαίνονται οι κυριότερες κατηγορίες αλγορίθμων μηχανικής μάθησης.

Πίνακας 4. Οι κατηγορίες της μηχανικής μάθησης και οι αλγόριθμοι που περιλαμβάνουν.

| ΚΑΤΗΓΟΡΙΑ | ΠΕΡΙΓΡΑΦΗ | ΑΛΓΟΡΙΘΜΟΙ |
|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------|
| Supervised learning | Στόχος είναι η παραγωγή συνάρτησης μέσα από δοκιμαστικά δεδομένα | Decision tree learning K-nearest neighbour (k-NN) Naive Bayes classifier |
| Unsupervised learning | Ανιχνεύει τη δομή που μπορεί να έχουν τα δοκιμαστικά δεδομένα. Η διαφορά του με την προηγούμενη κατηγορία είναι ότι κάνει χρήση unlabeled data | K-means FCM Hierarchical clustering Subtractive |
| Semi-supervised learning | Συνδυασμός των δυο πιο πάνω μεθόδων | Constrained clustering Transductive learning |

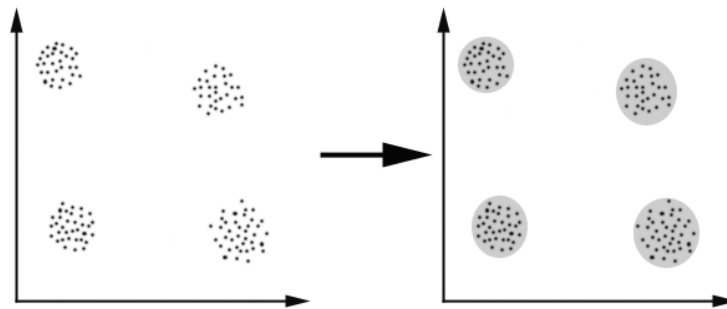
4.1.4 ΕΞΑΓΩΓΗ ΑΣΑΦΩΝ ΚΑΝΟΝΩΝ ΑΠΟ ΔΕΔΟΜΕΝΑ

Μια ακόμη προσέγγιση για την κατασκευή βάσης κανόνων είναι η χρήση των αλγορίθμων **LFE** (Learning From Examples) και **MLFE** (Modified Learning From Examples). Και οι δυο τεχνικές προσπαθούν να παράγουν fuzzy rules από μια βάση δεδομένων. Διαφέρουν από τις τεχνικές clustering στο ότι μπορεί μεν και οι δυο να έχουν ως αφετηρία δεδομένα εισόδου για να φτιάξουν τους ανάλογους κανόνες, αλλά οι LFE και MLFE δεν κάνουν βελτιστοποιήσεις. Η τεχνική LFE ζητά από τον χρήστη τις membership functions και παράγει τους κανόνες. Η MLFE είναι πιο γενική και βρίσκει από μόνη της τις membership functions. Και τις δυο τεχνικές θα μελετήσουμε εκτενώς – σχετικά με την εφαρμογή τους στο σενάριό μας - στο Κεφάλαιο 5.

4.2 ΣΥΣΤΑΔΟΠΟΙΗΣΗ

Η συσταδοποίηση (clustering) είναι ένα από τα κυριότερα προβλήματα της μη επιτηρούμενης μάθησης. Με τον όρο συσταδοποίηση [7] εννοούμε την διαμέριση ενός

συνόλου πολυδιάστατων δεδομένων σε υποσύνολα (clusters). Πιο απλά, είναι η οργάνωση σημείων σε ομάδες, των οποίων τα μέλη είναι «όμοια». Με τον τρόπο αυτό δεδομένα που ανήκουν σε ένα cluster έχουν κάποιο κοινό χαρακτηριστικό, ή αλλιώς κριτήριο ομοιότητας, που τα κάνει να διαφέρουν από δεδομένα που ανήκουν σε άλλα clusters. Το χαρακτηριστικό αυτό μπορεί να είναι για παράδειγμα η απόσταση τους που έχει υπολογιστεί από μια ορισμένη από τον χρήστη μετρική συνάρτηση. Στην εικόνα (Εικόνα 16) που ακολουθεί τα δεδομένα αποτελούν σημεία στο δισδιάστατο χώρο και το clustering γίνεται με βάση τη μεταξύ τους απόσταση.



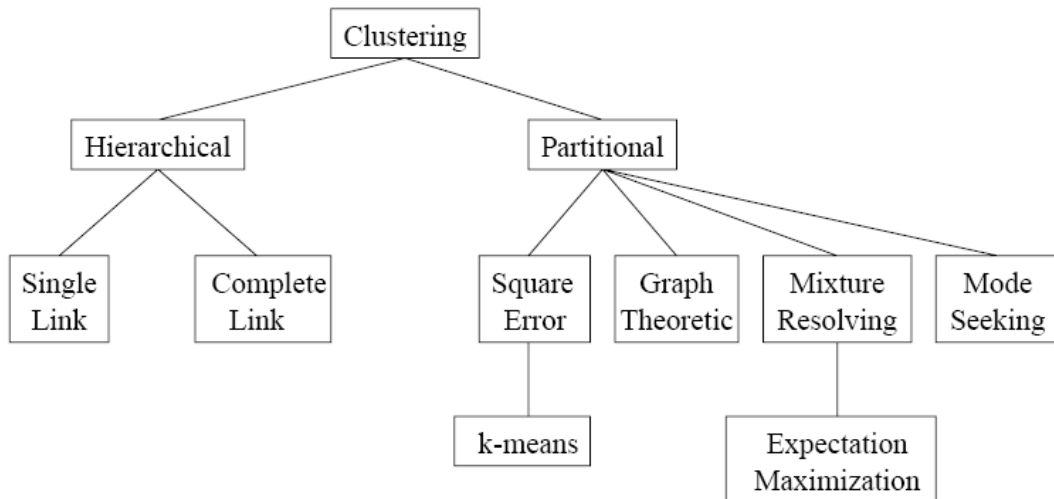
Εικόνα 16. Δημιουργία clusters από σημεία στο χώρο.

Η βασική διαφορά της συσταδοποίησης από την ταξινομική ανάλυση (discriminant analysis) είναι ότι στην πρώτη περίπτωση διαμερίζουμε ένα σύνολο από δεδομένα τα οποία δεν έχουν κάποιο ιδιαίτερο χαρακτηριστικό πέρα από τις συντεταγμένες τους (unlabeled data) [8]. Δεν υπάρχουν προκαθορισμένες κατηγορίες ούτε κάποιο παράδειγμα που θα έδειχνε ποιες επιθυμητές σχέσεις θα ήταν έγκυρες μεταξύ των δεδομένων. Στην ταξινομική ανάλυση μας δίνονται από πριν δεδομένα χωρισμένα σε συστάδες (clusters) και καλούμαστε εμείς να κατατάξουμε ένα νέο δεδομένο που θα μας έρθει ως είσοδος σε μια από τις υπάρχουσες συστάδες. Μπορούμε να πούμε πως η ταξινομική ανάλυση ακολουθεί τη διαδικασία της συσταδοποίησης. Λαμβάνει τις προσδιορισμένες συστάδες (βήμα προεπεξεργασίας) και λειτουργεί πάνω σε αυτές.

Η συσταδοποίηση μας παρέχει τη δυνατότητα να χειριζόμαστε μεγάλους όγκους δεδομένων. Τα δεδομένα από μια «ακατέργαστη» δομή μετατρέπονται σε μια «συμπιεσμένη». Έτσι, είναι ευκολότερο για το χρήστη να αναζητά γνώση μέσα από χιλιάδες εγγραφές μιας και ο όγκος μειώνεται δραματικά και μένει μόνο η χρήσιμη πληροφορία. Η ακρίβεια των αποτελεσμάτων της συσταδοποίησης είναι κύριο μέλημα μιας και μέσα από αυτά βγαίνει η γνώση για τα δεδομένα.

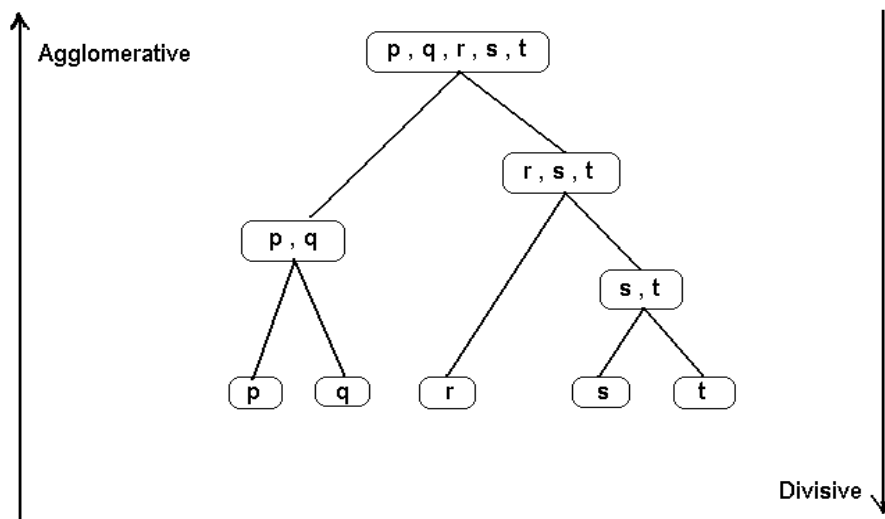
4.3 ΑΛΓΟΡΙΘΜΟΙ ΣΥΣΤΑΔΟΠΟΙΗΣΗΣ

Υπάρχουν πολλοί αλγόριθμοι με τους οποίους κάνουμε clustering σε ένα σύνολο δεδομένων. Ο κύριος διαχωρισμός τους γίνεται ανάμεσα σε ιεραρχικούς και διαμεριστικούς, όπως φαίνεται και στο σχήμα παρακάτω (Εικόνα 17) [8].



Εικόνα 17. Ιεραρχία αλγορίθμων συσταδοποίησης.

- Στις ιεραρχικές μεθόδους η διαδικασία της συσταδοποίησης μπορεί να απεικονιστεί με μια δενδρική δομή (βλέπε Εικόνα 18). Η κορυφή του δένδρου αποτελείται από ένα cluster που περιέχει όλα τα σημεία. Τα φύλλα του δένδρου είναι τόσα όσο και το πλήθος των σημείων. Κάθε φύλο είναι ένα cluster που περιέχει ακριβώς ένα σημείο. Η μέθοδος single link χτίζει αυτό το δένδρο από κάτω προς τα πάνω (δηλαδή από τα φύλλα προς την κορυφή) και η complete link αντίστροφα. “Κόβοντας” το δένδρο σε οποιοδήποτε ύψος θέλουμε, παίρνουμε τη διαμέριση των δεδομένων με ένα συγκεκριμένο πλήθος clusters.



Εικόνα 18. Δενδρική δομή clusters.

- Στις διαμεριστικές μεθόδους το πλήθος των clusters πρέπει να είναι δοσμένο εξαρχής, σε αντίθεση με προηγούμενη κατηγορία στην οποία κόβαμε όπου θέλαμε το δένδρο και παίρναμε ποικίλους αριθμούς clusters με την ίδια υλοποίηση του αλγορίθμου. Όταν ξεκινά την εκτέλεση του ένας τέτοιος αλγόριθμος αρχικά θα μοιράσει με τυχαίο τρόπο τα δεδομένα εισόδου στα clusters. Σε κάθε βήμα του αλγορίθμου τα δεδομένα αλλάζουν cluster έτσι ώστε να βελτιστοποιείται το αποτέλεσμα. Το αποτέλεσμα είναι να βρεθούν «αντιπροσωπευτικές» τιμές για τα κέντρα των clusters καθώς και το που ανήκει κάθε σημείο (σε ποιο cluster). Μερικοί αλγόριθμοι που ανήκουν σε αυτή τη κατηγορία και με τους οποίους θα ασχοληθούμε παρακάτω είναι ο K-means και ο Fuzzy C-means (FCM).

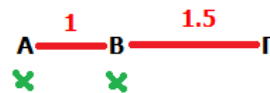
4.3.1 K-means ή HCM (Hard C-means)

Ο K-means (ή αλλιώς Hard C-means) αποτελεί έναν από τους απλούστερους αλγόριθμους συσταδοποίησης n σημείων σε k clusters. Το χαρακτηριστικό του είναι ότι πρέπει ο αριθμός k να είναι γνωστός από πριν (a priori), γι' αυτό ανήκει και στην γενικότερη κατηγορία των διαμεριστικών μεθόδων.

Αρχικά, ο αλγόριθμος διαλέγει k από τα n σημεία εισόδου να αποτελέσουν τα κέντρα των clusters. Η επιλογή αυτή είναι τυχαία. Κατά τη διάρκεια εκτέλεσης του K-means τα αρχικά αυτά κέντρα θα αλλάξουν τιμές προσεγγίζοντας τις βέλτιστες για τα σημεία εισόδου με την δεδομένη αρχικοποίηση. Η εκτέλεση σταματά όταν η συνάρτηση κόστους δεν ελαττώνεται πλέον.

Ο K-means ποτέ δεν βρίσκει ακριβή αποτελέσματα. Όμως το πόσο καλή προσέγγιση μπορεί να γίνει εξαρτάται πολύ από την αρχικοποίηση. Είπαμε ότι η επιλογή είναι τυχαία, γι' αυτό χρειάζεται να τρέξουμε τον αλγόριθμο παραπάνω από μια φορές για να πάρουμε ικανοποιητικά αποτελέσματα. Ακολουθεί ένα παράδειγμα που δείχνει το πόσο σημαντική είναι η αρχική επιλογή κέντρων.

Έστω τα σημεία A, B και Γ που θέλουμε να τα χωρίσουμε σε 2 clusters. Θα περιμέναμε ως αποτέλεσμα το ένα cluster να περιέχει τα σημεία A, B με κέντρο το μέσο τους και το άλλο cluster να περιέχει το Γ, με κέντρο το ίδιο. Όμως αν επιλέξουμε ως αρχικά κέντρα των cluster τα A, B όπως δείχνει και η εικόνα παρακάτω:



Εικόνα 19. Αρχική επιλογή κέντρων στον K-means.

τότε τελικά θα έχουμε 2 clusters, το ένα θα περιέχει το A και το άλλο τα B, Γ.



Εικόνα 20. Τελικά κέντρα των clusters μετά τον τερματισμό του K-means.

Το αποτέλεσμα αυτό όμως δεν είναι το επιθυμητό!

Τυπολόγιο:

- **Συνάρτηση κόστους** που προσπαθεί να ελαχιστοποιήσει ο K-means. Στην ουσία προσπαθεί να είναι ελάχιστη η απόσταση ενός σημείου από το κέντρο ενός cluster. Όσο περισσότερα είναι τα clusters, τόσο κοντά στο μηδέν πλησιάζει η τιμή της.

$$J = \sum_{i=1}^c J_i = \sum_{i=1}^c \left(\sum_{k, x_k \in G_i} \|x_k - c_i\|^2 \right) \quad (3)$$

- **Συνάρτηση συμμετοχής.** Αν ένα σημείο βρίσκεται πιο κοντά στο κέντρο ενός άλλου cluster, σε σχέση με το cluster στο οποίο ανήκει μέχρι στιγμής, τότε θα ανήκει από δω και πέρα στο κοντινότερο cluster.

$$u_{ij} = \begin{cases} 1, & \text{if } \|x_j - c_i\|^2 \leq \|x_j - c_k\|^2, \text{ for each } k \neq i, \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

- Η τιμή που θα έχει το **κέντρο ενός cluster** είναι ο μέσος όρος των τιμών των σημείων που ανήκουν στο cluster αυτό (όπου $|G_i|$ είναι ο πληθικός αριθμός ενός cluster, δηλαδή το πλήθος των σημείων που ανήκουν στο cluster αυτό)

$$c_i = \frac{1}{|G_i|} \sum_{k, x_k \in G_i} x_k \quad (5)$$

Ο αλγόριθμος σε ψευδοκώδικα:

Αλγόριθμος K-means

ΕΙΣΟΔΟΣ : n δεδομένα

ΕΞΟΔΟΣ : k κέντρα clusters

ΑΡΧΗ

B1: επέλεξε k από τα n σημεία εισόδου ως αρχικά κέντρα των clusters.

B2: υπολόγισε τον πίνακα u με βάση τον τύπο (4).

B3: υπολόγισε της συνάρτησης κόστους με βάση τον τύπο (3) και σταμάτα αν είναι κάτω από την προσδοκώμενη τιμή.

B4: επαν-υπολόγισε τα κέντρα των clusters σύμφωνα με τον τύπο (5).

B5: επιστροφή στο βήμα B2.

ΤΕΛΟΣ

4.3.2 FCM (Fuzzy C-means)

Στην θεωρία των ασαφών συνόλων (fuzzy sets) θεωρούμε ένα σύμπαν αναφοράς (universe of discourse) όπου τα στοιχεία του ανήκουν σε ένα ή περισσότερα σύνολα με ένα βαθμό συμμετοχής $[0, 1.0]$. Στο fuzzy clustering το σύμπαν αναφοράς είναι τα σημεία (δεδομένα) και τα fuzzy sets είναι τα clusters. Ένα σημείο δεν ανήκει μοναδικά σε ένα cluster, αλλά σε όλα με ένα βαθμό συμμετοχής για το κάθε cluster.

