ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ

ΑΝΑΠΤΥΞΗ ΓΡΑΦΙΚΟΥ ΠΕΡΙΒΑΛΛΟΝΤΟΣ ΣΕ MATLAB ΓΙΑ ΣΥΣΤΑΔΟΠΟΙΗΣΗ ΜΕΣΩ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ ISODATA

Μαρκαντωνάτου Μαρία Α.Μ.: 379

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: Δρ. Τσιμπίρης Αλκιβιάδης

Περιεχόμενα

Ευχαριστίες

Η παρούσα πτυχιακή δεν θα μπορούσε να υλοποιηθεί χωρίς την ουσιαστική συμβολή του επιβλέποντα καθηγητή μου κ. Τσιμπίρη Αλκιβιάδη ο οποίος μου έδωσε τη δυνατότητα να ασχοληθώ με το θέμα αυτό και με καθοδήγησε σε όλη τη διάρκεια της εργασίας. Επίσης θα ήταν παράλειψη μου να μην ευχαριστήσω τους γονείς μου Διονύση και Ολυμπία για αμέριστη συμπαράσταση και υπομονή που έδειξαν καθ όλη τη διάρκεια των σπουδών μου.

Εισαγωγή

Με το όρο συσταδοποίηση (clustering) εννοούμε την στατιστική διαδικασία με την οποία προσπαθούμε να οργανώσουμε τα δεδομένα σε ομάδες (clusters), οι οποίες δεν είναι από πριν γνωστές, αλλά προκύπτουν δυναμικά. Ο αλγόριθμος ISODATA προσπαθεί να βρει την καλύτερη ομάδα από τα κέντρα των συστάδων για ένα δεδομένο πλήθος σημείων σε d-διαστάσεις, ακλουθώντας μια επαναληπτική προσέγγιση έως ότου επιτευχθεί ένας μέγιστος αριθμός επαναλήψεων.

Ο σκοπός της πτυχιακής εργασίας είναι:

α) η αναλυτική παρουσίαση του αλγορίθμου ISODATA

β) η σύγκριση της απόδοσής του με άλλες μεθόδους όπως με τον k-means.

γ) η προγραμματιστική υλοποίησή του σε Matlab

δ) η δημιουργία ολοκληρωμένου προγράμματος με γραφικό περιβάλλον (GUI) σε Matlab που θα περιλαμβάνει

i) σύνδεση με βάσεις δεδομένων μέσω ODBC για φόρτωση των δεδομένων.

ii) την εφαρμογή του αλγορίθμου ISODATA για εύρεση συστάδων

iii) Οπτική παρουσίαση των αποτελεσμάτων με γραφήματα

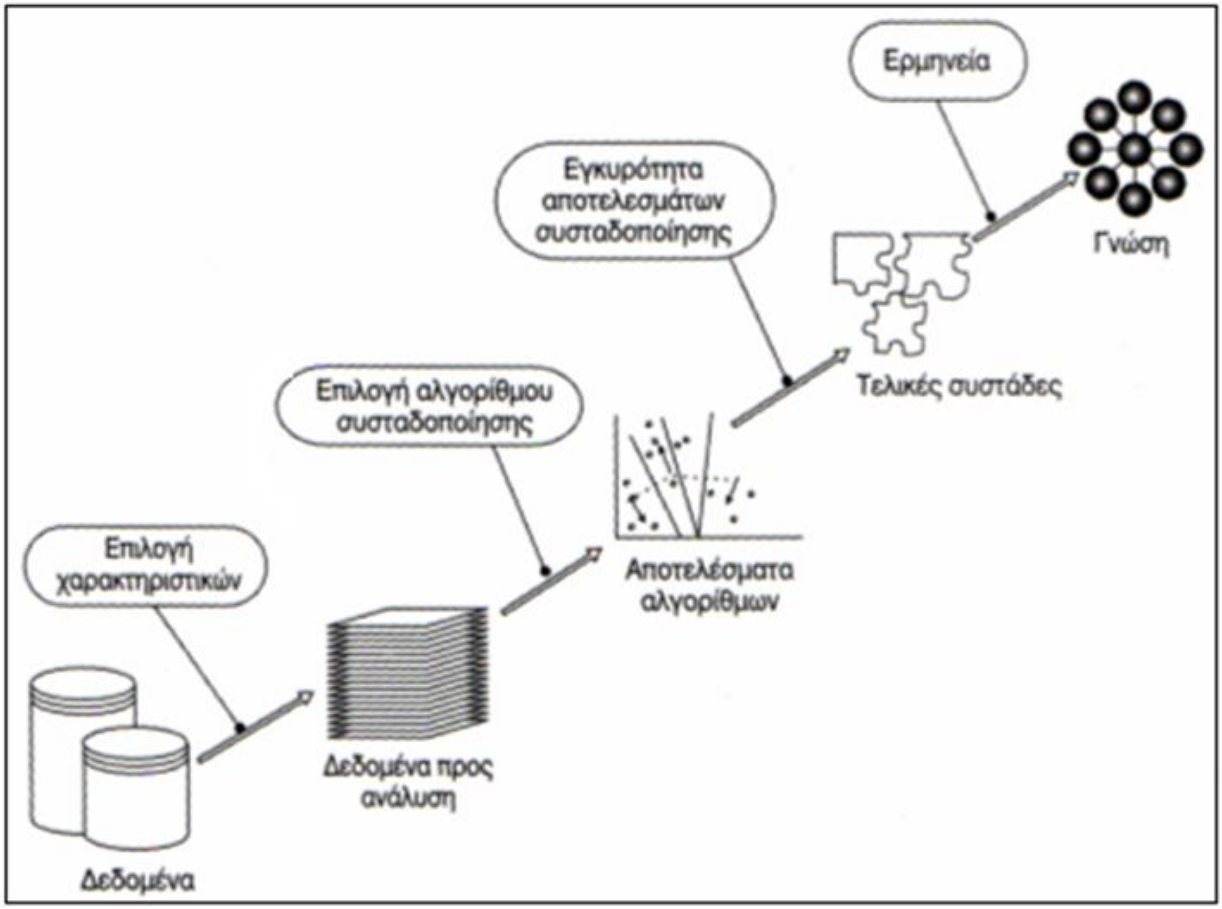
iv) αποθήκευση των δεδομένων σε βάσεις δεδομένων.

Στην συνέχεια στο κεφάλαιο 1 θα παρουσιάσουμε θεωρητικά κάποια στοιχεία για τεχνικές εξόρυξης γνώσης, και θα εξηγήσουμε τον τρόπο λειτουργίας των αλγορίθμων k-means και Isodata. Στο κεφάλαιο 2 θα παρουσιάσουμε βήμα βήμα τον τρόπο υλοποίησης της εφαρμογής μας, στο κεφάλαιο 3 θα χρησιμοποιήσουμε δεδομένα για να κάνουμε την σύγκριση μεταξύ των αλγορίθμων και θα παρουσιάσουμε γραφικά τα αποτελέσματα και στο τέλος θα παραθέσουμε τα συμπεράσματά μας.

Κεφάλαιο 1. Συσταδοποίση (clustering)

1.1 Εξόρυξη Γνώσης

Με τον όρο εξόρυξη γνώσης (είναι επίσης γνωστή και ως Knowledge Discovery in Databases-KDD) εννοούμε τη διαδικασία εύρεσης συσχετισμών ή μοτίβα ανάμεσα σε δεκάδες τομείς σε μεγάλες σχεσιακές βάσεις δεδομένων. Η διαδικασία αυτή χρησιμοποιεί διάφορες τεχνικές έτσι ώστε να παρουσιάσει στον χρήστη τα αποτελέσματα σε μία πιο κατανοητή μορφή.



Εικόνα 1 Εξόρυξη Γνώσης

Τα βήματα της συγκεκριμένης διαδικασίας φαίνονται στην εικόνα 1. Αναλυτικότερα βλέπουμε πως από μια συλλογή δεδομένων επιλέγουμε κάποια χαρακτηριστικά έτσι ώστε τα δεδομένα που είναι προς ανάλυση να επεξεργαστούν από τους κατάλληλους αλγορίθμους. Τα αποτελέσματα των αλγόριθμων αυτών θα δημιουργήσουν τις τελικές συστάδες, οι οποίες στην συνέχεια θα ερμηνευτούν και θα καταλήξουμε στα τελικά μας συμπεράσματα.

Στον πεδίο της εξόρυξης γνώσης συμπεριλαμβάνονται αλγόριθμοι εποπτευόμενης και μη εποπτευόμενης μάθησης. Με τον όρο μη εποπτευόμενη μάθηση ή εκμάθηση χωρίς επόπτη (unsupervised) εννοούμε τη μηχανική μάθηση η οποία επικεντρώνεται σε προβλήματα κατηγοριοποίησης δεδομένων για τα οποία δεν είναι εκ των προτέρων γνωστή η κλάση προέλευσης των διαθέσιμων δεδομένων. Πρόκειται δηλαδή για μια ειδική κατηγορία προβλημάτων ταξινόμησης για τα οποία η μοναδική διαθέσιμη πληροφορία είναι μια συλλογή δειγμάτων χωρίς κάποια επιπλέον ετικέτα η οποία να προσδιορίζει την προέλευσή τους. Αντίθετα με τον όρο εποπτευόμενη μάθηση ή εκμάθηση με επόπτη (supervised) εννοούμε τη μηχανική μάθηση η οποία επικεντρώνεται σε προβλήματα κατηγοριοποίησης δεδομένων για τα οποία είναι εκ των προτέρων γνωστή η κλάση προέλευσης των διαθέσιμων δεδομένων. Στη συνέχεια θα αναλύσουμε κάποιες από τις κατηγορίες εξόρυξης δεομένων στις οποίες μπορούμε να συναντήσουμε τους αλγορίθμους αυτούς. Οι κυριότερες είναι οι συσταδοποίηση (clustering), η κατηγοριοποίηση (classification), τα πρότυπα ακολουθιών (sequential patterns), τα δέντρα απόφασης (decision trees) , οι κανόνες συσχέτισης (association rules) καθώς και η παλινδρόμηση (regression).

- **Συσταδοποίηση (clustering):** Με τον όρο συσταδοποίηση (clustering) αναφερόμαστε στη διαδικασία ταξινόμησης δεδομένων σε ομάδες (clusters) χωρίς να υφίσταται κανενός είδους επίβλεψη (unsupervised). Τα προς συσταδοποίηση δεδομένα μπορεί να είναι παρατηρήσεις ή διανύσματα μεταβλητών. Συνήθως αναπαριστώνται ως διανύσματα μετρήσεων ή ως σημεία σε κάποιον πολυδιάστατο χώρο. Η τεχνική της συσταδοποίησης επιχειρεί να οργανώσει μια συλλογή από δεδομένα σε ομάδες βασιζόμενη στο χαρακτηριστικό της ομοιότητας. Διαισθητικά, τα δεδομένα που ανήκουν σε μία ομάδα θα πρέπει να είναι περισσότερο όμοια μεταξύ τους από εκείνα που ανήκουν σε μία διαφορετική ομάδα [12].

  Στην περίπτωση της κατηγοριοποίησης χωρίς επίβλεψη (clustering), μιας συλλογής «αμαρκάριστων» (μη κατηγοριοποιημένων από πριν) δεδομένων επιχειρείται η δημιουργία λογικών ομάδων από αυτά, τα

οποία και «μαρκάρονται» με ένα είδους ετικέτας της ομάδας τους. Οι ετικέτες αυτές προκύπτουν δυναμικά και αποκλειστικά από τα ίδια τα δεδομένα της συλλογής.

- **Κατηγοριοποίηση (classification):** Αντίθετα, στην κατηγοριοποίηση με επίβλεψη (classification), δοθείσης μιας συλλογής «μαρκαρισμένων» (κατηγοριοποιημένων με κάποιο τρόπο από πριν) δεδομένων επιχειρείται η απόδοση ετικέτας από τις ήδη υπάρχουσες ομάδες δεδομένων σε κάθε νέο «αμαρκάριστο» δεδομένο που προκύπτει για ταξινόμηση. Στην ουσία, η αρχική συλλογή των ήδη ταξινομημένων δεδομένων δίνεται ως οδηγός εκμάθησης (training set) των χαρακτηριστικών κάθε ομάδας προκειμένου να μπορέσει να ταξινομηθεί σε κάποια από αυτές κάθε νέο δεδομένο που προκύπτει. Αυτή η ειδοποιός διαφορά ανάμεσα στις τεχνικές κατηγοριοποίησης με και χωρίς επίβλεψη είναι και ο λόγος που σε περιπτώσεις λήψης αποφάσεων και μηχανικής μάθησης, η τεχνική της συσταδοποίησης κρίνεται ως η πλέον κατάλληλη για ταξινόμηση δεδομένων, καθώς είναι διαθέσιμη ελάχιστη πρότερη πληροφορία για τα δεδομένα αυτά και επομένως η ταξινόμηση θα πρέπει να γίνει με όσο το δυνατό λιγότερες υποθέσεις και παραδοχές. Η κατηγοριοποίηση βασίζεται στην εξέταση των χαρακτηριστικών μιας συλλογής δεδομένων που με βάση αυτά τα χαρακτηριστικά χωρίζεται σε έναν προκαθορισμένο αριθμό κλάσεων. Τα δεδομένα που πρόκειται να κατηγοριοποιηθούν ανατίθενται σε κάποιες από τις προκαθορισμένες κλάσεις. Η βασική εργασία της κατηγοριοποίησης είναι να αναθέσει τα δεδομένα που δεν ανήκουν κάπου σε κάποια από τις ήδη υπάρχουσες ομάδες. Στις περισσότερες περιπτώσεις υπάρχει ένας συγκεκριμένος αριθμός κλάσεων και εμείς καλούμαστε να εντάξουμε τα δεδομένα σε κάποια από αυτές. Για το σκοπό αυτό υπάρχουν κάποιες τεχνικές όπως τα νευρωνικά δίκτυα (neural networks) τα δέντρα απόφασης (decision trees) και άλλα [7].
- **Παλινδρόμηση (regression):** Η παλινδρόμηση συνήθως συναντάται στην στατιστική. Σκοπός της είναι η πρόβλεψη της τιμής μιας μεταβλητής μελετώντας τις τιμές που είχε στο παρελθόν. Η παλινδρόμηση

καλύπτει ένα μεγάλο κομμάτι που έχει να κάνει με την εξόρυξη γνώσης όσο αναφορά στην πρόβλεψη δεδομένων [9].

- **Κανόνες συσχέτισης (association rules):** Οι κανόνες συσχέτισης παρέχουν έναν πιο σύντομο και ενδεχομένως πιο κατανοητό τρόπο για να εκφραστούν οι πληροφορίες που αναζητούν οι τελικοί χρήστες. Οι κανόνες συσχέτισης βρίσκουν τις συσχετίσεις μεταξύ των γνωρισμάτων ενός συνόλου δεδομένων.
- **Πρότυπα ακολουθιών (sequential patterns):** Πρόκειται για την εξόρυξη των πιο συχνά εμφανιζόμενων στοιχείων σε ακολουθίες δεδομένων.

1.2 Συσταδοποίηση Δεδομένων

Στην εργασία αυτή επικεντρώσαμε στην συσταδοποίηση και για τον λόγο αυτό θα αναφερθούμε εκτενέστερα. Η συσταδοποίηση είναι μια διαδικασία ταξινόμησης δεδομένων με βάση κάποιον δείκτη ομοιότητας. Πρόκειται για μια υποκειμενική διαδικασία καθώς το ίδιο σύνολο δεδομένων μπορεί να χρειαστεί να ομαδοποιηθεί διαφορετικά στα πλαίσια διαφορετικών εφαρμογών. Αυτό συνεπάγεται την αδυναμία ενός μόνο αλγορίθμου να καλύψει όλες αυτές τις διαφορετικές εφαρμογές, με αποτέλεσμα να είναι απαραίτητη η χρήση της γνώσης πάνω στο επιστημονικό πεδίο των δεδομένων, η οποία και χρησιμοποιείται άμεσα ή έμμεσα σε μία ή περισσότερες φάσεις του αλγορίθμου συσταδοποίησης. Η συσταδοποίηση όπως και η κατηγοριοποίηση πραγματοποιείται σε μια σειρά από βήματα. Το πιο σημαντικό και συγχρόνως πιο πολύπλοκο βήμα είναι αυτό της εξαγωγής των πιο χρήσιμων μεταβλητών καθώς και της αναπαράστασης των δεδομένων. Από αυτό κρίνεται σε μεγάλο βαθμό η επιτυχία του ίδιου του αλγορίθμου ταξινόμησης.

Επόμενο βήμα είναι η επιλογή του κατάλληλου δείκτη ομοιότητας. Έχουν αναπτυχθεί διάφοροι δείκτες καθορισμού της ομοιότητας ή ακόμη και της ανομοιότητας ανάμεσα στα προς ταξινόμηση στοιχεία και ο κάθε αλγόριθμος χρησιμοποιεί τον πλέον κατάλληλο ανάλογα με τη φύση των δεδομένων αλλά και του ίδιου του προβλήματος ταξινόμησης. Ο πιο

συνηθισμένος δείκτης ανομοιότητας είναι η μέτρηση της απόστασης ανάμεσα στα προς ταξινόμηση στοιχεία. Ακολουθεί το βήμα της ομαδοποίησης (clustering), με ένα πλήθος αλγορίθμων να έχουν αναπτυχθεί στο βήμα αυτό. Πολύ γενικά, υπάρχουν δυο κύριες κατηγορίες αλγορίθμων, οι ιεραρχικοί και οι αλγόριθμοι διαμέρισης.