Ο αλγόριθμος Fuzzy C-means ανήκει σε αυτή την κατηγορία. Προτάθηκε από τον Bezdek το 1973 [9] σαν μια βελτίωση του Hard C-means. Σε αυτό τον αλγόριθμο κάθε δεδομένο ανήκει σε ένα cluster ως ένα βαθμό (membership grade). Στόχος του FCM είναι να ελαχιστοποιήσει μια συνάρτηση κόστους που μετρά το βαθμό «μη ομοιότητας» ενός σημείου σε ένα cluster.

Τυπολόγιο:

- Ένα σημείο ανήκει σε όλα τα cluster με διαφορετικό **βαθμό συμμετοχής**, όμως το άθροισμα των βαθμών του σημείου πρέπει να είναι 1.0 (όπως και στα fuzzy sets):

$$\forall x \sum_{k=1}^{\text{num. clusters}} u_k(x) = 1 \quad (6)$$

όπου u_k είναι ο βαθμός συμμετοχής του σημείου x σε ένα συγκεκριμένο cluster.

- Ο υπολογισμός της τιμής του **κέντρου ενός cluster** (ή των συντεταγμένων του κέντρου, αν μιλάμε για πολυδιάστατους χώρους) γίνεται λαμβάνοντας υπόψη τις τιμές που έχουν τα σημεία που ανήκουν στο cluster καθώς και ο βαθμός συμμετοχής τους, ο τύπος υπολογισμού δίνεται παρακάτω:

$$\text{center}_k = \frac{\sum_k u_k(x)^m x}{\sum_k u_k(x)^m} \quad (7)$$

- Ο υπολογισμός του **βαθμού συμμετοχής** ενός σημείου σε ένα cluster δίνεται από τον ακόλουθο τύπο:

$$u_k(x) = \frac{1}{\sum_i \left(\frac{d(\text{center}_k, x)}{d(\text{center}_i, x)} \right)^{\frac{2}{m-1}}} \quad (8)$$

Όπου d μια μετρική συνάρτηση που υπολογίζει την απόσταση μεταξύ δυο σημείων και m παράμετρος που προσδιορίζει πόσο ασαφής είναι η τιμή της $u(x)$, με $m \geq 1$. Όταν το m πάρει τιμές κοντά στο 1, το κέντρο του cluster που είναι κοντά στο εκάστοτε σημείο έχει περισσότερη συμμετοχή και ο FCM αρχίζει να μοιάζει με τον K-means [9]. Για πολύ μεγάλες τιμές του m , όλα τα σημεία θα έχουν ίδιο βαθμό συμμετοχής σε όλα τα clusters. Συνήθως μια τιμή κοντά στο 2 δίνει καλά αποτελέσματα και με μικρό υπολογιστικό κόστος.

- Η **συνάρτηση κόστους** που προσπαθεί να ελαχιστοποιήσει ο FCM υπολογίζεται από τον τύπο:

$$J(U, c_1, \dots, c_c) = \sum_{i=2}^c J_i = \sum_{i=1}^c \sum_{j=1}^n u_{ij}^m d_{ij}^2 \quad (9)$$

Ο αλγόριθμος σε ψευδοκώδικα:

Αλγόριθμος FCM

ΕΙΣΟΔΟΣ : n δεδομένα

ΕΞΟΔΟΣ : k κέντρα clusters, βαθμός συμμετοχής σημείων

ΑΡΧΗ

B1: αρχικοποίησε με τυχαίες τιμές του πίνακα u ώστε να ισχύει ο τύπος (6)

B2: υπολόγισε τις τιμές των κέντρων των clusters με βάση τον τύπο (7)

B3: υπολόγισε το κόστος σύμφωνα με τον τύπο (9) και σταμάτα αν είναι κάτω από την προσδοκώμενη τιμή

B4: επαν-υπολογισμός του πίνακα u σύμφωνα με τον τύπο (8)

B5: επιστροφή στο βήμα B2

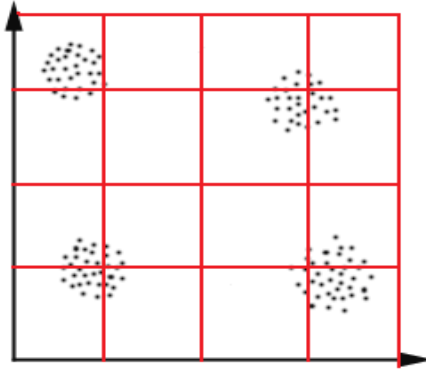
ΤΕΛΟΣ

Η απόδοση του FCM εξαρτάτε από την αρχικοποίηση του πίνακα u . Ενδείκνυται ο αλγόριθμος να τρέχει περισσότερες από μια φορές με διαφορετικές αρχικές τυχαίες τιμές για το u κάθε φορά. Τα αποτελέσματα που θα δώσει, δηλαδή τις τιμές των κέντρων των clusters, επηρεάζονται αρνητικά από σημεία που δεν ανήκουν 100% σε κάποιο cluster. Δηλαδή ο βαθμός συμμετοχής τους είναι πολύ κοντά στο 0 (noise points).

Παρότι ο FCM έχει γραμμική πολυπλοκότητα, είναι πιο αργός σε σχέση με τον K-means. Ο λόγος είναι ότι έχει να εκτελέσει πολλές πράξεις για να υπολογίζει βαθμούς συμμετοχής, κάτι το οποίο κοστίζει.

4.3.3 Subtractive clustering

Η μέθοδος subtractive clustering αποτελεί εξέλιξη της mountain clustering [9]. Αρχικά, η προσέγγιση του mountain clustering προσπαθεί να βρει τις τιμές που θα έχουν τα κέντρα των clusters. Για το σκοπό αυτό χρησιμοποιεί ένα μέτρο πυκνότητας που λέγεται mountain function. Το πρώτο βήμα στο mountain clustering είναι να χωρίσουμε τον χώρο που ανήκουν τα σημεία εισόδου σε πλέγμα (βλέπε Εικόνα 21). Για παράδειγμα σε ένα χώρο δυο διαστάσεων το πλέγμα θα έχει την ακόλουθη μορφή.



Εικόνα 21. Διαμέριση δυδιάστατου χώρου με την τεχνική mountain clustering.

Τα σημεία τομής στο πλέγμα αποτελούν τα αρχικά υποψήφια κέντρα για το πρώτο cluster. Το επόμενο βήμα είναι να εφαρμόσουμε την mountain function σε κάθε ένα από αυτά τα σημεία. Όποιο έχει τη μεγαλύτερη τιμή ως αποτέλεσμα, αυτό θα αποτελεί και το πρώτο κέντρο cluster.

Το σημείο στο οποίο διαφοροποιείται η τεχνική subtractive clustering είναι ότι δεν χωρίζει το χώρο σε πλέγμα. Αυτό που κάνει είναι να εφαρμόζει την mountain function σε κάθε σημείο εισόδου. Όποιο σημείο εισόδου έχει μεγαλύτερη τιμή στη mountain function τότε αυτό το σημείο αποτελεί το πρώτο κέντρο. Παρακάτω ακολουθεί αναλυτικά ο αλγόριθμος.

Τυπολόγιο:

- Έστω ότι έχουμε τα σημεία εισόδου $\{x_1, x_2, \dots, x_n\}$ και σε κάθε ένα από αυτά εφαρμόζουμε τη mountain function

$$P_i = \sum_{j=1}^n \exp\left(-\frac{\|x_i - x_j\|^2}{(r_a/2)^2}\right) \quad (10)$$

Όποιο από τα σημεία έχει το μεγαλύτερο P_i , αυτό θα αποτελεί και το πρώτο κέντρο για το πρώτο cluster. Με r_a συμβολίζουμε μια σταθερά που αποτελεί την απόσταση (ακτίνα) που μπορεί να βρίσκονται δυο γειτονικά σημεία.

- Αυτό το βήμα επαναλαμβάνεται για κάθε ένα από τα υπόλοιπα κέντρα των clusters που ψάχνουμε να βρούμε. Αφού βρήκαμε το κέντρο του k -οστού cluster ($k \geq 1$) πάμε στα υπόλοιπα σημεία εισόδου και εφαρμόζουμε σε κάθε ένα από αυτά την mountain function για να βρούμε το κέντρο του $k+1$ cluster

$$P'_i = P_i - \text{PotVal}(c_k) \exp\left(-\frac{\|x_i - c_k\|^2}{(r_b/2)^2}\right) \quad (11)$$

(όπου c_k το κέντρο του k -οστού cluster, $PotVal(c_k)$ η τιμή που είχε η mountain function του k -οστού cluster και ήταν μέγιστη - γι' αυτό το επιλέξαμε και ως κέντρο - και $r_b = 1.5 \cdot r_a$). Τώρα όποιο σημείο έχει τη μέγιστη τιμή P_i' , το σημείο αυτό θα αποτελεί το $k+1$ κέντρο για το $k+1$ κατά σειρά cluster. Το r_b έχει τέτοια τιμή ώστε το επόμενο κέντρο να μην είναι κοντά με το προηγούμενο. Όσα σημεία ήταν κοντά στο k -οστό κέντρο θα είχαν από πριν P_i με τιμή που θα ήταν κοντά στη μέγιστη. Τώρα το P_i' τους θα έχει μειωθεί σημαντικά από πριν και έτσι θα είναι δύσκολο να επιλεγούν ως ένα επόμενο κέντρο.

Ο αλγόριθμος σε ψευδοκώδικα:

Αλγόριθμος Subtractive

ΕΙΣΟΔΟΣ : n δεδομένα

ΕΞΟΔΟΣ : k κέντρα clusters

ΑΡΧΗ

B1: υπολόγισε για κάθε ένα από τα n σημεία εισόδου την τιμή τους P_i με βάση τον τύπο (10).

B2: επέλεξε το σημείο εκείνο με το μεγαλύτερο P_i να είναι το κέντρο του πρώτου cluster.

ΕΠΑΝΑΛΑΒΕ

B3: υπολόγισε για κάθε ένα από τα υπόλοιπα σημεία εισόδου την τιμή τους P_i με βάση τον τύπο (11)

ΒΗΜΑ 4: επέλεξε το σημείο εκείνο με το μεγαλύτερο P_i να είναι το κέντρο του επόμενου cluster.

ΜΕΧΡΙΣ ΟΤΟΥ $P_k \leq e \cdot P_1$

ΤΕΛΟΣ

P_1 : Η τιμή της mountain function για το πρώτο cluster

P_k : Η τιμή της mountain function για το k -ιοστό cluster

e σταθερά, με $e \approx 0.15$

Η τεχνική subtractive clustering είναι προτιμότερη από την mountain clustering γιατί στη δεύτερη η πολυπλοκότητα αυξάνεται εκθετικά καθώς αυξάνεται η διάσταση

που έχουν τα σημεία εισόδου. Μπορεί μιν ο subtractive να κερδίζει σε πολυπλοκότητα, χάνει όμως σε ακρίβεια των αποτελεσμάτων γιατί επιλέγει τα κέντρα των clusters μόνο από τα σημεία εισόδου.

4.3.4 Agglomerative

Ο Agglomerative ανήκει στην κατηγορία των ιεραρχικών αλγορίθμων. Συγκεκριμένα, θεωρεί πως αρχικά κάθε σημείο αποτελεί ταυτόχρονα και ένα cluster. Τα clusters αυτά κατά την επανάληψη των βημάτων εκτέλεσης θα ενώνονται (merging) έως ότου στο τέλος έχουμε ένα cluster που σε αυτό θα ανήκουν πια όλα τα σημεία εισόδου. Η εκτέλεση του Agglomerative θα μας δώσει ένα δένδρο που αναπτύσσεται από τα φύλλα προς την κορυφή (bottom up).

Υπάρχουν τρεις τρόποι για να επιλέξει κανείς σε μια επανάληψη του αλγορίθμου πια θα είναι τα επόμενα δυο clusters που θα πρέπει να ενωθούν [10]:

A. Single Linkage. Με την μέθοδο αυτή θεωρούμε ως απόσταση μεταξύ δυο clusters την ελάχιστη απόσταση που υπάρχει μεταξύ δυο σημείων που ανήκουν στα clusters αυτά. Έστω τα clusters A, B με $x \in A$ και $y \in B$. Η απόσταση των A, B θα ορίζεται ως

$$\min\{d(x, y) : x \in A, y \in B\} \quad (12)$$

Άρα θα επιλέγονται ως υποψήφια τα δυο clusters με την μικρότερη απόσταση.

B. Complete Linkage. Θεωρούμε πως η απόσταση δυο clusters είναι η maximum απόσταση που μπορεί να έχουν δυο σημεία που ανήκουν σε αυτά τα clusters δηλαδή

$$\max\{d(x, y) : x \in A, y \in B\} \quad (13)$$

Επιλέγουμε να ενώσουμε τα clusters που έχουν την μικρότερη από τις max αποστάσεις.

Γ. Average Linkage. Τέλος στην περίπτωση αυτή θεωρούμε πως η απόσταση μεταξύ δυο clusters είναι αυτή των κέντρων τους και έτσι ενώνονται τα clusters που έχουν πιο κοντά τα κέντρα τους.

Ο αλγόριθμος σε ψευδοκώδικα:

Αλγόριθμος Agglomerative

ΕΙΣΟΔΟΣ : n clusters με ένα σημείο

ΕΞΟΔΟΣ : ένα cluster με όλα τα σημεία

ΑΡΧΗ

B1: υπολόγισε για όλους τους δυνατούς συνδυασμούς ζευγών clusters την μεταξύ τους απόσταση και επέλεξε τη μικρότερη.

B 2: ένωσε τα δυο clusters.

B3: υπολογισμός του clustering gain*.

B 4: αν το πλήθος των clusters δεν είναι ένα επιστροφή στο βήμα B1

ΤΕΛΟΣ

Οι ακόλουθοι πίνακες παρουσιάζουν σημαντικές παρατηρήσεις για τους αλγορίθμους που μελετήσαμε παραπάνω.

Πέρα από την παρουσίαση των αλγορίθμων πάμε να εξετάσουμε σε βάθος τις δυνατότητες και τα χαρακτηριστικά καθενός ξεχωριστά. Στους πίνακες που ακολουθούν παρουσιάζουμε τα πλεονεκτήματα και τα μειονεκτήματα τους.

Ο Πίνακας 5 εξετάζει συνολικά όλους τους αλγορίθμους ως προς την κατηγορία στην οποία ανήκουν (διαμεριστικοί, ιεραρχικοί), την υπολογιστική τους πολυπλοκότητα, το τι δέχονται στην είσοδο για να ξεκινήσουν την εκτέλεση, τι αποτελέσματα δίνουν στην έξοδο και ποιο το κριτήριο για να τερματίσουν

Πίνακας 5. Εξέταση των αλγορίθμων ως προς τα βασικά τους χαρακτηριστικά.

| Όνομα | Κατηγορία | Πολυπλοκότητα | Παράμετροι εισόδου | Αποτέλεσμα | Κριτήριο Τερματισμού |
|---------------|--------------|---------------|--------------------|---------------------------|----------------------|
| Kmeans | Partitional | $O(n)$ | Αριθμός cluster | Κέντρα cluster | κόστος |
| FCM | Partitional | $O(n)$ | Αριθμός Cluster, m | Κέντρα cluster | κόστος |
| Subtractive | Partitional | $O(n*n)$ | r_a, e | Κέντρα cluster | Πλήθος cluster |
| Agglomerative | Hierarchical | $O(n*n*logn)$ | -- | Ανάθεση σημείων σε ομάδες | Πλήθος cluster |

Όπως τονίσαμε και πριν, δε δίνουν όλοι οι αλγόριθμοι ακριβώς τα ίδια αποτελέσματα στην έξοδο (ίδια μορφή clusters), αν επαναλάβουμε την εκτέλεση τους. Για παράδειγμα δυο διαφορετικές εκτελέσεις του FCM πάνω στο ίδιο σύνολο σημείων μπορεί να μας δώσει διαφοροποιημένες τιμές στα κέντρα των clusters, διαφορετικό βαθμό συμμετοχής σημείου σε clusters κλπ. Ο Πίνακας 6 παρουσιάζει το χαρακτηριστικό αυτό.

* σημείωση: το clustering gain είναι μετρική για την οποία θα γίνει αναφορά παρακάτω.

Πίνακας 6. Μεταβλητότητα αλγορίθμων ως προς τα αποτελέσματα.

| Όνομα | Μεταβλητότητα αποτελεσμάτων μεταξύ διαδοχικών εκτελέσεων |
|---------------|----------------------------------------------------------|
| Kmeans | Ναι |
| FCM | Ναι |
| Subtractive | Όχι |
| Agglomerative | Όχι |
| BKmeans | Ναι |

Ως outliers [11] ορίζουμε τα σημεία που έχουν κάποια απόσταση από ένα cluster, αλλά ανήκουν σε αυτό και άρα έχουν μεγάλο βαθμό συμμετοχής. Παρομοιάζονται σαν δορυφόροι του cluster. Μπορεί η θέση τους να προέκυψε έτσι είτε από κάποιο σφάλμα ή όντως να θέλαμε να υπάρχει μεγάλη κατανομή τιμών. Όταν το πλήθος των outliers είναι αμελητέο, δεν πρέπει να επηρεάζει το κέντρο του cluster. Όμως, πολλά outliers μαζί μπορούν να δημιουργήσουν ένα μικρό cluster, όταν το πλήθος τους είναι αρκετά μεγάλο.