Οι αλγόριθμοι διαμέρισης επιχειρούν να μεγιστοποιήσουν τη συνάρτηση τετραγωνικού σφάλματος. Λόγω της αδυναμίας τους να πετύχουν κάτι τέτοιο, αναπτύχθηκαν πολλές και διαφορετικές προσεγγίσεις με στόχο την εύρεση της καθολικά καλύτερης λύσης για το πρόβλημα συσταδοποίησης. Σε ορισμένες εφαρμογές, όπως για παράδειγμα στην ανάκτηση κειμένου, μπορεί να είναι χρήσιμο η συσταδοποίηση να μην αποτελεί πλήρη διαμέριση του αρχικού συνόλου, αλλά οι ομάδες να επικαλύπτονται. Σε αυτές τις περιπτώσεις χρησιμοποιούνται οι λεγόμενοι fuzzy (ασαφής) αλγόριθμοι ταξινόμησης, οι οποίοι μπορούν να χειριστούν μεικτού τύπου δεδομένα. Ωστόσο, το πρόβλημα με τους αλγορίθμους αυτούς έγκειται στη δυσκολία εύρεσης των ποσοστών συμμετοχής στις ομάδες. Ο αλγόριθμος k-means και οι χάρτες Kohonen, ισοδύναμό του στα τεχνητά νευρωνικά δίκτυα, είναι οι πιο επιτυχημένοι αλγόριθμοι σε μεγάλα σύνολα δεδομένων. Αυτό συμβαίνει γιατί ο k-means είναι εύκολα υλοποιήσιμος και ιδιαίτερα ελκυστικός από υπολογιστικής άποψης εξαιτίας της σχεδόν γραμμικής χρονικής του πολυπλοκότητας. Ο αλγόριθμος ISODATA επίσης δείχνει ευέλικτος διότι δεν καθορίζεται εκ των προτέρων ο αριθμός των κλάσεων που θα χωριστούν τα δεδομένα.

Συνοψίζοντας, η συσταδοποίηση δεδομένων αποτελεί ένα ενδιαφέρον, ιδιαίτερα χρήσιμο και προκλητικό πρόβλημα. Έχει πολύ καλές προοπτικές σε εφαρμογές όπως η αναγνώριση αντικειμένων, το φιλτράρισμα και τέλος, η ανάκτηση πληροφοριών. Ωστόσο, μπορεί κανείς να εκμεταλλευτεί προς όφελός του τις προοπτικές αυτές μόνο εάν λάβει εκ των προτέρων ιδιαίτερα προσεκτικά πολλές παραμέτρους των προβλημάτων.

1.3 Τεχνικές Συσταδοποίησης

Πολύ γενικά, οι τεχνικές συσταδοποίησης διακρίνονται σε δύο κύριες κατηγορίες, τις ιεραρχικές και τις τεχνικές διαμέρισης. Οι ιεραρχικές παράγουν μια εμφωλευμένη ακολουθία από ομάδες δεδομένων, ενώ οι τεχνικές διαμέρισης παράγουν μόνο ένα σύνολο από ομάδες. Η τελική επιλογή της τεχνικής που θα χρησιμοποιηθεί σε κάθε περίπτωση θα πρέπει να γίνεται με γνώμονα το ίδιο το πρόβλημα ταξινόμησης, το είδος των προς ταξινόμηση δεδομένων, τις μεταβλητές που χαρακτηρίζουν τα δεδομένα, αλλά και τη μετρική ομοιότητας. Οποιαδήποτε τεχνική ταξινόμησης και αν χρησιμοποιηθεί, το παραγόμενο αποτέλεσμα θα είναι ομάδες ομοειδών στοιχείων (clusters), κάθε μία από τις οποίες θα έχει τα ακόλουθα χαρακτηριστικά :

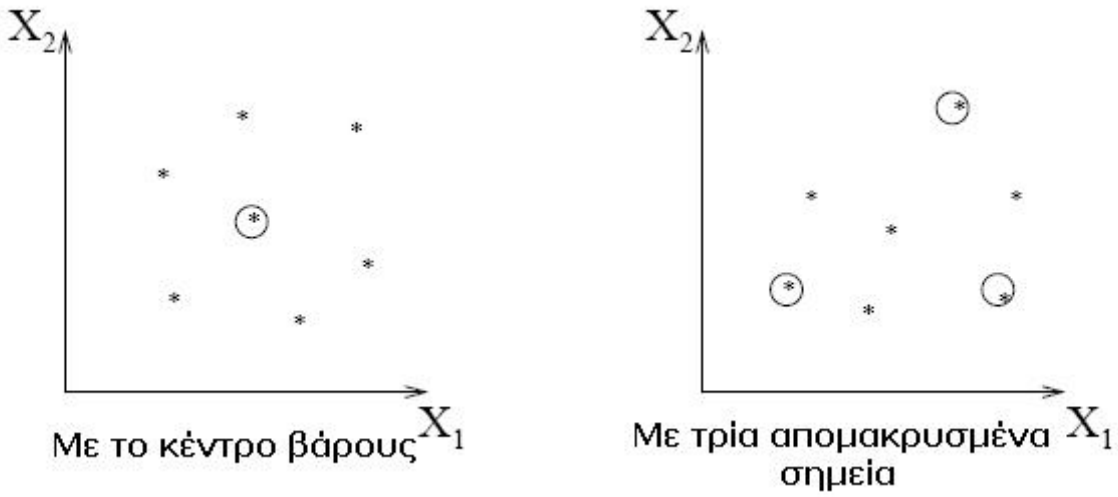
- Πυκνότητα : ορίζεται από το πλήθος των ομοειδών στοιχείων που τοποθετούνται στο χώρο.
- Διακύμανση : αναφέρεται στις αποστάσεις των σημείων μιας ομάδας από το κέντρο της. Όταν τα σημεία τοποθετούνται κοντά στο κέντρο βάρους, η ομάδα χαρακτηρίζεται ως συμπαγής, διαφορετικά θεωρείται χαλαρή.
- Διάσταση : εκφράζει την ακτίνα της σχηματιζόμενης έλλειψης, δηλαδή του σχήματος της ομάδας στο χώρο.
- Διαχωρισμό : αναφέρεται στη δυνατότητα οι ομάδες να αλληλοκαλύπτονται ή όχι.

1.3.1 Αναπαράσταση Ομάδων

Υπάρχουν τρείς δυνατοί τρόποι αναπαράστασης των ομάδων που προκύπτουν από κάποιον αλγόριθμο ταξινόμησης:

- Αναπαράσταση με χρήση των κέντρων βάρους κάθε ομάδας ή ενός συνόλου απομακρυσμένων στοιχείων της ομάδας.

- Αναπαράσταση με χρήση κόμβων σε ένα δένδρο ταξινόμησης.
- Αναπαράσταση με χρήση συνδετικών λογικών εκφράσεων.



Η αναπαράσταση των ομάδων με τα κέντρα βάρους τους είναι και η πλέον διαδεδομένη, με καλά αποτελέσματα για περιπτώσεις όπου οι ομάδες είναι συμπαγείς ή ισοτροπικές. Ωστόσο, για περιπτώσεις που οι ομάδες είναι επιμήκεις ή μη-ισοτροπικές, τα κέντρα βάρους αδυνατούν να τις αναπαραστήσουν σωστά, οπότε η χρήση ενός συνόλου απομακρυσμένων στοιχείων, που στην ουσία οριοθετούν τις ομάδες, αποδίδει το σχήμα τους αποτελεσματικά. Ο αριθμός των σημείων που χρησιμοποιούνται για την αναπαράσταση μιας ομάδας θα πρέπει να αυξάνει καθώς αυξάνει η πολυπλοκότητα του σχήματός της.

Η αφαίρεση δεδομένων είναι χρήσιμη σε περιπτώσεις λήψης αποφάσεων, καθώς :

- δίνει μια απλή και διαισθητική περιγραφή των ομάδων που γίνεται εύκολα κατανοητή από τους ανθρώπους-χρήστες.
- βοηθά στη συμπίεση των δεδομένων κατά τέτοιο τρόπο ώστε να είναι εύκολη η περαιτέρω επεξεργασία τους από τον υπολογιστή.
- αυξάνει την αποτελεσματικότητα της ίδιας της διαδικασίας της λήψης αποφάσεων.

Για παράδειγμα, στην περίπτωση συσταδοποίησης μιας μεγάλης συλλογής κειμένων, αν η αναπαράσταση των ομάδων γίνει με χρήση των κέντρων βάρους τους, όταν επιχειρείται ανάκτηση ενός κειμένου με κάποιο ερώτημα

προς τη συλλογή το μόνο που χρειάζεται είναι ο έλεγχος ομοιότητας του ερωτήματος με το κέντρο βάρους κάθε ομάδας και όχι με κάθε στοιχείο της ομάδας χωριστά.

1.3.4 Τα στάδια της ομαδοποίησης

Η τεχνική της ομαδοποίησης (που χρησιμοποιείται στην συσταδοποίηση αλλά και στην κατηγοριοποίηση των δεδομένων) περιλαμβάνει τα ακόλουθα βήματα:

1. Αναπαράσταση των προς ταξινόμηση δεδομένων, που περιλαμβάνει προαιρετικά τη διαδικασία εξαγωγής μεταβλητών (feature extraction) και/ή της επιλογής μεταβλητών (feature selection),
2. καθορισμός ενός μέτρου σύγκρισης της ομοιότητας των δεδομένων, κατάλληλου προς το πεδίο που δημιουργείται από αυτά,
3. συσταδοποίηση ή κατηγοριοποίηση των δεδομένων,
4. αφαίρεση δεδομένων *(data abstraction)*, αν χρειαστεί, και
5. αξιολόγηση των αποτελεσμάτων.