Από την άλλη, noise σημεία χαρακτηρίζουμε αυτά που δεν ανήκουν σε κανένα cluster. Η αντιμετώπιση για τέτοια σημεία είναι να μην έχουν συμμετοχή σε κανένα cluster.

Ο Πίνακας 7 παρουσιάζει τα προτερήματα των αλγορίθμων Fuzzy C-mean και K-means. Ο λόγος που τους συγκρίνουμε αυτούς τους δυο μαζί είναι γιατί ο FCM αποτελεί εξέλιξη του HCM. Θέλουμε έτσι να εξετάσουμε παράλληλα τα προτερήματα και τις αδυναμίες τους.

Πίνακας 7. Προτερήματα των FCM και HCM.

| FCM | HCM |
|-------------------------------------------------------------------|-------------------------------------------------------------------------|
| Ανεκτικός σε outliers | Εύκολος στην υλοποίηση |
| Μετά από αρκετές εκτελέσεις δίνει καλά αποτελέσματα | Υπολογιστικά γρήγορος και συνεπώς ενδείκνυται για μεγάλο όγκο δεδομένων |
| Ο βαθμός συμμετοχής δίνει καλύτερη αντίληψη για τη διαμέριση | Μετά από αρκετές εκτελέσεις δίνει καλά αποτελέσματα |
| Συνδέεται με τους ασαφείς κανόνες και την κατασκευή βάσης γνώσης. | Δίνει καλά αποτελέσματα για σφαιρικά clusters |
| Τα όρια των clusters δεν είναι αυστηρά ορισμένα. | Πάντα συγκλίνει-τερματίζει |
| Πάντα συγκλίνει-τερματίζει | |

Ο Πίνακας 8 προβάλλει αντίστοιχα τα μειονεκτήματα τους.

Πίνακας 8. Μειονεκτήματα των FCM και HCM.

| FCM | HCM |
|----------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| Υπολογιστικά αργός | Δεν μπορεί να προβλέψει τον βέλτιστο αριθμό clusters |
| Επηρεάζεται από την τυχαία επιλογή αρχικών κέντρων | Επηρεάζεται από την τυχαία επιλογή αρχικών κέντρων |
| Ευαίσθητος σε noise data | Προσπαθεί να κατασκευάσει συνεκτικά clusters χωρίς να κοιτά αν η γενική εικόνα χρειάζεται βελτιστοποίηση |
| | Επηρεάζεται από outliers |

Στον Πίνακα 9 εξετάζουμε τον Agglomerative ξεχωριστά. Παραθέτουμε τα σημεία στα οποία πλεονεκτεί καθώς και ποιες οι αδυναμίες του.

Πίνακας 9. Τα υπέρ και τα κατά του Agglomerative.

| ΠΛΕΟΝΕΚΤΗΜΑΤΑ | ΜΕΙΟΝΕΚΤΗΜΑΤΑ |
|--------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------|
| Δεν χρειάζεται να ορίσει ο χρήστης στην είσοδο το πλήθος cluster | Μεγάλη υπολογιστική πολυπλοκότητα για την ανάπτυξη του δένδρου |
| Μας δίνεται μια οπτική απεικόνιση της όλης διαδικασίας μέσω του δένδρου των clusters | Αλλάζουν τα αποτελέσματα ανάλογα με τη μέθοδο που χρησιμοποιούμε για να υπολογίσουμε την απόσταση δυο clusters |
| Βρίσκει εφαρμογή στους τομείς της βιολογίας | Κάνει ένωση των clusters κοιτώντας το τοπικό βέλτιστο (κοντινή απόσταση) και όχι τη γενική εικόνα των σημείων |
| | Αν γίνει λάθος στην ομαδοποίηση σημείων, αυτό δεν αλλάζει όσο συνεχίζει να εκτελείται ο αλγόριθμος |

Όμοια και στον Πίνακα 10 εξετάζουμε τον Subtractive.

Πίνακας 10. Τα υπέρ και τα κατά του Subtractive.

| ΠΛΕΟΝΕΚΤΗΜΑΤΑ | ΜΕΙΟΝΕΚΤΗΜΑΤΑ |
|--------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|
| Δεν αυξάνει η πολυπλοκότητα του με την αύξηση των διαστάσεων των σημείων | Δεν δίνει καλύτερα αποτελέσματα από την τεχνική mountain clustering |
| Υπολογίζει από μόνος του το πλήθος των clusters | Πρέπει ο χρήστης να επιλέξει ανάμεσα στην καλύτερη ακρίβεια των αποτελεσμάτων και την αύξηση της πολυπλοκότητας |

Συμπεραίνουμε λοιπόν πως οι αλγόριθμοι μεταξύ τους διαφέρουν σε πολλά χαρακτηριστικά. Για παράδειγμα, δεν είναι όλοι σε θέση να μας βρίσκουν το βέλτιστο αριθμό clusters από τα σημεία εισόδου που τους δίνουμε. Πολλές φορές όμως θα το περιμέναμε εμείς να γίνεται μιας και μπορεί να μην έχουμε κάποια γραφική κατανομή των σημείων για να ήμαστε σε θέση να κρίνουμε πόσα clusters θέλουμε. Ένα άλλο σημείο στο οποίο διαφέρουν είναι η μεταβλητότητα των αποτελεσμάτων. Στην περίπτωση αυτή, θα πρέπει εμείς να εξετάσουμε αν θέλουμε να εκτελέσουμε πολλές φορές έναν αλγόριθμο ή απαιτούμε μια μόνο εκτέλεση. Μην ξεχνάμε βέβαια και την αλληλεξάρτηση μεταξύ πολυπλοκότητας και ακρίβειας αποτελεσμάτων. Σε επόμενα κεφάλαια θα γίνει εκτενέστερη μελέτη των αποτελεσμάτων και συνεπώς επιλογή του καταλληλότερου για τις ανάγκες μας (με βάση το σενάριο) αλγορίθμου.

4.4 ΠΛΗΘΟΣ CLUSTERS (optimal clustering)

Ένα ακόμη ζήτημα που προκύπτει κατά τη διάρκεια εφαρμογής μιας μεθόδου συσταδοποίησης πάνω σε ένα σύνολο δεδομένων είναι ποιος θα είναι ο αριθμός των clusters που θα χωριστούν τα δεδομένα αυτά. Οι αλγόριθμοι που είδαμε πιο πάνω, έχουν αναπτυχθεί με σκοπό να μειώσουν την υπολογιστική πολυπλοκότητα και παράλληλα να μπορούν να εφαρμόζονται για μεγάλο πλήθος δεδομένων εισόδου. Κανένας όμως από αυτούς δεν είναι σε θέση να προβλέψει ποιο είναι το βέλτιστο πλήθος clusters. Πολλές φορές πρέπει εμείς να δίνουμε τον αριθμό αυτό σαν παράμετρο εισόδου στο πρόγραμμα. Στον subtractive υπάρχει η σταθερά e που αποτελεί ένα όριο για να τερματίσει ο αλγόριθμος καθώς και η r_a ως ακτίνα των clusters. Έτσι με αυτό τον τρόπο προσπαθεί να κάνει μια προσέγγιση.

Παρότι έχει δοθεί ορισμός για το τι είναι cluster, είναι δύσκολο να μπορέσει ένας αλγόριθμος συσταδοποίησης να κρίνει αν τα clusters που έφτιαξε πλησιάζουν τη βέλτιστη λύση. Ο λόγος που συμβαίνει αυτό είναι ότι δεν δίνεται στον αλγόριθμο η γνώση για το ποια δομή έχουν τα δεδομένα εισόδου a priori. Ακόμη είναι υπολογιστικά ανέφικτο για μεγάλο όγκο σημείων εισόδου και με πολλές διαστάσεις να υπολογιστούν όλοι οι δυνατοί συνδυασμοί κατάταξης σημείου σε cluster. Έτσι αλγόριθμοι που ανήκουν στη κατηγορία των διαμεριστικών (βλέπε K-means) περιμένουν από τον προγραμματιστή να τους δώσει ένα επιθυμητό αριθμό.

Μια λύση σε αυτό το πρόβλημα δίνεται από τους ιεραρχικούς αλγορίθμους συσταδοποίησης - εμείς συγκεκριμένα θα αναφερθούμε στον Agglomerative - και κάποιες μετρικές με τις οποίες θα μπορούμε να βρούμε μια προσεγγιστικά καλή τιμή για το πλήθος των clusters. Ακολουθούν δυο προσεγγίσεις [12] στις οποίες χρησιμοποιούμε τον Agglomerative με διαφορετική μετρική κάθε φορά.

Προσέγγιση clustering balance.

Η πιο διαδεδομένη μετρική είναι η συνάρτηση κόστους (βλέπε τύπο (3)). Χρησιμοποιείται συχνά για clustering και είναι γνωστή και ως intra-cluster error sum. Μια ακόμη γνωστή μετρική είναι η inter-cluster error sum

$$\Gamma = \sum_{i=1}^c \|c_i - p_0\|^2 \quad (14)$$

όπου p_0 ο μέσος όρος όλων των σημείων (ολικό κέντρο). Οι δύο αυτές μετρικές έχουν μια χαρακτηριστική σχέση, όπως θα διαπιστώσουμε παρακάτω. Όταν χρησιμοποιούμε τον Agglomerative για clustering αρχικά κάθε σημείο αποτελεί ταυτόχρονα και cluster. Στην περίπτωση αυτή το intra-cluster error sum έχει τιμή μηδέν. Καθώς γίνονται merge clusters η τιμή της μετρικής αυτής αυξάνει ώσπου γίνεται max, όταν έχω ένα cluster που περιέχει όλα τα σημεία. Την αντίθετη συμπεριφορά εμφανίζει η inter-cluster error sum (το ολικό κέντρο παραμένει σταθερό όσα clusters κι αν έχουμε κάθε στιγμή). Έτσι το πρόβλημα του optimal clustering τροποποιείται και πλέον αναζητούμε το βήμα εκείνο του Agglomerative στο οποίο οι δυο μετρικές τέμνονται. Τη δεδομένη εκείνη στιγμή το πλήθος των clusters θεωρείται βέλτιστο.

Προσέγγιση clustering gain.

Έστω Δ_j η τιμή του clustering gain για το j-οστό cluster που υπολογίζεται από τον τύπο:

$$\Delta_j = \gamma_j - \lambda_j \quad (15)$$

με

$$\gamma_j = \sum_i \|x_i - P_0\|^2 - \|P_j - P_0\|^2 \quad (16)$$

όπου P_j το κέντρο του cluster j , P_0 το ολικό κέντρο και

$$\lambda_j = \sum_i \|x_i - P_j\|^2 \quad (17)$$

Από πράξεις προκύπτει τελικά ότι:

$$\Delta = \sum_j \Delta_j = (n_j - 1) \|P_0 - P_j\|^2 \quad (18)$$

Η μετρική αυτή εφαρμόζεται σε κάθε βήμα του Agglomerative. Όταν πάρει τη μέγιστη τιμή της, σημαίνει ότι το πλήθος των clusters είναι βέλτιστο.

Η μετρική clustering gain, την οποία χρησιμοποιήσαμε και εμείς στην υλοποίηση μας, είναι καλύτερη από την clustering balance μιας και έχει λιγότερες πράξεις, άρα απαιτεί λιγότερη υπολογιστική ισχύ.

4.5 ΠΑΡΑΤΗΡΗΣΕΙΣ

Είδαμε και πριν πως υπάρχει μια πληθώρα αλγορίθμων για να κάνει κανείς clustering σε ένα σύνολο σημείων. Ποιος όμως είναι ο λόγος που υπάρχουν τόσοι πολλοί; Η εξήγηση αυτού [13] πηγάζει από το γεγονός ότι ο κάθε προγραμματιστής έχει τη δική του αντίληψη για το τι είναι cluster. Ως αποτέλεσμα θα υπάρχουν πολλοί επαγωγικοί κανόνες (inductive principles) και μοντέλα για clustering και θα προσδιορίζει κάθε συνδυασμός από αυτά ένα αλγόριθμο.

Αρχικά, ένα θεμελιώδες ερώτημα που συναντάμε στη θεωρία της συσταδοποίησης (cluster analysis) είναι ποιο είναι το κριτήριο για να είναι δυο σημεία όμοια. Επακόλουθα, πως θα ξεχωρίζουν δυο clusters. Πολλοί διαισθητικοί κανόνες που ορίζουν τι είναι αυτό που κάνει ένα cluster συνεκτικό καταλήγουν στο να γίνονται επαγωγικοί κανόνες.

Θα προσπαθήσουμε να περιγράψουμε τι είναι επαγωγικός κανόνας και τι μοντέλο μέσα από ένα παράδειγμα. Έστω ότι έχουμε ένα σύνολο από n σημεία $\{x_1, \dots, x_n\}$, ένα μοντέλο θα υπαγορεύει κάθε σημείο να ανήκει σε ένα μόνο cluster και όχι σε πολλά. Ο επαγωγικός κανόνας, με τη σειρά του, καθορίζει ότι ένα σημείο θα

ανήκει σε ένα cluster αν και μόνο αν ελαχιστοποιείται η απόσταση μεταξύ του σημείου και του κέντρου του cluster σύμφωνα με τον τύπο (19):

$$\text{minimize } L_2(c) = \sum_{i=1}^n \text{EUCLID}^2(x_i, \text{REP}[x_i, C]) \quad (19)$$

Με βάση το μοντέλο και τον κανόνα αυτό μας δίνεται ο αλγόριθμος K-means.

Ένας άλλος επαγωγικός κανόνας με το ίδιο μοντέλο θα έλεγε ότι το σημείο ανήκει στο cluster αν:

$$\text{minimize } L_1(c) = \sum_{i=1}^n \text{EUCLID}(x_i, \text{REP}[x_i, C]) \quad (20)$$

Με βάση το μοντέλο και τον κανόνα αυτό μας δίνεται ο αλγόριθμος K-medoids.

Παρατηρούμε ότι οι δύο επαγωγικοί κανόνες διαφέρουν ως προς τον τύπο (και την υπολογιστική πολυπλοκότητα). Ένα άλλο μοντέλο μπορεί να απαιτεί ένα σημείο να ανήκει σε παραπάνω από ένα cluster (βλέπε FCM). Αυτό το μοντέλο με τη σειρά του μπορεί να έχει πολλούς διαφορετικούς επαγωγικούς κανόνες. Συμπεραίνουμε λοιπόν πως αυτή η μεγάλη ποικιλία μοντέλων και επαγωγικών κανόνων μας δίνει τόσους πολλούς αλγορίθμους clustering.

Συνήθως η κατηγοριοποίηση των διαφόρων αλγορίθμων για clustering (είδαμε σχηματικά την ιεραρχία πιο πριν) γίνεται με βάση τα διάφορα μοντέλα (επηρεάζουν την υπολογιστική πολυπλοκότητα) και όχι με βάση τους επαγωγικούς κανόνες (επηρεάζουν την ακρίβεια των αποτελεσμάτων). Στις διαμεριστικές μεθόδους ακολουθούνται πιθανοτικά ή ασαφή μοντέλα.

ΚΕΦΑΛΑΙΟ 5

ΕΦΑΡΜΟΓΗ ΜΕΘΟΔΟΛΟΓΙΑΣ ΑΥΤΟΜΑΤΗΣ ΕΞΟΡΥΞΗΣ ΣΕ ΗΛΕΚΤΡΟΝΙΚΕΣ ΑΓΟΡΕΣ

5.1 ΓΕΝΙΚΗ ΠΕΡΙΓΡΑΦΗ

Στο προηγούμενο κεφάλαιο παρουσιάσαμε τις τεχνικές που υπάρχουν για την δημιουργία ασαφών κανόνων. Μελετήσαμε διεξοδικά την κατηγορία των αλγορίθμων συσταδοποίησης. Στο παρόν κεφάλαιο θα πάμε ένα βήμα παρακάτω και θα δούμε πως ακριβώς τα clusters συνδέονται με τους fuzzy rules. Ακόμη, θα περιγράψουμε την μεθοδολογία που ακολουθούμε για την κατηγορία εξαγωγής ασαφών κανόνων από δεδομένα.

5.2 ΕΞΑΓΩΓΗ ΑΣΑΦΩΝ ΚΑΝΟΝΩΝ ΑΠΟ CLUSTERS

5.2.1 FCM

Εκτελούμε τον FCM για ένα πλήθος δεδομένων εισόδου. Το αποτέλεσμα της εκτέλεσης είναι τα κέντρα των clusters καθώς και ο βαθμός συμμετοχής κάθε σημείου σε κάθε cluster.

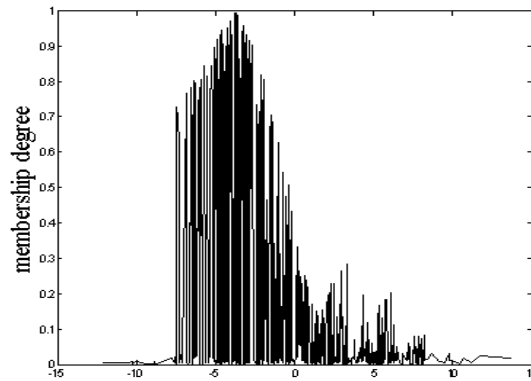
Η μεθοδολογία που ακολουθούμε [14] λέει πως κάθε cluster αντιστοιχεί σε ένα fuzzy rule. Συγκεκριμένα, αν x^* το κέντρο ενός cluster τότε ο ανάλογος fuzzy rule θα είναι

$$\text{IF } x_1 \text{ is } x_1^* \text{ AND } x_2 \text{ is } x_2^* \dots \text{ AND } x_{n-1} \text{ is } x_{n-1}^* \text{ THEN } x_n \text{ is } x_n^*$$

όπου n η διάσταση των δεδομένων και x ένα σημείο που θέλουμε να εφαρμόσουμε πάνω του τον fuzzy rule για να πάρουμε την τιμή x_n . Βλέπουμε πως οι $n-1$ πρώτες διαστάσεις των δεδομένων είναι το antecedent part του fuzzy rule και η n -ιοστή συντεταγμένη είναι το consequent. Πρέπει να αναφέρουμε εδώ πως το x_n^* έχει διακριτές τιμές. Fuzzy rules αυτού του τύπου ανήκουν στην κατηγορία Takagi-Sugeno.

Το ζητούμενο είναι να βρούμε τις συναρτήσεις συμμετοχής για όλες τις ασαφείς μεταβλητές που υπάρχουν σε όλους τους κανόνες. Ακολουθούν τα βήματα μετάβασης από ένα fuzzy cluster σε ένα fuzzy rule. Μέσα από τη διαδικασία αυτή προσπαθούμε να προσεγγίσουμε τη γραφική παράσταση των συναρτήσεων συμμετοχής.

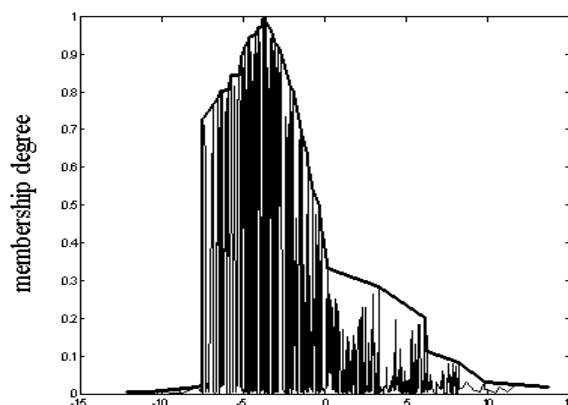
- Για κάθε cluster, όλα τα δεδομένα εισόδου έχουν ένα βαθμό συμμετοχής με το cluster αυτό. Σε κάθε διάσταση του χώρου πάμε και προβάλλουμε τον βαθμό συμμετοχής όλων των δεδομένων στο cluster. Η εικόνα παρακάτω (Εικόνα 22) δείχνει τι αποτέλεσμα προκύπτει μετά την προβολή σε μια διάσταση:



Εικόνα 22. Προβολή βαθμών συμμετοχής.

Τα σημεία με συντεταγμένη κοντά στο -5 έχουν μεγάλο βαθμό συμμετοχής στο cluster και άρα είναι πολύ κοντά στο κέντρο του. Η κατανομή των βαθμών συμμετοχής στα σημεία δεν είναι συμμετρική από αριστερά και δεξιά του κέντρου του cluster. Πρέπει εμείς λοιπόν να βρούμε τη γραφική παράσταση της συνάρτησης συμμετοχής που να προσεγγίζει όσο το δυνατόν καλύτερα την προβολή τιμών που έχουμε.

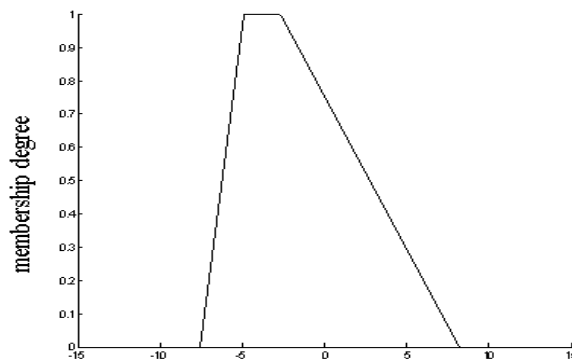
- Το επόμενο βήμα είναι να προσεγγίσουμε την προβολή αυτή με ένα τραπέζιο, όπως φαίνεται και στην ακόλουθη εικόνα (Εικόνα 23):



Εικόνα 23. Προσέγγιση συνάρτησης συμμετοχής.

Παρατηρούμε πως γίνεται μια «τρογγυλοποίηση» και κάποια σημεία χάνουν τον πραγματικό βαθμό συμμετοχής τους. Η «τρογγυλοποίηση» αυτή συνεχίζει και

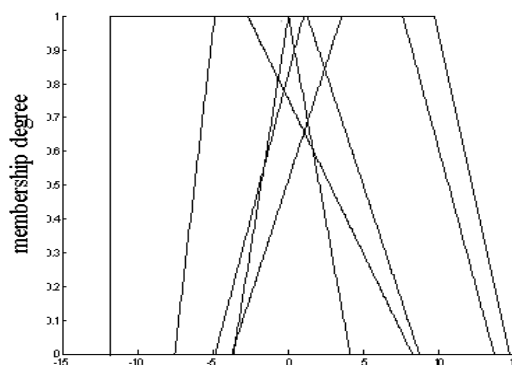
το τελικό αποτέλεσμα που θα πάρουμε στην ακόλουθη γραφική παράσταση (βλέπε Εικόνα 24) είναι ένα τραπέζιο.



Εικόνα 24. Συνάρτηση συμμετοχής.

Η γραφική παράσταση αυτή είναι η συνάρτηση συμμετοχής μιας ασαφούς τιμής, για ένα ασαφή κανόνα. Κάνοντας την σύγκριση της Εικόνας 22 με την Εικόνα 24, βλέπουμε πως προσπαθούμε να κάνουμε μια όσο το δυνατόν καλύτερη προσέγγιση. Σε κάποια σημεία ο βαθμός συμμετοχής αυξάνεται (όσων ο βαθμός συμμετοχής είναι κοντά στο 1.0) και σε άλλο ελαττώνεται (βαθμός συμμετοχής κοντά στο 0). Αυτό συνεπάγεται χάσιμο πληροφορίας, είναι όμως αναπόφευκτο. Να σημειώσουμε πως δεν είναι ανάγκη η προσέγγιση μας να γίνεται μόνο με τραπεζοειδής συναρτήσεις συμμετοχής. Για μεγαλύτερη ευκολία στην υλοποίηση μπορούμε να χρησιμοποιήσουμε και Gaussian λίγο παραμορφωμένες (δεν θα υπάρχει συμμετρία στην κατανομή μεταξύ του κέντρου, truncated Gaussian). Και η προσέγγιση αυτή δίνει καλά αποτελέσματα.

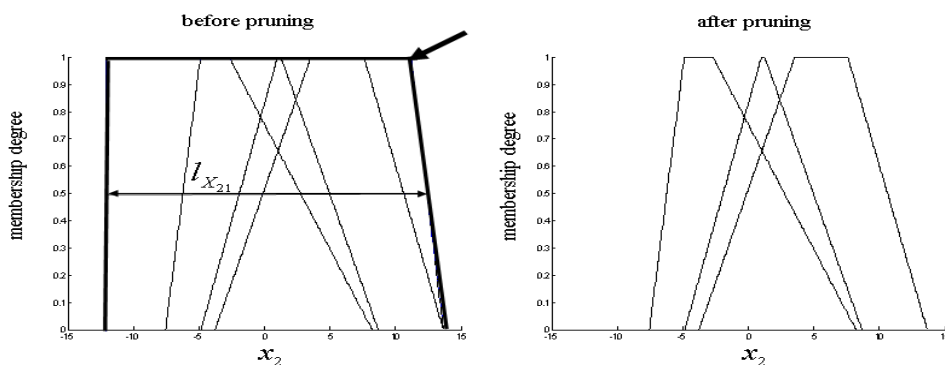
Τη διαδικασία αυτή την κάνουμε για όλα τα clusters σε όλες τις διαστάσεις. Έτσι για μια διάσταση προκύπτει η εξής μορφή (Εικόνα 25):



Εικόνα 25. Συναρτήσεις συμμετοχής σε μια διάσταση.

Η εικόνα μας δείχνει τις membership functions όλων των clusters σε μια συγκεκριμένη διάσταση. Δηλαδή για την συγκεκριμένη διάσταση, που αποτελεί ασαφή μεταβλητή, υπάρχουν όλες οι δυνατές ασαφείς τιμές-σύνολα.

- Αν κοιτάξουμε καλύτερα τις γραφικές παραστάσεις στην Εικόνα 25 βλέπουμε ότι υπάρχουν συναρτήσεις συμμετοχής με μεγάλο εύρος τιμών. Αυτό σημαίνει ότι αν πάμε να βρούμε το βαθμό συμμετοχής ενός σημείου εφαρμόζοντας το σε αυτή τη συνάρτηση συμμετοχής, θα μας δίνει πάντα 1.0, κάτι το οποίο μας είναι άχρηστο. Είναι σαν να έχουμε ένα σύνολο στο οποίο ανήκουν όλα τα σημεία με βαθμό 1.0. (universe of discourse). Τέτοιες γραφικές παραστάσεις μας είναι άχρηστες γι' αυτό σε κάθε διάσταση σβήνουμε τις συναρτήσεις που εκτείνονται σε μεγάλη περιοχή τιμών όπως φαίνεται και στην Εικόνα 26.



Εικόνα 26. Απαλοιφή συναρτήσεων συμμετοχής.

Τέλος, με την επανάληψη των προηγούμενων βημάτων σε όλες τις διαστάσεις, θα έχουν φτιαχτεί όλοι οι fuzzy rules, τόσοι όσα και τα clusters. Ακόμη θα έχουν φτιαχτεί όλες οι membership functions έτσι ώστε αν μετά μας δοθεί μια είσοδος από σημεία, εμείς να τα εφαρμόσουμε στη rule base και να βγάλουμε αποτέλεσμα.

5.2.2 Subtractive

Ποια είναι η γέφυρα για να πάμε από subtractive clusters σε fuzzy rules; Η μέθοδος [15] λέει τα εξής:

- Εφαρμόζουμε τον subtractive clustering αλγόριθμο για να παράγουμε τα κέντρα των clusters. Κάθε cluster αντιστοιχεί σε ένα fuzzy rule όπως και πριν.
- Δεν κάνουμε προβολή των τιμών συμμετοχής μιας και κάθε σημείο ανήκει αποκλειστικά σε ένα cluster. Σύμφωνα με τη βιβλιογραφία σε κάθε κανόνα ο βαθμός συμμετοχής δίνεται από τον τύπο (21):

$$\mu_i = e^{-a\|x-x_i^*\|^2} \quad (21)$$

που x το σημείο, x_i^* το i -οστό cluster center, $a = (2/r_a)^2$. Ο τύπος αυτός* ισχύει και σε πολυδιάστατους χώρους (ανάλογα υπολογίζεται η απόσταση μεταξύ σημείου και cluster center). Το input data θα το εφαρμόσουμε σε όλους τους fuzzy rules και θα πάρουμε τα αντίστοιχα μ_i . Το σχήμα των membership functions ακολουθεί την Gaussian κατανομή. Άρα δεν χρειάζεται να βρούμε εμείς τη γραφική μορφή, όπως κάναμε πριν με τον FCM.

Στο σημείο αυτό πρέπει να αναφέρουμε πως μεγάλο ρόλο παίζει η επιλογή της τιμής που θα έχει η παράμετρος r_a . Την παράμετρο αυτή τη χρησιμοποιούμε τόσο κατά τη διάρκεια εύρεσης clusters όσο και στο στάδιο αυτό. Προσδιορίζει ποια θα είναι η ακτίνα ενός cluster. Θεωρούμε πως έχουμε κανονικοποιήσει τις τιμές των input data έτσι ώστε να παίρνουν τιμές στο διάστημα $[0,1.0]$. Στο σενάριο που μελετάμε για buyers και sellers έχουμε κάποια predefined labeled data πάνω στα οποία εφαρμόσαμε την μεθοδολογία αυτή και παρατηρήσαμε κάποιες ενδείξεις για το r_a και τις τιμές του.

Πίνακας 11. Σύγκριση τιμών για διαφορετικές εκτελέσεις του Subtractive.

| r_a | Clusters number | Variation (%) |
|-------|-----------------|---------------|
| 0.5 | 100 | 2.628 |
| 0.6 | 64 | 8.68 |
| 0.8 | 30 | 13.41 |
| 1.0 | 11 | 34.26 |

Όπου:

Cluster number: ο βέλτιστος αριθμός clusters

Variation: προκύπτει από την εφαρμογή test input data στην rule base. Για τα δεδομένα αυτά γνωρίζουμε από πριν την τιμή της n -οστής συντεταγμένης, αλλά βλέπουμε τι τιμή θα δώσει η rule base αν δεν την γνωρίζαμε

Variation = $(x(n) - y) / x(n) * 100\%$, με $x(n)$ την πραγματική τιμή και y την τιμή από τον fuzzy controller.

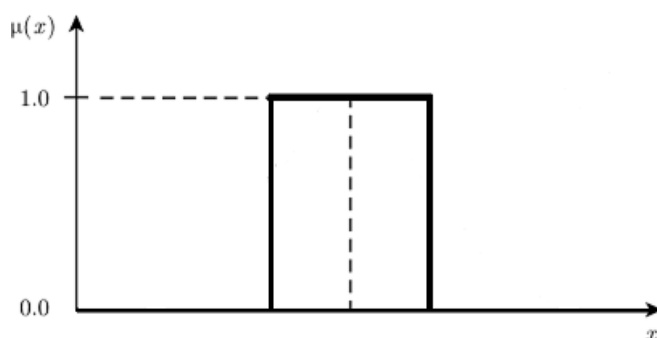
5.2.3 HCM και Agglomerative

Για αυτούς τους δύο αλγορίθμους ισχύουν τα εξής:

- Ένα σημείο ανήκει αποκλειστικά σε ένα cluster

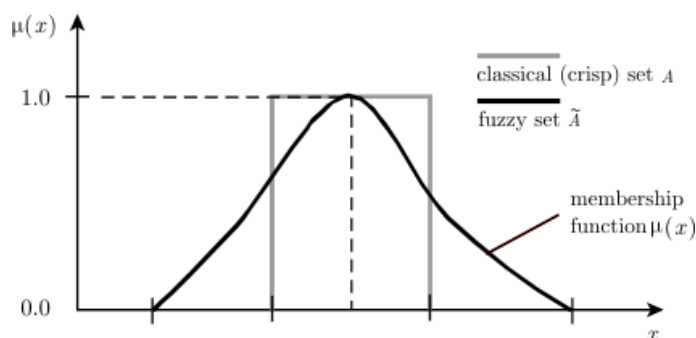
- Δεν υπάρχει μια καθορισμένη ακτίνα για το cluster όπως με τον subtractive.

Για να παράγουμε τις συναρτήσεις συμμετοχής κάνουμε ακριβώς τα ίδια βήματα όπως στον FCM. Το αποτέλεσμα όμως θα είναι μια γραφική παράσταση όπως η επόμενη (βλέπε Εικόνα 27).



Εικόνα 27. Συνάρτηση συμμετοχής.

Εμείς όμως θέλουμε ασαφή σύνολα, γι' αυτό θα προσεγγίσουμε την γραφική παράσταση με μια Gaussian όπως φαίνεται παρακάτω



Εικόνα 28. Προσέγγιση με Gaussian.

Παρατηρήσεις:

- Ο FCM μας διευκολύνει στο γεγονός ότι υπάρχει ήδη η έννοια της συμμετοχής και μπορεί εύκολα να γίνει μετάβαση σε ασαφής κανόνες.
- Η προσέγγιση με την Gaussian που κάνουμε στον HCM συνεπάγεται χάσιμο πληροφορίας, αλλά ο HCM είναι πιο γρήγορος υπολογιστικά (σε σύγκριση με τον FCM).