1. Η αναπαράσταση των δεδομένων αφορά τον αριθμό των ομάδων, τον αριθμό των διαθέσιμων δεδομένων και τον αριθμό, τον τύπο και τη διακύμανση των μεταβλητών που είναι διαθέσιμες στον αλγόριθμο ταξινόμησης. Είναι δυνατό τμήμα των πληροφοριών αυτών να μη μπορούν να ελεγχθούν από το χρήστη. Η διαδικασία της επιλογής μεταβλητών αναφέρεται στην αναγνώριση του πιο αποτελεσματικού (feature section) υποσυνόλου μεταβλητών από τις αρχικές μεταβλητές της συλλογής των δεδομένων ώστε να χρησιμοποιηθεί από τον αλγόριθμο ταξινόμησης. Η διαδικασία εξαγωγής μεταβλητών (feature extraction) αφορά τη χρήση ενός ή περισσότερων μετασχηματισμών των μεταβλητών εισόδου προκειμένου να παραχθούν νέες μεταβλητές. Κάθε μία ξεχωριστά ή και οι δύο μαζί, οι διαδικασίες εξαγωγής και επιλογής μεταβλητών μπορούν να χρησιμοποιηθούν αποδοτικά

προκειμένου να παραχθεί ένα σύνολο μεταβλητών για τα δεδομένα, κατάλληλο προς χρήση κατά τη διάρκεια της συσταδοποίησης.

2. Συνήθως, ως μέτρο σύγκρισης της ομοιότητας δύο δεδομένων χρησιμοποιείται μια συνάρτηση υπολογισμού της απόστασης των δεδομένων αυτών. Η πιο απλή συνάρτηση αυτού του είδους είναι η Ευκλείδεια απόσταση, η οποία χρησιμοποιείται για τον καθορισμό της ανομοιότητας ανάμεσα σε δύο δεδομένα. Αυτή τη συνάρτηση χρησιμοποιήσαμε και στην εργασία μας.
3. Το βήμα της συσταδοποίησης των δεδομένων μπορεί να πραγματοποιηθεί με μια πληθώρα μεθόδων και αλγορίθμων :
    a. Το παραγόμενο αποτέλεσμα μπορεί να περιλαμβάνει μόνο ομάδες που αποτελούν μικρότερα τμήματα της αρχικής συλλογής και κάθε στοιχείο να ανήκει αποκλειστικά σε μία από τις ομάδες αυτές (hard αλγόριθμοι) ή ομάδες στις οποίες κάθε στοιχείο της αρχικής συλλογής συμμετέχει με ένα μεταβλητό ποσοστό συμμετοχής (fuzzy αλγόριθμοι).
    b. Οι λεγόμενοι ιεραρχικοί αλγόριθμοι παράγουν μια εμφωλευμένη σειρά από τμήματα της αρχικής συλλογής, βασισμένοι σε ένα κριτήριο σύγκρισης ομοιότητας με το οποίο συνενώνουν ή διαχωρίζουν τις ομάδες των δεδομένων.
    c. Οι αλγόριθμοι διαμέρισης (partitional algorithms) αναγνωρίζουν τα τμήματα εκείνα της αρχικής συλλογής που βελτιστοποιούν, συνήθως τοπικά, ένα κριτήριο ταξινόμησης.
    d. Υπάρχουν πιθανοθεωρητικές και γραφοθεωρητικές μέθοδοι ταξινόμησης.
4. Η αφαίρεση των δεδομένων είναι η διαδικασία εξαγωγής μιας απλής και συμπαγούς αναπαράστασης του συνόλου των δεδομένων. Η έννοια της απλότητας έχει να κάνει κυρίως με το κατά πόσο είναι δυνατό το παραγόμενο σύνολο δεδομένων είτε να υποστεί επεξεργασία άμεσα και ικανοποιητικά από μια μηχανή είτε να γίνει άμεσα και απλά κατανοητό από τον άνθρωπο-χρήστη. Μια τυπική αφαίρεση δεδομένων στον κόσμο της ομαδοποίησης αναφέρεται σε μια συμπαγή περιγραφή κάθε ομάδας, συνήθως με όρους πρωτοτύπων

ή αντιπροσωπευτικών για κάθε ομάδα στοιχείων, όπως είναι για παράδειγμα τα κέντρα βάρους (centroids) κάθε ομάδας.

5. Στις περιπτώσεις όπου χρησιμοποιούνται στατιστικές μέθοδοι ταξινόμησης, η αξιολόγηση της εγκυρότητας των αποτελεσμάτων γίνεται με προσεκτική εφαρμογή στατιστικών μεθόδων και υποθέσεων που χρησιμοποιούνται για έλεγχο. Υπάρχουν τρεις τρόποι αξιολόγησης της εγκυρότητας των αποτελεσμάτων ταξινόμησης :

    a. η εξωτερική αξιολόγηση της εγκυρότητας, η οποία συγκρίνει το παραγόμενο αποτέλεσμα με ένα *a priori* αποτέλεσμα.
    b. η εσωτερική αξιολόγηση της εγκυρότητας, η οποία προσπαθεί να καθορίσει αν το αποτέλεσμα είναι φυσικά κατάλληλο για τα δεδομένα προς ταξινόμηση.
    c. ο έλεγχος σχετικότητας, ο οποίος συγκρίνει δύο δομές και υπολογίζει το μερίδιο σχετικότητάς τους.

1.3.5 Χρήσιμοι Ορισμοί

- Ένα στοιχείο ή διάνυσμα μεταβλητών ή παρατήρηση x είναι ένα δεδομένο που χρησιμοποιείται από τον αλγόριθμο ταξινόμησης. Αποτελείται από ένα διάνυσμα *d* μετρήσεων : $\boldsymbol{x} = (x1, \dots, xd)$.
- Τα ξεχωριστά, μοναδιαία συστατικά *xi* του δεδομένου *x* καλούνται μεταβλητές ή ιδιότητες.
- Το *d* εκφράζει τη διάσταση του δεδομένου ή του χώρου που δημιουργείται από τα δεδομένα προς ταξινόμηση.
- Ένα σύνολο δεδομένων συμβολίζεται $\boldsymbol{X} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\}$. Το *i*-στο στοιχείο του $\boldsymbol{X}$ αναφέρεται ως $\boldsymbol{x^i} = (\boldsymbol{x^i_1}, \dots, \boldsymbol{x^i_d})$. Σε πολλές περιπτώσεις το σύνολο των δεδομένων που πρόκειται να ταξινομηθούν αντιμετωπίζεται ως ένας $\boldsymbol{n \times d}$ πίνακας δεδομένων.

- Οι *hard* τεχνικές ταξινόμησης αποδίδουν μία ετικέτα ομάδας l_i σε κάθε δεδομένο x^i που ανήκει στην ομάδα αυτή. Το σύνολο όλων των ετικετών για το σύνολο δεδομένων X είναι το $L = \{l_1, \ldots, l_n\}$ όπου τα l_i ανήκουν στο σύνολο {1, …, *k*}, όπου *k* είναι ο αριθμός των ομάδων.
- Οι fuzzy τεχνικές ταξινόμησης αποδίδουν σε κάθε δεδομένο εισόδου xi ένα ποσοστό συμμετοχής του, f_{ij}, σε κάθε παραγόμενη ομάδα ταξινόμησης (cluster) j.
- Μια μετρική απόστασης (ειδική περίπτωση μετρικής ομοιότητας) είναι μια μετρική που εφαρμόζεται στο χώρο που δημιουργείται από τα δεδομένα εισόδου με στόχο να ποσοτικοποιήσει την ομοιότητα μεταξύ τους και πληροί τους παρακάτω περιορισμούς:
  Έστω δύο σημεία χ και y τότε **D**(x,y)**>=**0 και **D**(x,y)**=** **D**(y,x) και **D**(x,y)**<D**(x,A)+**D**(A,y) όπου A ένα τρίτο σημείο