5.2.4 Nearest Neighborhood Clustering (NNC)

Η τεχνική που θα εξετάσουμε [16] τώρα αφορά στο να χτίζουμε πάλι μια rule base μέσα από predefined labeled data, μόνο που τώρα δεν θα χρειαστεί να φτιάξουμε clusters. Ακολουθούν τα βήματα της μεθόδου:

- Παίρνουμε το πρώτο σημείο/δεδομένο, έστω $(x_1, x_2, \dots, x_n, y)$. Το σημείο αυτό θεωρούμε ότι είναι και ο πρώτος rule για τη rule base που χτίζουμε. Θα αρχικοποιήσουμε $u = (x_1, x_2, \dots, x_n)$, $A_1 = y, B_1 = 1$.
- Προχωρώντας παίρνουμε το επόμενο σημείο έστω $(x'_1, x'_2, \dots, x'_n, y')$ και για κάθε rule που υπάρχει στη rule base μας ελέγχουμε αν $|x - u_i| \leq e_f$ όπου u_i αφορά τον i -οστό rule και e_f θετική σταθερά που επιλέγουμε εμείς την τιμή της. Ο πρώτος rule που θα βρεθεί να ικανοποιεί αυτή την ανισότητα αλλάζει και γίνεται $A_i += y'$ και $B_i += 1$ χωρίς να προσθέσουμε κάποιο νέο κανόνα στη βάση μας.
- Τι γίνεται όμως αν δεν βρεθεί κανένας rule που να ικανοποιεί την ανισότητα; Τότε γίνεται προσθήκη νέου κανόνα στη rule base με τον ίδιο ακριβώς τρόπο που έγινε και για το πρώτο input data.

Μέχρι αυτό το σημείο παρατηρούμε πως:

- το B σε ένα κανόνα προσδιορίζει το πλήθος των σημείων που ανήκουν στον κανόνα αυτό
- δεν αλλάζουν τα A, u σε ένα κανόνα όταν προσθέτουμε ένα σημείο.

Έως εδώ κατασκευάσαμε τη rule base. Τι κάνουμε όμως όταν ένα test input data έρθει και περιμένουμε να υπολογιστεί το “y” του; Στηριζόμενοι στο [15] από τον τύπο (22):

$$f(x | \theta) = \frac{\sum_{i=1}^R A_i \prod_{j=1}^n \exp\left(-\left(\frac{x_j - u_j^i}{2\sigma}\right)^2\right)}{\sum_{i=1}^R B_i \prod_{j=1}^n \exp\left(-\left(\frac{x_j - u_j^i}{2\sigma}\right)^2\right)} \quad (22)$$

Εμείς έχουμε γνωστά τα x , σ , u , A , B και βρίσκουμε ως αποτέλεσμα μια τιμή για το y . Το πόσο καλά θα προσεγγίσουμε την πραγματική τιμή που θα είχε το y εξαρτάται από τις παραμέτρους σ , e_f . Το σ επηρεάζει την μορφή που θα έχει η Gaussian, για μικρό σ είναι στενή. Το e_f , από την άλλη, καθορίζει την απόσταση που θα πρέπει να έχει ένα test input data από το u του εκάστοτε rule.

Πίνακας 12. Σύγκριση εκτελέσεων για $\sigma = 0.2$.

| e_f | Clusters number | Variation (%) |
|-------|-----------------|---------------|
| 0.2 | 108 | 29.08 |
| 0.5 | 56 | 55.75 |
| 0.9 | 16 | 73.91 |

Πίνακας 13. Σύγκριση εκτελέσεων για $\sigma = 0.1$.

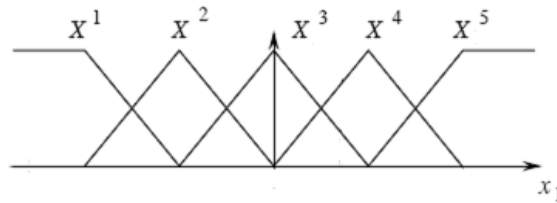
| e_f | Clusters number | Variation (%) |
|-------|-----------------|---------------|
| 0.2 | 108 | 1.81 |
| 0.5 | 56 | 46.04 |
| 0.9 | 16 | 74.74 |

5.3 ΕΞΑΓΩΓΗ ΑΣΑΦΩΝ ΚΑΝΟΝΩΝ ΑΠΟ ΔΕΔΟΜΕΝΑ

5.3.1 Learning from Examples (LFE)

Η προσέγγιση που θα μελετήσουμε τώρα λειτουργεί πάλι με το να χτίζουμε τη rule base μέσα από data. Ο τρόπος που γίνεται αυτό [16] παρουσιάζεται στα επόμενα βήματα:

- Αρχικά, παίρνουμε κάθε διάσταση του χώρου και για το διάστημα στο οποίο επεκτείνεται (εδώ $[0.0, 1.0]$) ορίζουμε συναρτήσεις συμμετοχής (βλέπε Εικόνα 29).



Εικόνα 29. Συναρτήσεις συμμετοχής ορισμένες από τον χρήστη.

Οι συναρτήσεις μπορούν να έχουν οποιαδήποτε μορφή μας βολεύει (Gaussian, τριγωνικές).

- Στη συνέχεια, για το πρώτο σημείο που έχουμε για να χτίσουμε τη βάση, παίρνουμε μια προς μια κάθε συντεταγμένη του και βλέπουμε σε ποιο fuzzy set έχει μέγιστο βαθμό συμμετοχής, για παράδειγμα για το x μπορεί το x_1 να ανήκει στο fuzzy set X^5 , το x_2 στο X^1 κλπ. Έτσι θα φτιαχτεί ο πρώτος κανόνας που θα έχει τη μορφή:

$$\text{IF } x_1 \text{ is } X_1 \text{ AND... AND } x_n \text{ is } X_n \text{ THEN } y \text{ is } Y$$

Ο βαθμός του κανόνα αυτού είναι $\min\{\mu(x_1), \dots, \mu(x_n), \mu(y)\}$.

- Για κάθε επόμενο σημείο, κατασκευάζουμε προσωρινά ένα fuzzy rule, όπως κάναμε και για το πρώτο σημείο, αλλά εφαρμόζουμε το σημείο αυτό και σε όλους τους rules που υπάρχουν ήδη στη βάση.
 - Αν ο προσωρινός κανόνας δεν μοιάζει με κανένα από τους ήδη υπάρχοντες (δεν έχουν δηλαδή ίδιο **antecedent part**), τότε ο προσωρινός προστίθεται στη βάση.

- ο Αλλιώς, υπάρχει ένας rule που είναι ίδιος με τον προσωρινό. Σε αυτή την περίπτωση θα παραμείνει στη βάση αυτός με το μεγαλύτερο βαθμό. Μπορεί δηλαδή να γίνει αντικατάσταση του παλιού με τον προσωρινό rule, αλλά μπορεί και η rule base να μείνει ανέπαφη.

Μεγάλο ρόλο παίζει το πώς θα επιλέξουμε τα fuzzy sets για κάθε διάσταση του προβλήματός μας. Αν δεν θέλουμε να δώσουμε τόση λεπτομέρεια σε μια διάσταση τότε ορίζουμε πιο «πλατιές» membership functions. Σημειώνουμε πως καθ' όλη την μελέτη όλων των σημείων, οι membership functions δεν αλλάζουν μορφή.

Όταν, λοιπόν, έρθει κάποιο test input data το οποίο θα εισάγουμε στη rule base για να πάρουμε το consequent part, θα εφαρμόσουμε τον τύπο (23):

$$f(x|\theta) = \frac{\sum_{i=1}^R y_i \prod_{j=1}^n \exp\left(-\frac{1}{2} \left(\frac{x_j - c_j^i}{2\sigma}\right)^2\right)}{\sum_{i=1}^R \prod_{j=1}^n \exp\left(-\frac{1}{2} \left(\frac{x_j - c_j^i}{2\sigma}\right)^2\right)} \quad (23)$$

Η διαδικασία αυτή είναι ίδια όπως και στην ενότητα με τον subtractive.

5.3.2 Modified Learning from Examples (MLFE)

Η προσέγγιση αυτή είναι μια γενίκευση της LFE. Αυτό που κάνει την διαφορά εδώ είναι το ότι οι membership functions αλλάζουν δυναμικά, ανάλογα με τα δεδομένα. Πάμε, όμως να δούμε την μέθοδο [16] αυτή πως αναπτύσσεται στα ακόλουθα βήματα.

- Παίρνουμε το πρώτο σημείο $x = (x_1, x_2, \dots, x_n, y)$, από τη συλλογή δεδομένων που έχουμε στη διάθεση μας. Κάνουμε τις αναθέσεις: $b = y$, $c = \{x_1, \dots, x_n\}$ και για κάθε διάσταση στο χώρο έχουμε μια ξεχωριστή τιμή για τη σταθερά “σ”, έστω $\sigma_1 = \sigma_2 = \dots = \sigma$ όπου σ μια αυθαίρετη αρχική τιμή. Το σημείο αυτό μαζί με τις παραμέτρους που αρχικοποιήσαμε αποτελεί τον πρώτο κανόνα για τη rule base μας.
- Για κάθε επόμενο σημείο x' υπολογίζουμε τον τύπο $|f(x|\theta) - y|$ όπου:

$$f(x|\theta) = \frac{\sum_{i=1}^R y_i \prod_{j=1}^n \exp\left(-\frac{1}{2} \left(\frac{x_j - c_j^i}{2\sigma}\right)^2\right)}{\sum_{i=1}^R \prod_{j=1}^n \exp\left(-\frac{1}{2} \left(\frac{x_j - c_j^i}{2\sigma}\right)^2\right)} \quad (24)$$

με βάση την υπάρχουσα rule base.

- Αν $|f(x|\theta) - y| \leq e_f$, όπου e_f θετική σταθερά, τότε δεν κάνουμε καμία αλλαγή στη rule base.
- Αλλιώς, προσθέτουμε ένα νέο rule με $b = y'$, $c = (x'_1, \dots, x'_n)$. Τα σ για τον νέο αυτό κανόνα υπολογίζονται με τον παρακάτω τρόπο
Για κάθε διάσταση i βρίσκουμε

$$n_j^* = \operatorname{argmin}\{|c_j^i - c_j^{n_i}| : i = 1, 2, \dots, R, i \neq j\}$$

όπου R το πλήθος των rule της υπάρχουσας rule base. Το σ για τον rule i στη διάσταση j υπολογίζεται από τον τύπο (25):

$$\sigma_j^i = \frac{1}{W} |c_j^i - c_j^{n_i}| \quad (25)$$

Πρέπει εδώ να σημειώσουμε πως το εκάστοτε “σί” δεν μπορεί να πάρει τη τιμή μηδέν. Ο λόγος είναι ότι η μέθοδος MLFE ορίζει η συλλογή από input data, που έχουμε για να φτιάξουμε τη βάση, περιέχει σημεία που διαφέρουν μεταξύ τους. Όχι μόνο δεν πρέπει να ταυτίζονται, αλλά για παράδειγμα να μην υπάρχουν ούτε δυο σημεία που $x = (x_1, x_2, \dots, x_n, y)$ $x' = (x'_1, x'_2, \dots, x'_n, y')$ $x_i = x'_i$ για κάποιο i

Αν όμως δεν ισχύει αυτός ο περιορισμός, τότε μπορεί το σ να μηδενιστεί. Για το λόγο αυτό θα του δίνουμε αυθαίρετα μια μικρή τιμή σ^i .

5.4 ΕΠΙΛΟΓΟΣ

Αφού λοιπόν βρούμε τους fuzzy rules και υπολογίσουμε τις membership functions, θα έχουμε ολοκληρώσει την κατασκευή της βάσης γνώσης. Σε επόμενο στάδιο, θα κλιθούμε να εφαρμόσουμε ένα σημείο στην rule base και να πάρουμε κάποιο αποτέλεσμα. Στο κεφάλαιο 6 θα παρουσιάσουμε το δικό μας σενάριο στο οποίο εφαρμόσαμε όλες τις τεχνικές που είδαμε μέχρι στιγμής πάνω στον έξυπνο πράκτορα μας για να του παρέχουμε τη γνώση ώστε να λαμβάνει αποφάσεις αυτόνομα.

ΚΕΦΑΛΑΙΟ 6

ΑΠΟΤΕΛΕΣΜΑΤΑ

Στο τελευταίο κεφάλαιο θα περιγράψουμε το σενάριο μας. Πως το υλοποιήσαμε και τι παραδοχές κάναμε. Θα εξετάσουμε το μοντέλο αγοραστή-πωλητή (buyer-seller), χρησιμοποιώντας όλους τους αλγόριθμους που παρουσιάσαμε σε προηγούμενα κεφάλαια. Μέσα από πολλές εκτελέσεις θα κάνουμε μια σύγκριση των αποτελεσμάτων που πήραμε. Τέλος θα βγάλουμε γενικά συμπεράσματα για κάθε αλγόριθμο.

6.1 ΓΕΝΙΚΗ ΠΕΡΙΓΡΑΦΗ

Το σενάριο μας περιλαμβάνει την επικοινωνία μεταξύ πολλών αγοραστών και πωλητών. Για λόγους απλότητας, θα μελετήσουμε το διάλογο που έχει ένας αγοραστής με ένα πωλητή. Οι παραδοχές που κάνουμε είναι οι εξής:

- Ο αγοραστής έχει αρχικά στη διάθεση του ένα σύνολο σημείων εισόδου.
- Τα σημεία αυτά έχουν 6 διαστάσεις και είναι οι εξής:
 - Relevance factor (r): μας δείχνει πόσο σχετικό είναι το προϊόν με τους στόχους του αγοραστή.
 - Price Difference (d): μας δείχνει τη διαφορά ανάμεσα στην τιμή που προτείνει ο seller με την τιμή που πρόκειται να προτείνει ο αγοραστής.
 - Belief (b): είναι η πίστη που έχει ο αγοραστής ότι ο πωλητής θα σταματήσει την διαδικασία.
 - Time difference (t): η διαφορά του τρέχοντος χρόνου με το deadline που έχει ο αγοραστής.
 - Valuation (V): δείχνει την αξία του προϊόντος στο μυαλό του αγοραστή.
 - Acceptance Degree (AD): ο βαθμός αποδοχής της προσφοράς μιας συγκεκριμένης τιμής.

Τα δεδομένα αυτά τα χρησιμοποιεί ο αγοραστής για να κατασκευάσει μια fuzzy rule base. Οι τεχνικές που έχει να επιλέξει είναι Subtractive, Agglomerative, FCM, HCM, NNC, LFE, MLFE. Ο πωλητής δεν έχει να χτίσει κάποια rule base, μιας και δεν παίρνει αποφάσεις. Ένας αγοραστής μπαίνει σε μια αγορά και μαθαίνει από κάποια οντότητα/μεσολαβητή ποιοι πωλητές υπάρχουν. Στη συνέχεια, ο αγοραστής κάνει αρχικά μια χειραψία μέσω μηνυμάτων PROPOSE και ACCEPT PROPOSAL με τον

κάθε πωλητή ξεχωριστά. Ο εκάστοτε πωλητής του προτείνει μια τιμή για προσφορά. Την τιμή αυτή την υπολογίζει τυχαία. Ο αγοραστής λαμβάνει αυτήν τη τιμή και παράγει μια πεντάδα (r, d, b, t, V). Την δίνει ως είσοδο στη rule base και παίρνει μια τιμή ως αποτέλεσμα για το AD. Αν το AD έχει τιμή μεγαλύτερη ή ίση του 70, τότε κλείνεται συμφωνία. Αν το AD είναι μικρότερο του 70, τότε ο αγοραστής αντιπροτείνει με την σειρά του μια τιμή προς τον πωλητή. Ο πωλητής είτε δέχεται την αντιπρόταση ή προτείνει εκ νέου κάποια προσφορά. Αν δεν κλειστεί μια συμφωνία μέσα σε ένα καθορισμένο από τον αγοραστή αριθμό προσφορών, τότε η επικοινωνία διακόπτεται. Ακολουθούν οι αλγόριθμοι που περιγράφουν τη λειτουργία καθενός από τους αγοραστές και πωλητές.

Αλγόριθμος για την πλευρά του αγοραστή.