1.3.6 Μετρικές Ομοιότητας

Όπως έχει αναφερθεί, η ομοιότητα ανάμεσα στα δεδομένα του προς ταξινόμηση συνόλου αποτελεί το κλειδί για τη δημιουργία των ομάδων (clusters), επομένως η δυνατότητα μέτρησής της είναι πρωταρχικής σημασίας σε κάθε διαδικασία ταξινόμησης. Η μεγάλη ποικιλία στους τύπους και τις διακυμάνσεις των μεταβλητών του αρχικού συνόλου δεδομένων επιβάλλει την όσο το δυνατό προσεκτικότερη επιλογή της μετρικής ομοιότητας. Συνήθως, αυτό που μετράται είναι η ανομοιότητα ανάμεσα σε δύο στοιχεία του συνόλου δεδομένων, γεγονός που επιτυγχάνεται με τη μέτρηση της απόστασής τους όπως αυτή ορίζεται στο χώρο που δημιουργούν οι μεταβλητές του συνόλου. Η απόσταση ανάμεσα σε δύο σχετικά στοιχεία είναι μηδενική. Οι μετρικές ομοιότητας που παρουσιάζονται στη συνέχεια, χρησιμοποιούνται για στοιχεία των οποίων οι μεταβλητές έχουν όλες συνεχείς τιμές. Η πιο συνηθισμένη μετρική που χρησιμοποιείται για αυτού του είδους τις μεταβλητές είναι η Ευκλείδεια απόσταση [7] :

$$d_2\left(x_i, x_j\right) = \left(\sum_{k=1}^{d}\left(x_{ik} - x_{jk}\right)^2\right)^{\frac{1}{2}} = \left\|x_i - x_j\right\|_2 \qquad \textbf{Εξίσωση 1}$$

η οποία και αποτελεί ειδική περίπτωση της Minkowski μετρικής για p=2:

$$d_p\left(x_i, x_j\right) = \left(\sum_{k=1}^{d}\left(x_{ik} - x_{jk}\right)^p\right)^{\frac{1}{p}} = \left\|x_i - x_j\right\|_p$$ **Εξίσωση 2**

Η Ευκλείδεια μετρική χρησιμοποιείται κυρίως για τον προσδιορισμό της ανομοιότητας δύο οντοτήτων σε δισδιάστατο ή τρισδιάστατο χώρο, ενώ παράγει ικανοποιητικά αποτελέσματα σε περιπτώσεις όπου το σύνολο δεδομένων έχει συμπαγείς ή απομονωμένες ομάδες. Το μειονέκτημα της Ευκλείδειας απόστασης και κατ' επέκταση της Minkowski μετρικής είναι ότι κατά την εφαρμογή τους οι ευρείας κλίμακας μεταβλητές τείνουν να κυριαρχούν έναντι των άλλων. Οι λύσεις που έχουν δοθεί στο πρόβλημα αυτό έχουν να κάνουν κυρίως με την κανονικοποίηση των μεταβλητών συνεχών τιμών χρησιμοποιώντας τη μέση τιμή ή την απόκλιση κάθε μεταβλητής.

Μια άλλη μετρική που χρησιμοποιείται για τον προσδιορισμό της ομοιότητας των δεδομένων είναι και η μετρική Mahalanobis [7] :

$$d_M\left(x_i, x_j\right) = \left(x_i - x_j\right)\Sigma^{-1}\left(x_i - x_j\right)^T$$ **Εξίσωση 3**

όπου τα x_i και x_j είναι διανύσματα-γραμμές και :

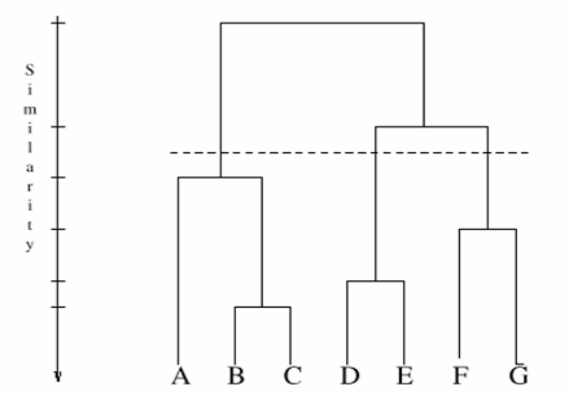
$$\Sigma = \frac{1}{n}\sum_{i=1}^{n}\left(x_i - \bar{x}\right)\left(x_i - \bar{x}\right) \text{ με } \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$ **Εξίσωση 4**

Μερικοί αλγόριθμοι ταξινόμησης δουλεύουν πάνω σε έναν πίνακα με προσεγγιστικές τιμές αντί για το αρχικό σύνολο δεδομένων. Σε αυτές τις περιπτώσεις είναι χρησιμότερο να υπολογίζονται από πριν όλες οι n(n-1)/2 τιμές για τις αποστάσεις των ζευγών των n δεδομένων και να αποθηκεύονται σε έναν συμμετρικό πίνακα. Ο υπολογισμός των αποστάσεων ανάμεσα σε δεδομένα των οποίων κάποιες ή όλες οι μεταβλητές δεν έχουν συνεχείς τιμές είναι προβληματικός, καθώς δεν είναι δυνατό να συγκριθούν οι διαφορετικού τύπου μεταβλητές. Εντούτοις, έχουν προταθεί διάφορες μετρικές ομοιότητας για ετερογενή δεδομένα που συνδυάζουν μετρικές για δεδομένα που έχουν αναπαρασταθεί με ποσοτικές μεταβλητές και μετρικές για δεδομένα που έχουν αναπαρασταθεί με ποιοτικές μεταβλητές

1.4 Είδη αλγορίθμων συσταδοποίησης

1.4.1 Ιεραρχικοί Αλγόριθμοι [7]

Οι ιεραρχικοί αλγόριθμοι είναι οι πιο συχνά χρησιμοποιούμενοι για την ταξινόμηση δεδομένων. Ένας ιεραρχικός αλγόριθμος παράγει ένα δενδρόγραμμα που αναπαριστά την εμφωλευμένη ομαδοποίηση των δεδομένων, καθώς και επίπεδα ομοιότητας στα οποία αλλάζει η ομαδοποίηση.



Εικόνα 2 Δενδροδιάγραμμα Ιεραρχικού αλγόριθμου

Το δενδρόγραμμα (Εικόνα 2) μπορεί να διασπαστεί σε διαφορετικά επίπεδα ώστε να παραχθούν διαφορετικές ομαδοποιήσεις των δεδομένων. Οι ιεραρχικοί αλγόριθμοι στηρίζονται στην επαναληπτική συνένωση ή διάσπαση επιμέρους ομάδων δεδομένων, η οποία γίνεται με βάση κάποια κριτήρια ομοιότητας ή διαφοράς ανάμεσα στις επιμέρους ομάδες δεδομένων που ονομάζονται κριτήρια σύνδεσης. Οι περισσότεροι ιεραρχικοί αλγόριθμοι ταξινόμησης είναι παραλλαγές των αλγορίθμων που χρησιμοποιούν ως κριτήρια σύνδεσης τα :

- single-link (μονή σύνδεση) : η απόσταση ανάμεσα σε δύο ομάδες είναι η ελάχιστη από τις αποστάσεις ανάμεσα σε όλα τα ζεύγη δεδομένων των δύο ομάδων.
- complete-link (πλήρης σύνδεση) : η απόσταση ανάμεσα σε δύο ομάδες είναι η μέγιστη από τις αποστάσεις ανάμεσα σε όλα τα ζεύγη δεδομένων των δύο ομάδων.

- average-link (μέση σύνδεση) : η απόσταση ανάμεσα σε δύο ομάδες είναι η μέση απόσταση από αυτές που προκύπτουν για όλα τα ζεύγη δεδομένων των δύο ομάδων.

Από αυτούς, οι πιο διαδεδομένοι είναι οι single-link και complete-link. Και στις δύο περιπτώσεις, δύο ομάδες δεδομένων συνενώνονται για να δημιουργήσουν μια μεγαλύτερη ομάδα με βάση τα κριτήρια ελάχιστης απόστασης. Ο complete-link αλγόριθμος παράγει στενά δεμένες ή συμπαγείς ομάδες δεδομένων, ενώ ο singlelink αλγόριθμος έχει την τάση να παράγει ομάδες που είναι ακανόνιστες ή επιμήκεις. Ο single-link αλγόριθμος είναι πιο ευπροσάρμοστος από τον complete-link, ωστόσο έχει παρατηρηθεί ότι σε πολλές εφαρμογές ο complete-link αλγόριθμος παράγει πιο χρήσιμες ιεραρχίες από τον single-link.

Οι ιεραρχικοί αλγόριθμοι ταξινόμησης διακρίνονται σε συσσωρευτικούς και σε διαιρετικούς, με τους συσσωρευτικούς να ξεκινούν θεωρώντας κάθε στοιχείο ως μια μοναδιαία ομάδα και με βάση τα κριτήρια σύνδεσης να προχωρούν στη συνένωση των ομάδων για τη δημιουργία μεγαλυτέρων μέχρι όλα τα στοιχεία να ανήκουν σε μία μεγάλη ομάδα, ενώ οι διαιρετικοί ξεκινούν θεωρώντας ότι όλα τα δεδομένα ανήκουν σε μία μεγάλη ομάδα και προχωρούν σε διαρκεί διάσπαση ομάδων κάνοντας χρήση των κατάλληλων κριτηρίων, μέχρι να παραχθούν μοναδιαίες ομάδες δεδομένων.