ΕΙΣΟΔΟΣ : n δεδομένα

ΑΡΧΗ

B1: επέλεξε αλγόριθμο κατασκευής fuzzy rule base

B2: χτίσε τη rule base χρησιμοποιώντας τα δεδομένα εισόδου

B3: ψάξε όλους τους seller agents που είναι εγγεγραμμένοι στην αγορά

ΓΙΑ ΚΑΘΕ SELLER

ΕΠΑΝΑΛΑΒΕ

B4: στείλε μήνυμα PROPOSE στον seller

B5: περίμενε μήνυμα ACCEPT PROPOSAL από τον seller

B6: παρήγαγε τυχαίες τιμές για τις παραμέτρους VALUATION και DEADLINE

B7: στείλε μήνυμα REQUEST στον seller

ΕΠΑΝΑΛΑΒΕ

B8: περίμενε μήνυμα PROPOSE από τον seller που περιέχει μια προτεινόμενη τιμή για το προϊόν

B9: υπολόγισε τιμές για RELEVANCE FACTOR, PRICE DIFFERENCE, TIME DIFFERENCE και BELIEF

B10: εφάρμοσε τις τιμές αυτές στη rule base και πάρε ως αποτέλεσμα τιμή για το AD

B11:

AN $AD \geq 70$

στείλε μήνυμα AGREE στον seller και έξοδος

ΑΛΛΙΩΣ

στείλε REFUSE στον seller και πρότεινε του εσύ μια τιμή

ΑΝ δεχτείς AGREE

έξοδος

ΑΛΛΙΩΣ

Μείωσε DEADLINE και πήγαινε στο βήμα B8

ΜΕΧΡΙΣ ΟΤΟΥ DEADLINE == 0

ΤΕΛΟΣ

Αλγόριθμος για την πλευρά του πωλητή

ΑΡΧΗ

B1: εγγράψου στο DF

ΟΣΟ ΛΑΜΒΑΝΕΙΣ PROPOSE ΑΠΟ BUYERS

ΕΠΑΝΑΛΑΒΕ

B2: στείλε μήνυμα ACCEPT PROPOSAL στον buyer

B3: περίμενε μήνυμα REQUEST από τον buyer

B4: στείλε μήνυμα PROPOSE με μια προτεινόμενη τιμή για το προϊόν,
που υπολόγισες τυχαία

B5:

ΑΝ έλαβες μήνυμα AGREE

έξοδος

ΑΛΛΙΩΣ

Υπολόγισε πάλι τυχαία τιμή για το προϊόν

ΑΝ ο buyer σου πρότεινε μικρότερη

στείλε AGREE και έξοδος

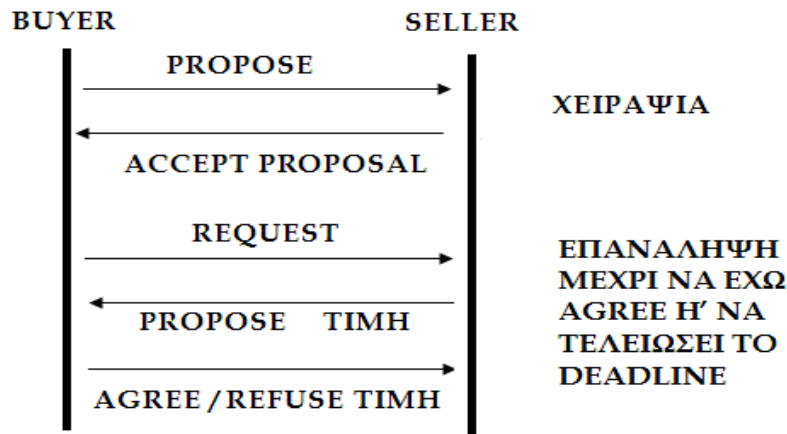
ΑΛΛΙΩΣ

Πήγαινε στο βήμα B4

ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ

Η ανταλλαγή μηνυμάτων μεταξύ των δυο μερών φαίνεται πιο απλά στο σχήμα που ακολουθεί (Εικόνα 30).



Εικόνα 30. Επικοινωνία buyer-seller.

Ο λόγος που βάλαμε ένα άνω φράγμα στην τιμή AD είναι για να μελετήσουμε αν ο εκάστοτε αλγόριθμος, που επιλέξαμε για το χτίσιμο της rule base, μπορεί να προσεγγίσει όλες τις τιμές για το AD από 1 έως 100. Στο Κεφάλαιο 5 είπαμε ότι όταν προσπαθούμε να υπολογίσουμε την μορφή των συναρτήσεων συμμετοχής, κάποιες φορές το κάνουμε κατά προσέγγιση και έτσι υπάρχει δυνατότητα να χάνουμε στην ακρίβεια. Και εμείς με αυτόν τον τρόπο θέλουμε να δούμε αν υπάρχει προσέγγιση σε βέλτιστα αποτελέσματα. Βέβαια, δεν περιμένουμε συνέχεια το AD να πάρει τιμές πάνω από 70. Σε επόμενες ενότητες θα δούμε αν στατιστικά το AD εμφανίζει μεγάλες τιμές σε ικανοποιητικό πλήθος.

6.2 ΣΥΓΚΡΙΤΙΚΗ ΑΠΟΤΙΜΗΣΗ ΑΛΓΟΡΙΘΜΩΝ

Αρχικά θα συγκρίνουμε τις διάφορες τεχνικές ως προς το χρόνο. Συγκεκριμένα, έχοντας υλοποιήσει τους αλγόριθμους και τρέχοντας πραγματικές εφαρμογές buyer-seller μετρήσαμε το χρόνο που χρειάζεται για να χτιστεί η fuzzy rule base έχοντας ως είσοδο δοκιμαστικά σημεία.

Πίνακας 14. Χρόνος εκτέλεσης αλγορίθμων.

| Αλγόριθμος | Χρόνος (ms) | Αλγόριθμος | Χρόνος (ms) |
|------------------------|-------------|---------------|-------------|
| Subtractive $r_a=0.5$ | 78 | NNC | 50 |
| Subtractive $r_a=0.75$ | 46 | LFE | 50 |
| Subtractive $r_a=1.0$ | 46 | MLFE | 60 |
| FCM | 3000 | Agglomerative | 281 |
| HCM | 60 | | |

Παρατηρούμε πως:

- Ο FCM είναι πολύ αργός. Παρότι είναι επέκταση του HCM και ακολουθεί την ίδια φιλοσοφία, έχει να εκτελέσει πολλές πράξεις για το βαθμό συμμετοχής και έτσι χάνει χρόνο.
- Ο Subtractive όσο περισσότερο ακριβής θέλουμε να είναι (r_a κοντά στο 0.5) τόσο αυτό μας κοστίζει.
- Ο Agglomerative είναι και αυτός σχετικά αργός γιατί έχει να χτίσει το δένδρο με τα clusters. Συνεπώς για μεγάλο όγκο αρχικών σημείων εισόδου, το κόστος του μπορεί να γίνει απαγορευτικό.
- Ο NNC παρότι ανήκει στην κατηγορία των αλγορίθμων που παράγουν clusters (σε αντίθεση με τους LFE, MLFE), είναι σχετικά γρήγορος. Ο λόγος είναι ότι δεν προσπαθεί να βρει ανάμεσα σε όλους τους δυνατούς συνδυασμούς διαμέρισης των σημείων αυτόν που είναι ο καλύτερος. Αυτό που κάνει είναι να ενώνει σημεία που η μεταξύ τους απόσταση να είναι μικρότερη από e_f .
- Ο LFE δεν ανήκει στην κατηγορία των αλγορίθμων συσταδοποίησης. Προσπαθεί να παράγει κανόνες χωρίς να έχει στόχο να βελτιστοποιήσει κάποιος παραμέτρους, όπως κάποια συνάρτηση κόστους ή την κατανομή των σημείων. Είναι αρκετά γρήγορος στην εκτέλεση, όμως απαιτεί γνώση για την κατανομή των σημείων που του δίνουμε ως είσοδο. Αυτό όμως τις περισσότερες φορές είναι ανέφικτο.
- Ο MLFE ανήκει στην ίδια κατηγορία με τον LFE. Δεν χρειάζεται να έχουμε εμείς γνώση για την κατανομή των σημείων και μπορεί από μόνος του να παράγει ασαφείς κανόνες και τις συναρτήσεις συμμετοχής τους. Παρατηρούμε ακόμη ότι και αυτός είναι πολύ γρήγορος στην εκτέλεση.

Ένα ακόμη σημείο που θέλουμε να συγκρίνουμε είναι τα αποτελέσματα που δίνουν οι διάφοροι αλγόριθμοι, όταν πάμε να εφαρμόσουμε στη rule base ένα σημείο. Πήραμε 20 σημεία όπως φαίνεται στον πίνακα παρακάτω (Πίνακας 15).

Πίνακας 15. Δείγμα από διαδοχικά σημεία εισόδου.

| r | d | b | t | V |
|----------|----------|----------|----------|----------|
| 0.90 | 0.01 | 0.57 | 0.88 | 0.23 |
| 0.30 | 0.19 | 0.92 | 0.67 | 0.79 |
| 0.81 | 0.32 | 0.46 | 0.68 | 0.08 |
| 0.95 | 0.66 | 0.71 | 0.04 | 0.51 |
| 0.51 | 0.33 | 0.34 | 0.20 | 0.3 |

| | | | | |
|------|------|------|------|------|
| 0.88 | 0.04 | 0.80 | 0.86 | 0.33 |
| 0.85 | 0.68 | 0.41 | 0.03 | 0.48 |
| 0.90 | 0.37 | 0.86 | 0.01 | 0.79 |
| 0.14 | 0.22 | 0.95 | 0.36 | 0.99 |
| 0.85 | 0.51 | 0.58 | 0.51 | 0.58 |
| 0.81 | 0.09 | 0.96 | 0.49 | 0.68 |
| 0.48 | 0.24 | 0.81 | 0.58 | 0.98 |
| 0.33 | 0.35 | 0.89 | 0.63 | 0.46 |
| 0.05 | 0.53 | 0.27 | 0.53 | 0.43 |
| 0.03 | 0.52 | 0.00 | 0.43 | 0.15 |
| 0.90 | 0.17 | 0.58 | 0.84 | 0.18 |
| 0.93 | 0.34 | 0.23 | 0.24 | 0.30 |
| 0.36 | 0.5 | 0.84 | 0.66 | 0.37 |
| 0.45 | 0.27 | 0.13 | 0.11 | 0.1 |
| 0.94 | 0.05 | 0.21 | 0.07 | 0.35 |

Τις πεντάδες αυτές τις εφαρμόσαμε στη rule base για να πάρουμε τιμή για το AD. Τα αποτελέσματα που δίνει για το AD ο κάθε αλγόριθμος δίνονται στους επόμενους πίνακες (Πίνακας 16-α, Πίνακας 16-β).

Πίνακας 16-α. Αποτελέσματα στην έξοδο.

| Subtractive $r_a = 0.5$ | Subtractive $r_a = 0.75$ | Subtractive $r_a = 1.0$ | Agglomerative | FCM |
|-----------------------------------|------------------------------------|-----------------------------------|----------------------|------------|
| 74.03 | 71,96 | 53,65 | 77,86 | 60,81 |
| 35.34 | 32,84 | 53,40 | 32,45 | 39,08 |
| 60.51 | 59,66 | 46,87 | 63,17 | 52,46 |
| 83.45 | 78,39 | 71,34 | 87,50 | 68,87 |
| 37.90 | 43,63 | 39,85 | 23,26 | 29,16 |
| 77.92 | 78,00 | 60,68 | 78,08 | 62,72 |
| 48.38 | 53,75 | 55,69 | 35,80 | 36,74 |
| 95.76 | 94,57 | 78,98 | 94,57 | 76,33 |
| 30.47 | 28,03 | 52,30 | 40,25 | 31,08 |
| 79.38 | 70,00 | 62,55 | 88,05 | 64,22 |
| 84.73 | 79,38 | 73,73 | 94,92 | 72,30 |
| 48.80 | 48,95 | 67,66 | 40,25 | 42,73 |
| 39.00 | 34,27 | 43,63 | 28,43 | 38,21 |
| 13.41 | 16,96 | 21,10 | 16,63 | 20,34 |
| 12.54 | 14,26 | 17,57 | 14,52 | 20,55 |
| 74.73 | 72,18 | 53,00 | 77,98 | 61,93 |
| 67.94 | 63,74 | 58,20 | 69,33 | 52,25 |
| 37.30 | 33,20 | 40,04 | 28,24 | 35,73 |
| 33.33 | 28,21 | 29,87 | 23,05 | 27,13 |
| 70.22 | 70,12 | 64,09 | 69,33 | 55,05 |

Πίνακας 16-β. Αποτελέσματα στην έξοδο.

| HCM | LFE | MLFE | NNC |
|-------|-------|-------|-------|
| 64,72 | 84,27 | 77,85 | 77,04 |
| 45,82 | 43,17 | 36,16 | 34,93 |
| 48,59 | 70,98 | 21,04 | 61,83 |
| 57,86 | 88,16 | 85,00 | 84,76 |
| 28,75 | 45,98 | 10,72 | 36,97 |
| 68,46 | 88,72 | 81,06 | 79,93 |
| 28,96 | 64,62 | 47,74 | 47,83 |
| 67,21 | 89,38 | 99,93 | 98,83 |
| 40,56 | 32,49 | 27,28 | 28,56 |
| 48,81 | 79,60 | 47,83 | 82,89 |
| 66,24 | 85,89 | 92,98 | 89,29 |
| 52,88 | 59,71 | 48,95 | 48,64 |
| 45,97 | 43,94 | 40,19 | 36,75 |
| 15,70 | 8,34 | 10,03 | 14,02 |
| 14,62 | 5,26 | 10,09 | 12,60 |
| 65,54 | 84,23 | 77,88 | 77,30 |
| 43,49 | 56,05 | 68,09 | 69,97 |
| 37,25 | 45,26 | 27,40 | 37,08 |
| 29,27 | 38,35 | 26,91 | 34,11 |
| 47,93 | 59,21 | 72,15 | 71,86 |

Βλέπουμε πως παρότι χτίσαμε την rule base με τα ίδια αρχικά σημεία, τα αποτελέσματα των αλγορίθμων είναι διαφορετικά. Αξιοσημείωτο είναι πως οι FCM, HCM διαφέρουν κατά πολύ από τις άλλες τεχνικές. Οι τιμές που δίνουν ως αποτέλεσμα για το AD τείνουν να είναι κοντά στο 50. Δύσκολα πιάνουν τιμές χαμηλές (< 20) ή πολύ υψηλές (> 80). Ο λόγος είναι ότι σε αυτούς τους δύο αλγορίθμους πολλές φορές γίνεται προσέγγιση των συναρτήσεων συμμετοχής με συνέπεια να χάνουμε στην ακρίβεια των τιμών. Ακόμη να τονίσουμε πως με μια εκτέλεση βγάλαμε αποτελέσματα για το AD, συνήθως ενδείκνυται να έχουμε πολλές επαναλήψεις της ίδιας εισόδου για να πάρουμε καλά αποτελέσματα γιατί οι αρχικοποιήσεις των clusters γίνονται τυχαία.

Οι έξοδοι του Subtractive εξαρτώνται πολύ από την τιμή του r_a . Για τιμές κοντά στο 0.5, μπορεί να προσεγγίσει ακραίες τιμές στο AD. Όσο το r_a αυξάνει, αυτό συνεπάγεται ότι τα clusters γίνονται μεγαλύτερα, είναι λιγότερα και έχουμε μικρότερη ακρίβεια τιμών.

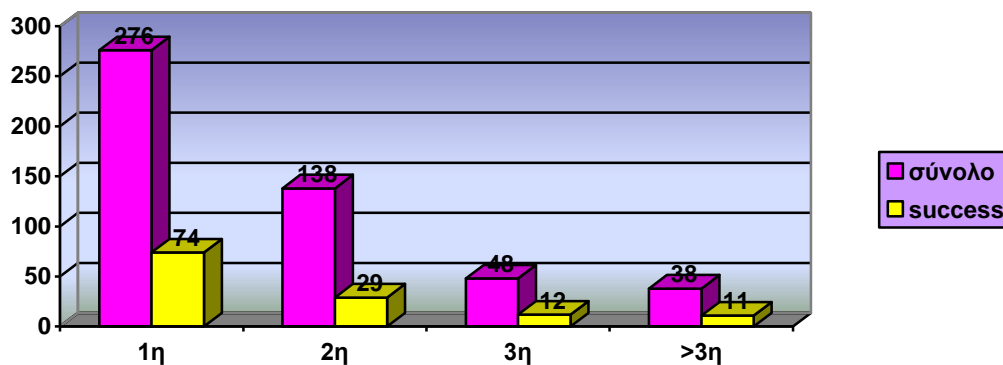
Ο LFE αγγίζει ένα μεγάλο εύρος τιμών. Καμία φορά πιο πολύ και από τον Subtractive με $r_a = 0.5$ που αργεί πιο πολύ. Όμως ο LFE έχει το μεγάλο μειονέκτημα ότι πρέπει να του δίνουμε τις συναρτήσεις συμμετοχής. Κάτι τέτοιο είναι πολύ δύσκολο να το παραβλέψουμε.

Το τελευταίο σημείο το οποίο θα εξετάσουμε για αυτούς τους αλγόριθμους είναι το ποσοστό στο οποίο είχαμε κλείσιμο μιας συμφωνίας επειδή το AD ήταν πάνω από 70 (και όχι επειδή ο πωλητής συμφώνησε σε μια τιμή που του αντιπρότεινε ο αγοραστής). Κάθε συμφωνία που κλείστηκε με τιμή για το AD πάνω από 70 θα την ονομάζουμε “success”. Για κάθε αλγόριθμο κάναμε 100, 200 και 500 συνδιαλλαγές μεταξύ αγοραστών και πωλητών. Σε αυτές μετρήσαμε πόσες φορές ο πωλητής πρότεινε τιμές μέχρι να κλείσει η συμφωνία (με την 1^η, με την 2^η κοκ). Σε κάθε μια, μετρήσαμε πόσες φορές ο αγοραστής συμφώνησε από μόνος του ($AD \geq 70$) για να κλείσει η συμφωνία. Τα αποτελέσματα φαίνονται στους ακόλουθους πίνακες

Πίνακας 17. Τα αποτελέσματα των συναλλαγών με τη χρήση του Subtractive με r_a 0.5.

| | 1 ^η | success | 2 ^η | success | 3 ^η | success | > 3 ^η | success |
|-----|----------------|---------|----------------|---------|----------------|---------|------------------|---------|
| 100 | 58 | 16 | 24 | 6 | 11 | 4 | 7 | 3 |
| 200 | 115 | 33 | 51 | 13 | 21 | 5 | 13 | 2 |
| 500 | 276 | 74 | 138 | 29 | 48 | 12 | 38 | 11 |

Η Εικόνα 31 παρουσιάζει ένα γράφημα που έγινε για τη γραμμή με τις 500 συναλλαγές. Προσπαθούμε να αποτυπώσουμε για κάθε κατηγορία (με την 1^η, με την 2^η κλπ) πόσες συμφωνίες είχαμε και από αυτές πόσα είναι τα successes.



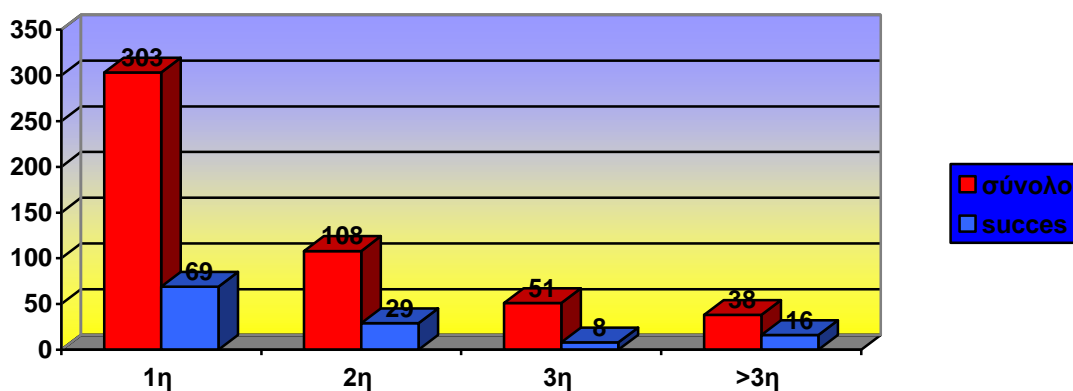
Εικόνα 31. Γραφική αναπαράσταση πλήθους success για τον Subtractive με r_a 0.5.

Δεν πάμε να συγκρίνουμε τις κατηγορίες και τα success μεταξύ τους. Ούτε θέλουμε να συγκρίνουμε πόσες φορές έχουμε πχ με την 1^η success. Το επιθυμητό σε κάθε αλγόριθμο είναι να υπάρχουν αρκετά success σε κάθε κατηγορία. Το να μην είναι όλα success δεν είναι λάθος, γιατί το AD θέλουμε να παίρνει όλες τις τιμές από 1 έως 100. Στόχος μας είναι σε κάθε αλγόριθμο να πιάνονται και οι τιμές $AD > 70$.

Πίνακας 18. Τα αποτελέσματα των συναλλαγών με τη χρήση του Subtractive με r_a

0.75.

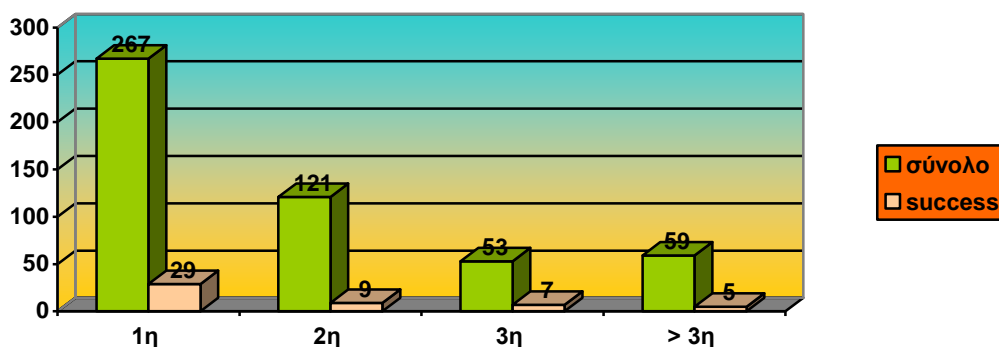
| | 1 ^η | success | 2 ^η | success | 3 ^η | success | > 3 ^η | success |
|-----|----------------|---------|----------------|---------|----------------|---------|------------------|---------|
| 100 | 57 | 9 | 24 | 7 | 12 | 2 | 7 | 2 |
| 200 | 116 | 33 | 50 | 15 | 26 | 8 | 8 | 3 |
| 500 | 303 | 69 | 108 | 29 | 51 | 8 | 38 | 16 |



Εικόνα 32. Γραφική αναπαράσταση πλήθους success για τον Subtractive με r_a 0.75.

Πίνακας 19. Τα αποτελέσματα των συναλλαγών με τη χρήση του Subtractive με r_a 1.0

| | 1 ^η | success | 2 ^η | success | 3 ^η | success | > 3 ^η | success |
|-----|----------------|---------|----------------|---------|----------------|---------|------------------|---------|
| 100 | 54 | 7 | 21 | 0 | 18 | 1 | 7 | 0 |
| 200 | 101 | 9 | 56 | 6 | 24 | 3 | 19 | 3 |
| 500 | 267 | 29 | 121 | 9 | 53 | 7 | 59 | 5 |

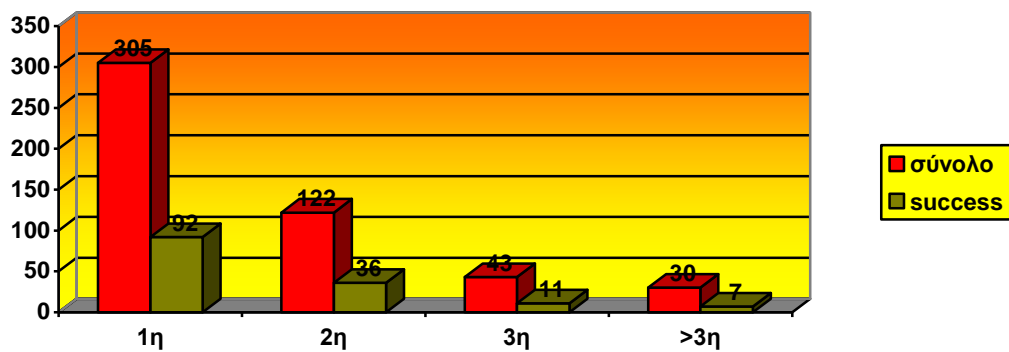


Εικόνα 33. Γραφική αναπαράσταση πλήθους success για τον Subtractive με r_a 1.0.

Από τις Εικόνες 31, 32, 33 επαληθεύονται οι παρατηρήσεις μας για τον Subtractive και συγκεκριμένα για την τιμή του r_a . Παρότι είχαμε διαφορετικές εκτελέσεις για τις διαφορετικές τιμές του r_a , δηλαδή δεν είχαμε τα ίδια σημεία εισόδου και να βλέπαμε διαφορά στις τιμές του AD, παρατηρούμε πως η παράμετρος success έχει σημαντικό ποσοστό στο σύνολο των συναλλαγών που έγιναν με την 1^η, 2^η κοκ όταν το r_a είναι μικρό. Αυτό σημαίνει αυτόματα ότι μπορεί ο αλγόριθμος να είναι ευέλικτος στα αποτελέσματα που δίνει.

Πίνακας 20. Τα αποτελέσματα των συναλλαγών με τη χρήση του Agglomerative.

| | 1 ^η | success | 2 ^η | success | 3 ^η | success | > 3 ^η | success |
|-----|----------------|---------|----------------|---------|----------------|---------|------------------|---------|
| 100 | 63 | 19 | 24 | 7 | 6 | 1 | 7 | 1 |
| 200 | 133 | 47 | 38 | 15 | 13 | 5 | 16 | 5 |
| 500 | 305 | 92 | 122 | 36 | 43 | 11 | 30 | 7 |



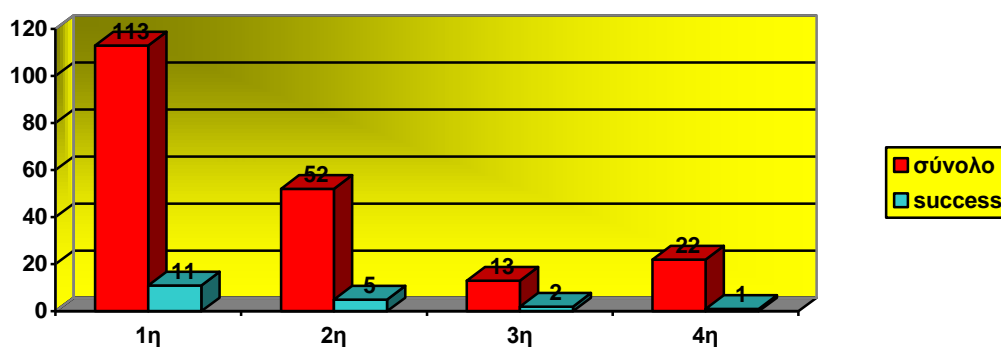
Εικόνα 34. Γραφική αναπαράσταση πλήθους success για τον Agglomerative.

Πίνακας 21. Τα αποτελέσματα των συναλλαγών με τη χρήση του FCM.

| | 1 ^η | success | 2 ^η | success | 3 ^η | success | > 3 ^η | success |
|-----|----------------|---------|----------------|---------|----------------|---------|------------------|---------|
| 100 | 55 | 8 | 24 | 5 | 14 | 3 | 7 | 0 |
| 200 | 113 | 11 | 52 | 5 | 13 | 2 | 22 | 1 |

Πίνακας 18

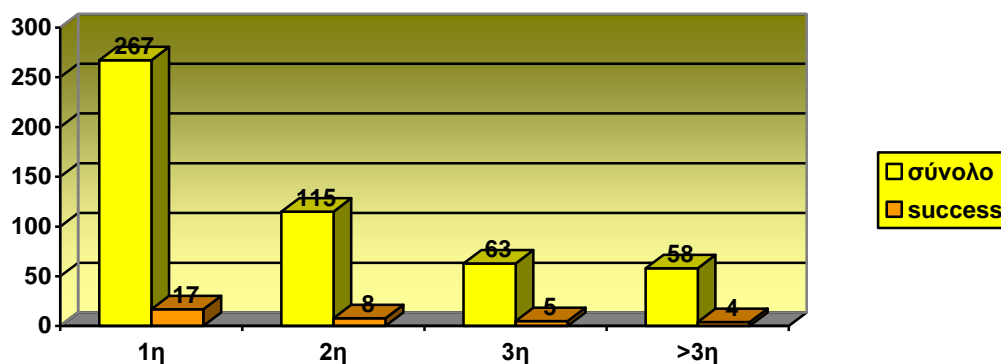
Δεν προσπαθήσαμε να έχουμε 500 συναλλαγές μιας και ο FCM είναι ιδιαίτερα αργός για να χτίσει τη βάση κανόνων



Εικόνα 35. Γραφική αναπαράσταση πλήθους success για τον FCM.

Πίνακας 22. Τα αποτελέσματα των συναλλαγών με τη χρήση του HCM.

| | 1 ^η | success | 2 ^η | success | 3 ^η | success | > 3 ^η | success |
|-----|----------------|---------|----------------|---------|----------------|---------|------------------|---------|
| 100 | 42 | 4 | 29 | 2 | 12 | 1 | 17 | 1 |
| 200 | 104 | 5 | 54 | 3 | 19 | 0 | 23 | 1 |
| 500 | 264 | 17 | 115 | 8 | 63 | 5 | 58 | 4 |

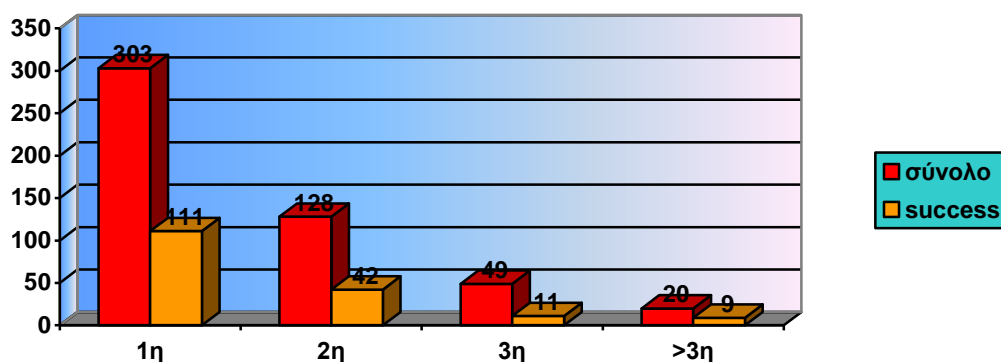


Εικόνα 36. Γραφική αναπαράσταση πλήθους success για τον HCM.

Κοιτώντας τις Εικόνες 35, 36 παρατηρούμε πως λίγες φορές σημειώνονται συναλλαγές που χαρακτηρίζονται ως success για τους αλγορίθμους FCM, HCM. Αυτό αποτελεί και επαλήθευση στο γεγονός ότι είναι δύσκολο να βγάλουν με μια μόνο εκτέλεση επιθυμητά αποτελέσματα συν το γεγονός ότι γίνονται προσεγγίσεις στις συναρτήσεις συμμετοχής με αποτέλεσμα να μην είναι ακριβή τα αποτελέσματα.

Πίνακας 23. Τα αποτελέσματα των συναλλαγών με τη χρήση του LFE.

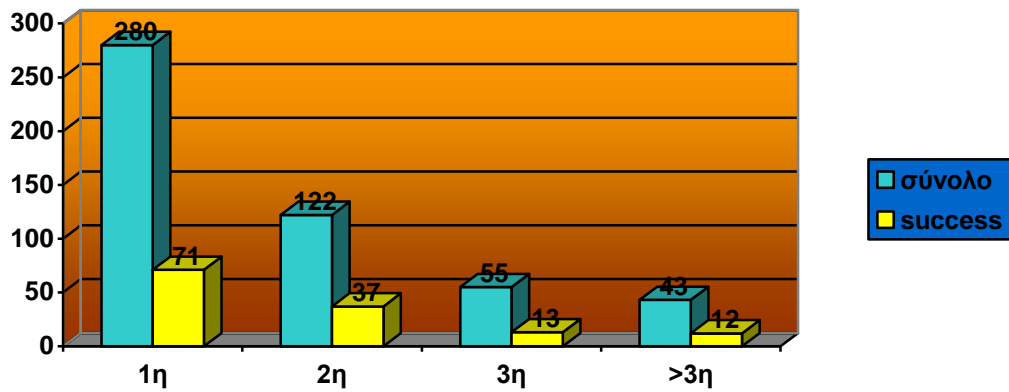
| | 1 ^η | success | 2 ^η | success | 3 ^η | success | > 3 ^η | success |
|-----|----------------|---------|----------------|---------|----------------|---------|------------------|---------|
| 100 | 61 | 15 | 23 | 4 | 8 | 1 | 8 | 3 |
| 200 | 118 | 46 | 50 | 17 | 15 | 7 | 17 | 8 |
| 500 | 303 | 111 | 128 | 42 | 49 | 11 | 20 | 9 |



Εικόνα 37. Γραφική αναπαράσταση πλήθους success για τον LFE.

Πίνακας 24. Τα αποτελέσματα των συναλλαγών με τη χρήση του MLFE.

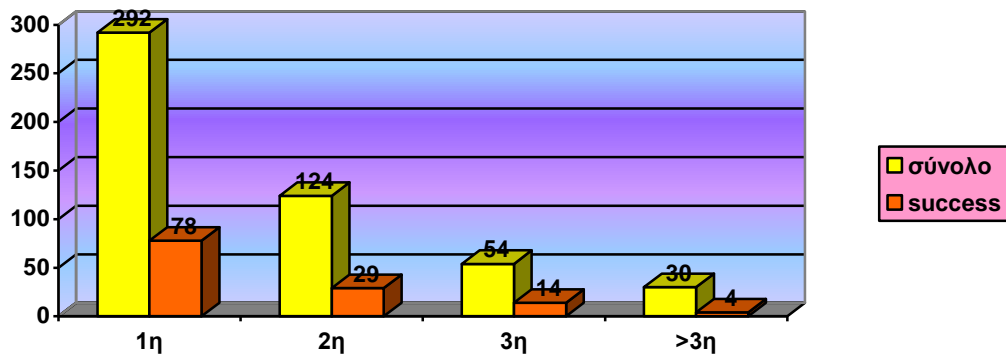
| | 1 ^η | success | 2 ^η | success | 3 ^η | success | > 3 ^η | success |
|-----|----------------|---------|----------------|---------|----------------|---------|------------------|---------|
| 100 | 54 | 16 | 29 | 6 | 9 | 2 | 8 | 7 |
| 200 | 115 | 32 | 45 | 7 | 24 | 11 | 16 | 5 |
| 500 | 280 | 71 | 122 | 37 | 55 | 13 | 43 | 12 |



Εικόνα 38. Γραφική αναπαράσταση πλήθους success για τον MLFE.

Πίνακας 25. Τα αποτελέσματα των συναλλαγών με τη χρήση του NNC.

| | 1 ^η | success | 2 ^η | success | 3 ^η | success | > 3 ^η | success |
|-----|----------------|---------|----------------|---------|----------------|---------|------------------|---------|
| 100 | 51 | 14 | 25 | 2 | 15 | 4 | 9 | 2 |
| 200 | 119 | 21 | 45 | 15 | 21 | 9 | 15 | 4 |
| 500 | 292 | 78 | 124 | 29 | 54 | 14 | 30 | 4 |



Εικόνα 39. Γραφική αναπαράσταση πλήθους success για τον NNC.