Συγκριτικά με τους αλγορίθμους διαμέρισης που θα αναφερθούν αναλυτικά στη συνέχεια, οι ιεραρχικοί αλγόριθμοι είναι περισσότερο ευπροσάρμοστοι. Για παράδειγμα, ο single-link αλγόριθμος λειτουργεί καλά σε σύνολα δεδομένων που περιέχουν μη ισοτροπικές (non-isotropic) ομάδες, συμπεριλαμβανομένων καλά διαχωρισμένων, σε στιλ αλυσίδας και ομόκεντρων ομάδων, ενώ ένας τυπικός αλγόριθμος διαμέρισης, όπως ο k-means για παράδειγμα, λειτουργεί καλά μόνο για σύνολα δεδομένων με ισοτροπικές ομάδες. Από την άλλη, οι πολυπλοκότητες χρόνου και χώρου των αλγορίθμων διαμέρισης είναι πολύ πιο χαμηλές σε σχέση με τους ιεραρχικούς. Παρόλα αυτά, είναι εφικτή η ανάπτυξη υβριδικών αλγορίθμων που θα εκμεταλλεύονται τα καλά στοιχεία και των δύο κατηγοριών.

1.4.2 Αλγόριθμοι Διαμέρισης [7]

Ένας αλγόριθμος διαμέρισης παράγει μια μοναδική διαμέριση του συνόλου των δεδομένων αντί για μια δομή ομαδοποίησης, όπως το δενδρόγραμμα μιας ιεραρχικής τεχνικής. Σε περιπτώσεις μεγάλου συνόλου δεδομένων, η χρήση ενός αλγόριθμου διαμέρισης είναι προτιμότερη, καθώς η δημιουργία του δενδρογράμματος του ιεραρχικού αλγορίθμου για ένα τόσο μεγάλο σύνολο θα ήταν, από υπολογιστικής άποψης, απαγορευτική.

Το πρόβλημα με τους αλγορίθμους διαμέρισης έγκειται στην επιλογή του επιθυμητού αριθμού των ομάδων που θα εξαχθούν από τον αλγόριθμο. Οι τεχνικές διαμέρισης παράγουν συνήθως ομάδες δεδομένων βελτιστοποιώντας κάποια συνάρτηση-κριτήριο η οποία ορίζεται είτε τοπικά, σε ένα υποσύνολο των δεδομένων προς ταξινόμηση, είτε καθολικά, σε όλο το σύνολο των δεδομένων. Στην πράξη, ο αλγόριθμος εκτελείται πολλές φορές, με διαφορετικές αρχικές καταστάσεις και η καλύτερη διαρρύθμιση που προκύπτει από όλες τις επαναλήψεις χρησιμοποιείται ως η τελική ομαδοποίηση – έξοδος του αλγορίθμου. Η πιο διαισθητική και συχνά χρησιμοποιούμενη συνάρτηση-κριτήριο στους αλγορίθμους διαμέρισης είναι το τετραγωνικό σφάλμα (squared-error criterion), το οποίο συνήθως αποδίδει σωστά σε περιπτώσεις απομονωμένων και συμπαγών ομάδων δεδομένων. Το τετραγωνικό σφάλμα για μια ομαδοποίηση L ενός συνόλου δεδομένων X με K ομάδες, δίνεται από τη σχέση :

$$e^2(X,L) = \sum_{j=1}^{K}\sum_{i=1}^{n_j}\left\|x_i^{(j)} - c_j\right\|^2 \quad \textbf{Εξίσωση 5}$$

όπου $x_i^{(j)}$ είναι το i-στο στοιχείο που ανήκει στην j-στη ομάδα και c_j είναι το κέντρο βάρους της j-στης ομάδας.

Ο πιο γνωστός αλγόριθμος διαμέρισης που χρησιμοποιεί τη συνάρτηση τετραγωνικού σφάλματος είναι ο k-means. Ξεκινά με μια τυχαία αρχική διαμέριση του συνόλου δεδομένων και επανατοποθετεί τα στοιχεία σε ομάδες με βάση την ομοιότητα ανάμεσα στα στοιχεία και τα κέντρα βάρους των ομάδων, μέχρι να ικανοποιηθεί κάποιο κριτήριο σύγκλισης (όπως για παράδειγμα, έπειτα από έναν αριθμό επαναλήψεων το τετραγωνικό σφάλμα να τείνει να μειώνεται σημαντικά). Ο αλγόριθμος k-means είναι ιδιαίτερα δημοφιλής χάρη στην εύκολη υλοποίησή του, αλλά και τη χρονική του

πολυπλοκότητα που είναι O(n), όπου n είναι ο αριθμός των προς ταξινόμηση στοιχείων. Ένα σημαντικό πρόβλημα του συγκεκριμένου αλγορίθμου είναι η ευαισθησία που επιδεικνύει στην επιλογή των αρχικών διαμερίσεων, με αποτέλεσμα ο αλγόριθμος να συγκλίνει τοπικά σε ένα ελάχιστο της συνάρτησης-κριτηρίου σε περιπτώσεις κακής επιλογής αρχικών διαμερίσεων και έτσι να μην δημιουργούνται τελικά οι καλύτερες δυνατές ομάδες δεδομένων.

Ο k-means αλγόριθμος αποτελείται συνοπτικά από τα ακόλουθα βήματα :

1. Από το αρχικό σύνολο δεδομένων, διάλεξε k τυχαία σημεία τα οποία και ονόμασε κέντρα βάρους των ομάδων.
2. Τοποθέτησε κάθε στοιχείο του συνόλου δεδομένων στην κοντινότερή του ομάδα με βάση την απόστασή του (μικρότερη) από το κέντρο βάρους της ομάδας.
3. Υπολόγισε ξανά τα κέντρα βάρους των ομάδων χρησιμοποιώντας τα στοιχεία που έχουν τοποθετηθεί στο προηγούμενο βήμα σε αυτές.
4. Αν δεν υπάρχει σύγκλιση με κάποιο κριτήριο τερματισμού, πήγαινε στο βήμα 2. Τυπικά κριτήρια σύγκλισης είναι : μηδενική ή μικρή επανατοποθέτηση στοιχείων στα νέα κέντρα βάρους των ομάδων ή σημαντική μείωση του τετραγωνικού σφάλματος.

Υπάρχουν πολλές παραλλαγές του αλγόριθμου k-means, μερικές από τις οποίες επιχειρούν να πετύχουν καλύτερη αρχική διαμέριση του συνόλου δεδομένων ώστε να βρουν ένα καθολικό – και όχι τοπικό - ελάχιστο της συνάρτησης τετραγωνικού σφάλματος. Μια άλλη παραλλαγή αφορά στη βελτιστοποίηση της αντιστοίχησης των δεδομένων στις ομάδες, επιτρέποντας τη διαίρεση και συνένωση των ομάδων που προκύπτουν στα βήματα εκτέλεσης του αλγορίθμου. Τυπικά, μια ομάδα διαιρείται όταν η διακύμανσή της είναι επάνω από ένα προκαθορισμένο κατώφλι, ενώ δύο ομάδες συνενώνονται όταν η απόσταση ανάμεσα στα κέντρα βάρους τους βρίσκεται κάτω από ένα άλλο προκαθορισμένο κατώφλι. Χρησιμοποιώντας αυτή την παραλλαγή, είναι δυνατό να παραχθούν βέλτιστες διαμερίσεις του συνόλου δεδομένων, ξεκινώντας από οποιαδήποτε τυχαία αρχική διαμέριση, αρκεί να έχουν καθοριστεί σωστά τα κατάλληλα κατώφλια. Ο αλγόριθμος ISODATA, η υλοποίηση του οποίου βρίσκεται στο επίκεντρο της παρούσας πτυχιακής εργασίας, αποτελεί μια παραλλαγή του αλγορίθμου k means.

1.5 Ο Αλγόριθμος k-means [7]

1.5.1 Συνοπτική περιγραφή του αλγορίθμου K – Means

Ο αλγόριθμος K – Means διατυπώθηκε από τους MacQueen, Anderbery, Forgy και άλλους τη δεκαετία του 1960 και αποτελεί ένα από τα χαρακτηριστικότερα παραδείγματα αλγορίθμων μη ιεραρχικής συσταδοποίησης. Ο συγκεκριμένος αλγόριθμος βασίζεται στα κέντρα βάρους των αρχικών συστάδων στις οποίες διαμερίζεται το δοσμένο σύνολο των δεδομένων. Η βασική φιλοσοφία του αλγορίθμου K – Means είναι ότι αναμένει από το χρήστη να προσδιορίσει το πλήθος των παραγόμενων συστάδων στις οποίες θα διαμεριστούν τα στιγμιότυπα των δεδομένων που έχουμε στη διάθεσή μας. Τα βήματα εκτέλεσης που ακολουθεί ο συγκεκριμένος αλγόριθμος περιγράφονται επιγραμματικά παρακάτω:

1. Ο αλγόριθμος τροφοδοτείται με το πλήθος των συστάδων και τα αντίστοιχα κέντρα βάρους τους. Τα κέντρα βάρους των προς συσταδοποίηση δεδομένων μπορούν να προσδιοριστούν με κάποιον από τους παρακάτω τρόπους:
    a. Να χρησιμοποιηθούν ως κέντρα βάρους των ομάδων το αποτέλεσμα που έχουν παραχθεί με τη χρήση κάποιου άλλου αλγορίθμου συσταδοποίησης.
    b. Να χρησιμοποιηθεί πρότερη γνώση η οποία όμως δεν έχει να κάνει με τη χρήση κάποιου αλγορίθμου συσταδοποίησης.
    c. Να χρησιμοποιηθούν τυχαία σημεία ως κέντρα των συστάδων κατά την εκκίνηση του αλγορίθμου τα οποία όμως θα ενημερώνονται κατά την εκτέλεση του αλγορίθμου μέχρι να καταλήξουν σε κάποια τελική τιμή που θα αποτελέσει και το τελικό αποτέλεσμα.
2. Για κάθε ένα από τα διαθέσιμα προς συσταδοποίηση δεδομένα, τα οποία από μαθηματική άποψη μπορούμε να τα θεωρήσουμε ως σημεία σε κάποιο πολυδιάστατο διανυσματικό χώρο, υπολογίζεται η απόστασή του από καθένα από τα K κέντρα βάρους (τα οποία αποτελούν επίσης σημεία του ίδιου πολυδιάστατου διανυσματικού χώρου). Η εν λόγω απόσταση υπολογίζεται με βάση την μετρική η οποία ορίζει τον

θεωρούμενο διανυσματικό χώρο. Το κάθε σημείο ανατίθεται τελικά σε εκείνα την συστάδα προς την οποία έχει την ελάχιστη απόσταση από το αντίστοιχο κέντρο βάρους.

3. Ο αλγόριθμος επαναϋπολογίζει τα κέντρα βάρους των συστάδων λαμβάνοντας υπόψη την μεταβολή των συστάδων που πραγματοποιήθηκε κατά το βήμα 2.
4. Ο αλγόριθμος τερματίζει στην περίπτωση κατά την οποία των πλήθος των δεδομένων που άλλαξαν συστάδα κατά το βήμα 2 είναι μικρότερος από κάποιο προκαθορισμένο κατώφλι. Διαφορετικά ο αλγόριθμος επιστρέφει στο βήμα 2.

Η αδυναμία του αλγόριθμου k-means εντοπίζεται στην επιλογή των αρχικών ομάδων, με συνέπεια τα αποτελέσματα που παράγει να είναι σε αρκετές περιπτώσεις χειρότερα από αυτά που προκύπτουν για παράδειγμα από κάποιο γενετικό αλγόριθμο, όταν, όμως, το σύνολο των δεδομένων προς ταξινόμηση δεν είναι μεγάλο. Το γεγονός ότι ο k-means είναι ιδιαίτερα δημοφιλής οφείλεται κυρίως :

- Στη χρονική του πολυπλοκότητα, που είναι O(nkl), όπου n είναι ο αριθμός των στοιχείων, k ο αριθμός των ομάδων και l ο αριθμός των επαναλήψεων μέχρι να συγκλίνει ο αλγόριθμος.
- Στη «χωρική» του πολυπλοκότητα, που είναι O(k+n). Απαιτείται επιπλέον χώρος για να αποθηκευτεί ο πίνακας δεδομένων, ο οποίος όμως είναι δυνατό να αποθηκευτεί σε δευτερεύουσα μνήμη και να γίνεται προσπέλαση κάθε στοιχείου όταν χρειάζεται. Ωστόσο, το τελευταίο σχήμα απαιτεί τεράστιο χρόνο προσπέλασης εξαιτίας της επαναληπτικής φύσης του αλγορίθμου, με επακόλουθο την τεράστια αύξηση του χρόνου εκτέλεσης.
- Δεν εξαρτάται από τη σειρά των δεδομένων. Ανεξάρτητα από το με ποια σειρά θα παρουσιαστούν τα δεδομένα, δοθέντος ενός αρχικού συνόλου κέντρων βάρους των ομάδων, ο αλγόριθμος θα δημιουργήσει την ίδια διαμέριση δεδομένων.

1.5.2. Μαθηματική περιγραφή του αλγορίθμου *K*–means

Ο αλγόριθμος K–means ανήκει σε μια οικογένεια αλγορίθμων συσταδοποίησης όπου η κεντρική φιλοσοφία τους συνίσταται στην ελαχιστοποίηση κάποιας συνάρτησης κόστους. Πιο συγκεκριμένα, θεωρούμε πως ο αλγόριθμος K–means δέχεται ως είσοδο ένα σύνολο $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ d–διάστατων διανυσμάτων προς ταξινόμηση τα οποία κατά την διάρκεια της εκτέλεσής του θα προσπαθήσει να τα διαμερίσει σε K υποσύνολα του S έτσι ώστε να ισχύει ότι $S = \bigcup_{j=1}^{j=K} S_j$ όπου S_j είναι το j – οστό υποσύνολο του S και μ_j το αντίστοιχο κέντρο βάρους. Η συνάρτηση κόστους που χρησιμοποιείται από τον συγκεκριμένο αλγόριθμο είναι η παρακάτω:

$$f(S_1,\ldots,S_K) = \sum_{i=1}^{k} \sum_{x_j \in S_i} \left\| x_j - \mu_i \right\|^2 \quad \textbf{Εξίσωση 6}$$

Με άλλα λόγια ο στόχος του αλγορίθμου K – Means είναι να βρεί εκείνη τη διαμέριση του συνόλου S σε K υποσύνολα ώστε να ελαχιστοποιηθεί η παραπάνω συνάρτηση κόστους η οποία εκφράζει το άθροισμα των αποστάσεων για κάθε δεδομένο σημείο από το κέντρο της συστάδας στο οποίο αντιστοιχεί. Η ελαχιστοποίηση του παραπάνω συναρτησοειδούς πραγματοποιείται μέσω μιας επαναληπτικής διαδικασίας την οποία διατύπωσε ο Lloyd στον ομώνυμο αλγόριθμο. Τα βήματα του συγκεκριμένου αλγορίθμου είναι τα παρακάτω:

1. Αρχικοποίηση: Θεωρούμε ένα αρχικό σύνολο κέντρων βάρους $\left\{\mu_1^{(1)}, \ldots, \mu_K^{(1)}\right\}$ τα οποία θα μπορούσαν να έχουν παραχθεί με τυχαίο τρόπο.

2. Βήμα Ανάθεσης: Στο συγκεκριμένο βήμα καθένα από τα στοιχεία του συνόλου S ανατίθεται σε εκείνη την συστάδα για την οποία η απόσταση του συγκεκριμένου στοιχείου από το κέντρο βάρους της να

γίνεται ελάχιστη. Πιο συγκεκριμένα, κατά την t – οστή εκτέλεση του συγκεκριμένου βήματος θα ισχύει ότι:

$$S = \bigcup_{j=1}^{j=K} S_j^{(t)} \text{ όπου } S_j^{(t)} = \left\{ x_j : \left\| x_j - \mu_i^{(t)} \right\| \le \left\| x_j - \mu_{i^*}^{(t)} \right\| \right\}, \forall\, i^* \in [K]$$