Και στις τρεις Εικόνες 37, 38, 39 παρατηρούμε μεγάλο ποσοστό για το success. Ίσως μεγαλύτερο και από την περίπτωση του Subtractive με r_a 0.5.

6.3 ΣΥΜΠΕΡΑΣΜΑΤΑ

Από την προηγούμενη ενότητα και από τις υλοποιήσεις που κάναμε για τις διάφορες τεχνικές βγάλαμε τα εξής συμπεράσματα για τους αλγόριθμους μας

Subtractive

Ο αλγόριθμος αυτός είναι αρκετά εύκολος στο να υλοποιηθεί. Το γεγονός ότι επιλέγουμε τα υποψήφια κέντρα μέσα από τα σημεία που έχουμε μειώνει κάνει ανεξάρτητη την πολυπλοκότητα από τις διαστάσεις των σημείων. Δίνει πολύ καλά αποτελέσματα για μικρές τιμές του r_a χωρίς βέβαια να γίνεται υπερβολικά αργός. Ένα τελευταίο θετικό σημείο που έχει είναι το ότι δεν απαιτείται από εμάς να δώσουμε εκτιμώμενο πλήθος clusters.

HCM

Από τους πιο απλούς αλγόριθμους στην υλοποίηση. Γρήγορος στην εκτέλεση μιας και δεν εκτελεί μεγάλο όγκο πράξεων. Προσπαθεί να βρει την βέλτιστη κατανομή clusters με βάση τυχαίες αρχικοποιήσεις. Το γεγονός αυτό μας αναγκάζει να εκτελούμε πολλές φορές την ίδια είσοδο για να πάρουμε μια εκτίμηση για την τιμή της εξόδου. Ένα ακόμη μειονέκτημα που έχει είναι ότι δεν μπορεί να υπολογίζει από μόνος του το πλήθος clusters.

FCM

Γενικά από τους πιο δύσκολους αλγόριθμους στην κατανόηση και την υλοποίηση. Απαιτεί πολλούς υπολογισμούς. Δεν μπορεί να βρει πλήθος clusters για ένα σύνολο σημείων και ακόμη δεν μπορεί να εκτελεστεί όταν του δίνουμε μεγάλο αριθμό εμείς για το πλήθος των clusters που επιθυμούμε. Το θετικό σε αυτό τον αλγόριθμο είναι όταν κάνουμε προβολή των βαθμών συμμετοχής μας δείχνει σε ποιους κανόνες ποιες ασαφείς μεταβλητές δεν χρειάζεται να λαμβάνονται υπόψη (οι συναρτήσεις συμμετοχής εκτείνονται σε μεγάλο εύρος τιμών). Το προτέρημα του αυτό το χρησιμοποιούμε στον αλγόριθμο LFE, όπου καλούμαστε εμείς να δώσουμε τις συναρτήσεις συμμετοχής.

Agglomerative

Παρότι είναι αλγόριθμος συσταδοποίησης, δεν βλέπει την γενική κατανομή που έχουν τα σημεία αλλά με μια «κοντόφθαλμη» λογική ενώνει σημεία που μπορεί να έχουν εκείνη τη στιγμή την μικρότερη απόσταση μεταξύ τους, αλλά που μπορεί κανένας άλλος αλγόριθμος για clustering να τα ένωνε. Απαιτεί μεγάλο χώρο αποθήκευσης του δένδρου που αναπτύσσει και αυτό είναι απαγορευτικό για μεγάλο αριθμό σημείων.

NNC

Ανήκει στους αλγόριθμους συσταδοποίησης. Με ένα πέρασμα στα σημεία προσπαθεί να φτιάξει clusters που να έχουν μεταξύ τους απόσταση e_r . Γι' αυτό το λόγο είναι και πολύ γρήγορος. Από εμάς απαιτείται να κάνουμε πολλές δοκιμές για το e_r για να μπορέσουμε να τον χρησιμοποιήσουμε στο λογισμικό του πράκτορα.

LFE

Αρκετά εύκολος αλγόριθμος τόσο στο θεωρητικό του μέρος όσο και στην υλοποίηση του. Το μειονέκτημα του που μπορεί να τον αχρηστέψει είναι όπως είπαμε πολλές φορές και πιο πριν ότι θέλει γνώση των συναρτήσεων συμμετοχής για να μας κατασκευάσει τη βάση ασαφών κανόνων. Στο δικό μας σενάριο τον χρησιμοποιήσαμε σε συνεργασία με τον FCM. Θα μπορούσαμε να πούμε πως ο ένας συμπληρώνει τις αδυναμίες του άλλου με στόχο να φτιάξουμε γρήγορα μια βάση γνώσης.

MLFE

Είναι αρκετά δύσκολος στην υλοποίηση και απαιτεί από εμάς ιδιαίτερη προσοχή στις παραδοχές που θα κάνουμε. Ο λόγος είναι ότι θέτει σαν περιορισμό να μην υπάρχουν ούτε δυο σημεία που να συμπίπτουν ούτε σε μια συντεταγμένη. Ακόμη απαιτεί από εμάς πολλές δοκιμές σε τιμές των παραμέτρων του. Κατά τα άλλα είναι και αυτός αρκετά γρήγορος στην εκτέλεση.

ΑΝΑΦΟΡΕΣ

- [1]. <http://agents.umbc.edu/introduction/ao/>
- [2]. Giovanni Caire, "JADE PROGRAMMING FOR BEGINNERS", 2003
- [3]. Andries P. Engelbrecht, "Computational Intelligence: An Introduction".
- [4]. <http://www.mathworks.com/>
- [5]. http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol1/hmw/article1.html
- [6]. Dave Anderson, George Mc Neill, "ARTIFICIAL NEURAL NETWORKS TECHNOLOGY", 1992
- [7] Clustering, http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/index.html
- [8] A.K. JAIN, M.N. MURTY AND P.J. FLYNN, "Data Clustering: A Review".
- [9]. Fakhreddine Karray, "A Comparative Study of Data Clustering Techniques».
- [10]. http://www.resample.com/xlminer/help/HClst/HClst_intro.htm
- [11]. Ahmed Ismail Shihab, "Fuzzy Clustering Algorithms and their Application to Medical Image", DECEMBER 2000.
- [12]. Yunjae Jung, Hansen Park, Ding-Zhu Du, Barry L. Drake, "A Decision Criterion for the Optimal Number of Clusters in Hierarchical Clustering", Jan. 21, 2002.
- [13]. Vladimir Estivill Castro, "Why so many clustering algorithms— A Position Paper".
- [14]. Enrico ZIO, Piero BARALDI, Irina Crenguta POPESCU, "FROM FUZZY CLUSTERING TO A FUZZY RULE-BASED FAULT CLASSIFICATION MODEL", 2007
- [15]. Stephen L. Chiu, "Extracting Fuzzy Rules from Data for Function Approximation and Pattern Classification", 1997
- [16]. Kevin M. Passino, Stephen Yurkovich, "Fuzzy Control"
- [17]. Distance Metrics,
http://www.improvedoutcomes.com/docs/WebSiteDocs/Clustering/Clustering_Parameters/Distance_Metrics_Overview.htm
- [18]. www.wikipedia.org

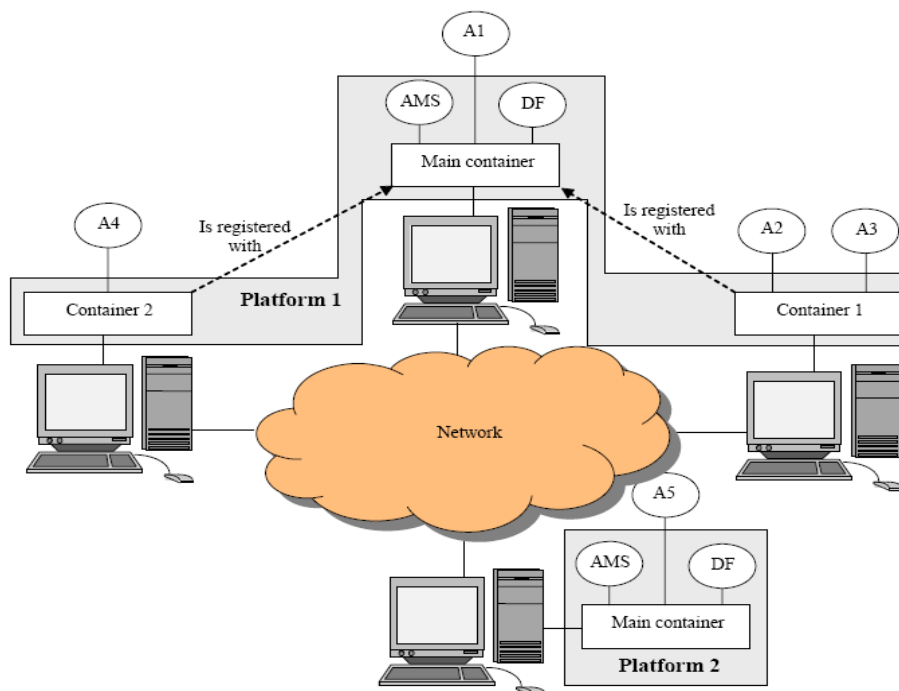
ΠΑΡΑΡΤΗΜΑ Α

JADE

Το Jade αποτελεί middleware [2] μέσω του οποίου μπορούν πράκτορες να επικοινωνούν σε κατανεμημένα περιβάλλοντα. Το Jade είναι γραμμένο σε Java και περιλαμβάνει:

- Περιβάλλον (Runtime environment) στο οποίο εκτελούνται και επικοινωνούν οι πράκτορες.
- Βιβλιοθήκη κλάσεων.
- Γραφικό περιβάλλον για να έχουμε οπτική αναπαράσταση των πρακτόρων.

Η εικόνα παρακάτω μας δείχνει ένα τυπικό Jade runtime environment.

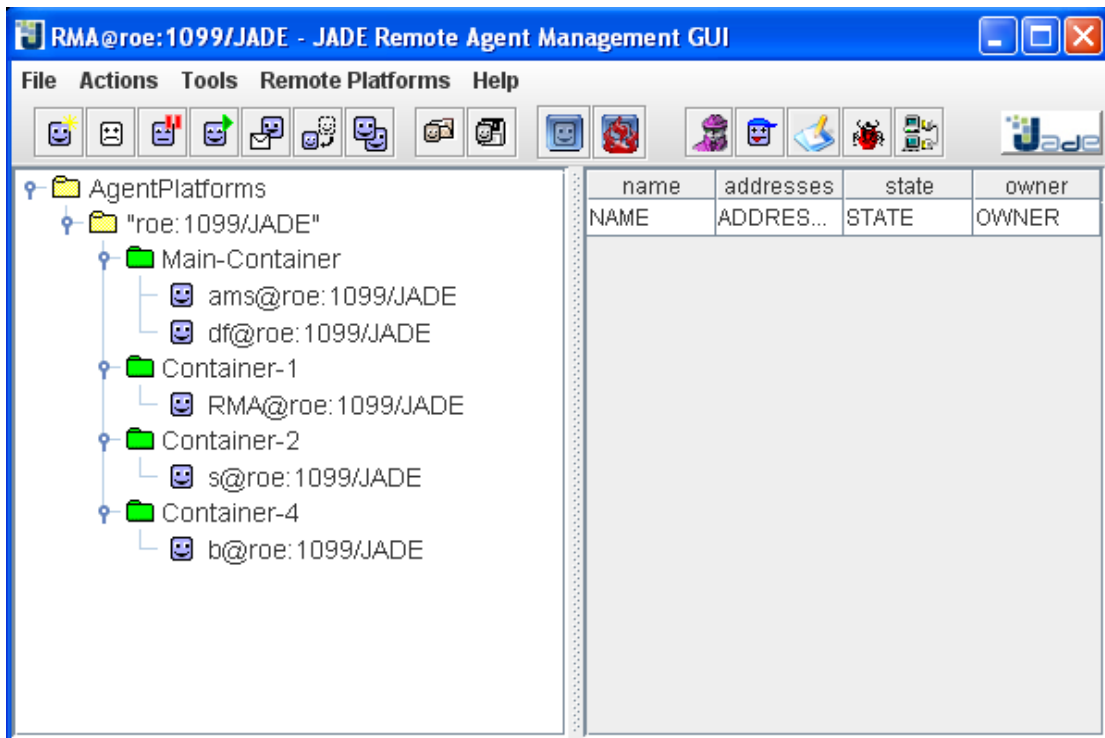


Εικόνα 40. Το runtime environment του JADE.

Κάθε πράκτορας (A1, A2,...) ανήκει σε ένα container, και κάθε container με τη σειρά του ανήκει σε μια Platform. Μια Platform πρέπει απαραίτητα να περιέχει ένα main container. Οι agents AMS, DF υπάρχουν από πριν στο σύστημα και διαχειρίζονται ζητήματα ονοματολογίας agents, καθώς επίσης παρέχουν και κάποιες υπηρεσίες. Στο δικό μας σενάριο παίζουν τον ρόλο των μεσολαβητών πρακτόρων.

Όπως είπαμε και στην αρχή, το Jade έχει βιβλιοθήκη κλάσεων με τις οποίες γράφονται τα προγράμματα των πρακτόρων. Ακόμη, υπάρχει η δυνατότητα γραφικής αναπαράστασης ενός στιγμιότυπου του runtime environment. Παρατηρώντας την

επόμενη εικόνα, το runtime environment αποτελείται από την Platform “roe:1099/JADE”. Αυτή με τη σειρά της περιέχει containers, υποχρεωτικά και τον main container. Ο RMA (Remote Management Agent) βοηθά στο να έχουμε αυτό το γραφικό περιβάλλον. Πέρα από το να έχουμε μια εικόνα, το GUI έχει απεριόριστες δυνατότητες όπως το να εισάγουμε νέους πράκτορες ή να κάνουμε kill κάποιον που ήδη υπάρχει.



Εικόνα 41. Γραφικό περιβάλλον του JADE.

Στο παράδειγμα αυτό υπάρχει ο buyer b και ο sellers s. Μπορούμε να έχουμε πολλές εκτελέσεις του ίδιου προγράμματος και άρα πολλούς πράκτορες που ανήκουν σε κάθε μια κατηγορία. Το ζητούμενο είναι το όνομα που έχει ο κάθε πράκτορας να είναι μοναδικό. Γι' αυτό φροντίζει ο AMS agent.

Πολλά ζητήματα προκύπτουν κατά την υλοποίηση των δυο πρακτόρων. Ποιες θα είναι οι χρονικές καθυστερήσεις, πως οι διάφοροι sellers θα μπορούν ταυτόχρονα να εξυπηρετήσουν πολλούς buyers, πως θα ξεχωρίσουμε τις διαφορετικές συνομιλίες μεταξύ ενός συγκεκριμένου ζεύγους buyer-seller κλπ.

Το Jade, και συγκεκριμένα η βιβλιοθήκη κλάσεων, παρέχει την abstract class Behaviour. Από αυτή μπορούν να δημιουργηθούν διάφορες κατηγορίες behaviour

- Primitive behaviours

- SimpleBehaviour: είναι απλή στην χρήση. Την κάνουμε extend για να εισάγουμε δικές μας λειτουργίες στους agents
- CyclicBehaviour: εκτελείτε συνεχώς και περιοδικά, κάνοντας μια συγκεκριμένη λειτουργία που εμείς θέλουμε
- OneShotBehaviour: εκτελείτε μια μόνο φορά από τον agent
- Composite Behaviours
 - ParallelBehaviour: περιλαμβάνει πολλές διαφορετικές behaviours (children behaviours) που εκτελούνται ταυτόχρονα.
 - SequentialBehaviour: περιλαμβάνει πολλές διαφορετικές behaviours (children behaviours) που εκτελούνται στη σειρά.

Οι πιο χρήσιμες για το δικό μας σενάριο είναι οι Composite Behaviours, που χρησιμοποιούνται συνδυαστικά.

Στο Jade, κάθε agent αντιστοιχεί σε ένα thread. Όμως, όπως είπαμε και πριν μπορεί να θέλουμε ένας agent να κάνει πολλά πράγματα ταυτόχρονα. Όμως αυτό γίνεται μέσω κλάσεων, και συγκεκριμένα από κλίσεις συναρτήσεων που υλοποιούμε εμείς. Οι συναρτήσεις αυτές ενεργούν κατά κάποιο τρόπο σαν Event Handlers, για παράδειγμα όταν λάβει ένας agent ένα συγκεκριμένο μήνυμα, εκτελεί μια δεδομένη συνάρτηση. Το ερώτημα είναι γιατί να χρησιμοποιούμε συναρτήσεις και όχι και απλά threads που είναι light-weight. Ο λόγος είναι ότι η μετάβαση από την εκτέλεση ενός thread σε ένα άλλο είναι 100 φορές πιο αργή από την κλίση μιας συνάρτησης. Ακόμη το γεγονός ότι μπορούμε να έχουμε παράλληλη εκτέλεση με τη χρήση ParallelBehaviour, μας κάνει να αποφεύγουμε τον προγραμματισμό με threads.