Εξίσωση 7

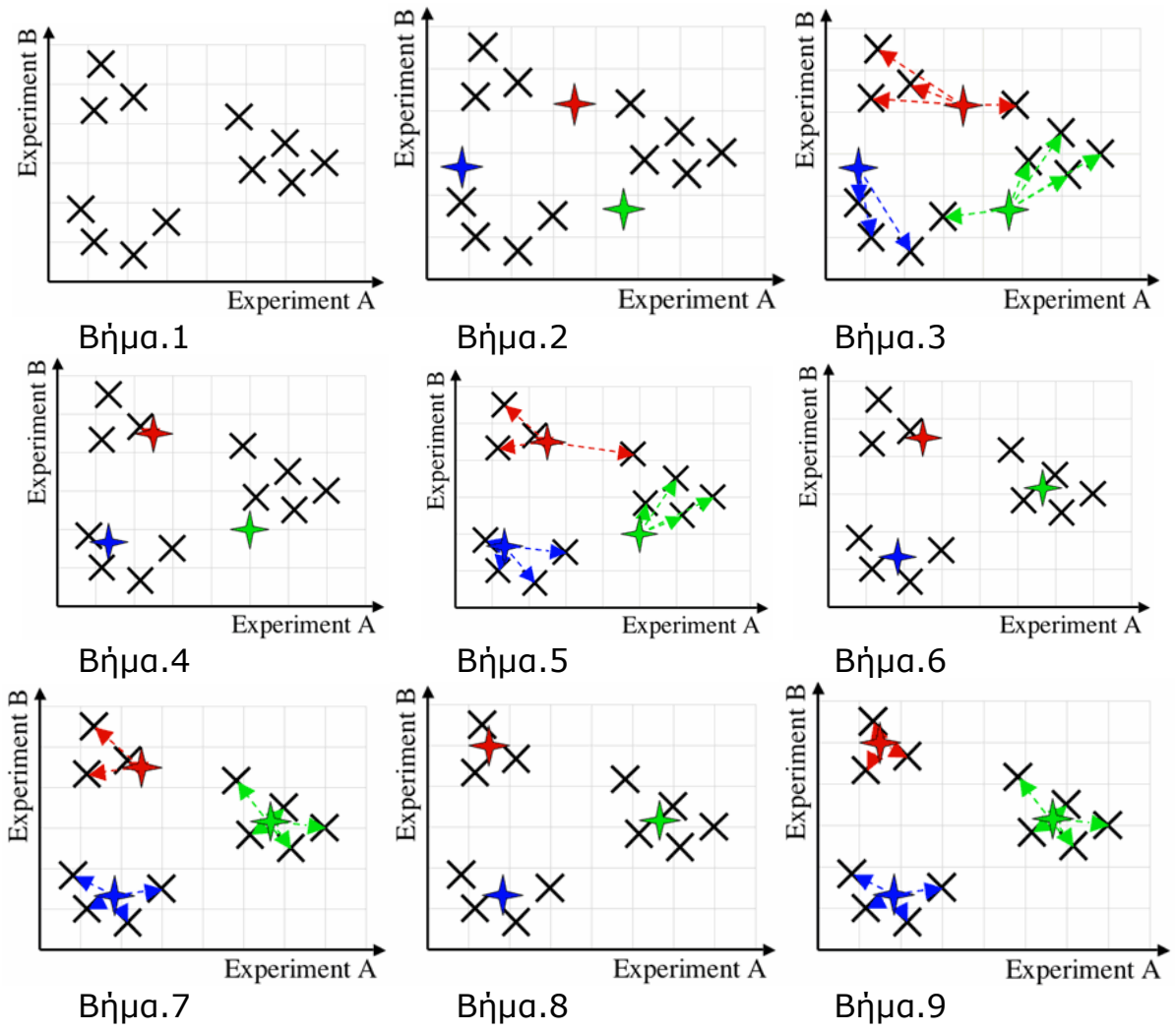
3. <u>Βήμα Ανανέωσης</u>: Κατά την εκτέλεση του συγκεκριμένου βήματος ο αλγόριθμος υπολογίζει εκ νέου τα κέντρα βάρους των συστάδων όπως αυτές θα έχουν διαμορφωθεί μετά την ολοκλήρωση του βήματος 2. Πιο συγκεκριμένα, κατά την t – οστή εκτέλεση του συγκεκριμένου βήματος θα ισχύει ότι:

$$\mu_i^{(t+1)} = \frac{1}{\left|S_i^{(t)}\right|} \sum_{x_j \in S_i^{(t)}} x_j$$

Εξίσωση 8

1.5.3 Τα Βήματα του αλγορίθμου k-means

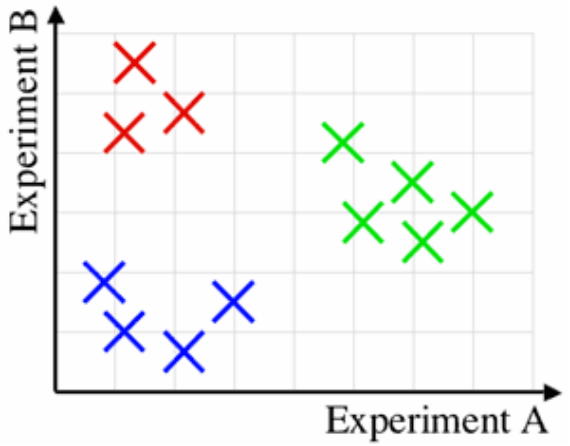
Στην ενότητα αυτή θα αναλύσουμε με την βοήθεια των γραφημάτων τα βήματα του αλγορίθμου k-means. [14]



Εικόνα 3 Τα βήματα του αλγορίθμου k-means

Στο *πρώτο* βήμα βλέπουμε τρεις ομάδες από παρόμοια στοιχεία. Ο αλγόριθμος K-means στοχεύει στον εντοπισμό των κέντρων βάρους , έτσι ώστε ένα αντικείμενο που ανήκει σε μία κλάση χ να είναι πιο κοντά στο κέντρο βάρους της κλάσης Χ παρά στο κέντρο βάρους των άλλων ομάδων. Ο αλγόριθμος απαιτεί μια σειρά ομάδων για να ξεκινήσει, σε αυτήν την περίπτωση 3, στο *δεύτερο* βήμα τοποθετούνται τα κέντρα βάρους σε τυχαίες θέσεις. Στη συνέχεια, στο *τρίτο* βήμα ο αλγόριθμος αποδίδει σε κάθε κέντρο βάρους όλα τα στοιχεία που είναι πιο κοντά σε αυτό από ό, τι σε οποιοδήποτε άλλο κέντρο βάρους. Τα κέντρα βάρους μετακινούνται στο κέντρο των κλάσεων, αυτό φαίνεται στο *τέταρτο* βήμα. Στο *πέμπτο* βήμα επαναλαμβάνεται το Βήμα 3 και επαναπροσδιορίζονται οι θέσεις των

κέντρων βάρους. Στη συνέχεια τα κέντρα βάρους μετακινούνται ξανά στο κέντρο της κάθε κλάσης, αυτό φαίνεται στο *έκτο* βήμα. Στο έβδομο βήμα ο αλγόριθμος αποδίδει σε κάθε κέντρο βάρους όλα τα στοιχεία που είναι πιο κοντά σε αυτό από ό, τι σε οποιοδήποτε άλλο κέντρο βάρους λαμβάνοντας υπόψη τις νέες θέσεις των κέντρων βάρους. Στο όγδοο βήμα μετακινούνται ξανά τα κέντρα βάρους στο κέντρο της κάθε κλάσης. Τέλος επαναλαμβάνεται το Βήμα 3 λαμβάνοντας υπόψη τις νέες θέσεις των κέντρων βάρους. Αυτή την φορά όμως δεν αλλάζει κάτι επομένως ο αλγόριθμος δεν θα επαναληφθεί ξανά.



Εικόνα 4 Οι τελικές κλάσεις που δημιουργήθηκαν. [14]

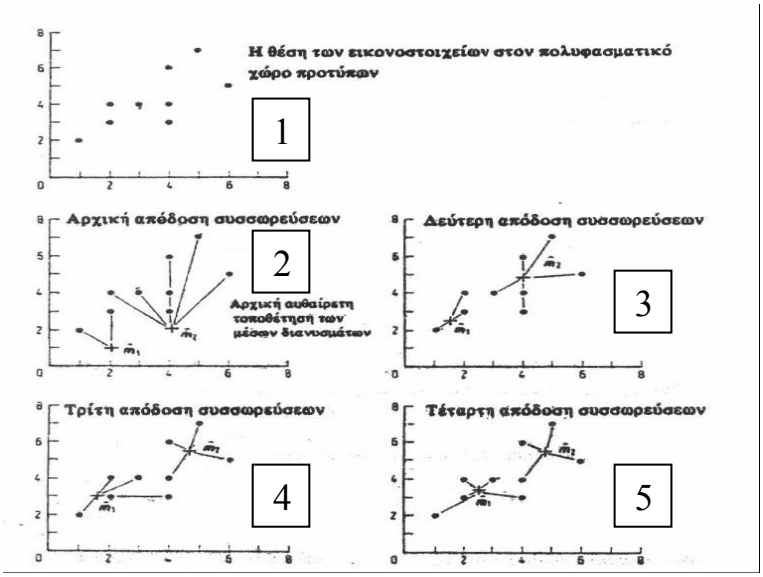
1.6 Ο Αλγόριθμος ISODATA

1.6.1 Συνοπτική περιγραφή του αλγορίθμου ISODATA[11]

Ο αλγόριθμος ISODATA προτάθηκε την δεκαετία του 1960 από τους Ball, Hall και άλλους και αποτελεί μια εξέλιξη του αλγορίθμου K–means. Τα βασικότερα σημεία νεοτερισμού του αλγορίθμου ISODATA σε σχέση με τον πρόγονό του συνίστανται στην δυνατότητα διάσπασης μιας συστάδας καθώς και την επεξεργασία της διάχυσης των κέντρων βάρους των συστάδων κατά την διάρκεια της εκτέλεσης του αλγορίθμου. Πιο συγκεκριμένα ο αλγόριθμος ISODATA διαθέτει την δυνατότητα να ελέγχει την πυκνότητα των δεδομένων τα οποία φιλοξενεί μια συγκεκριμένη συστάδα μέσω των διαδικασιών της διάσπασης και της συγχώνευσης των συστάδων που δημιουργούνται κατά την εκτέλεση του αλγορίθμου K-means. Στην

πραγματικότητα υπάρχουν πολλές εφαρμογές του αλγορίθμου, μια από αυτές είναι και αυτή του Paul M. Mather ο οποίος χρησιμοποίησε τον ISODATA έτσι ώστε να κατηγοριοποιήσει δεδομένα από αεροφωτογραφίες (1) , ή οι Stephanie A. Bany and Jerome E. Freier του πανεπιστημίου του Colorado (2) , οι οποίοι κατηγοριοποίησαν τα ζώα του Colorado σύμφωνα με τον αριθμό των δεδομένων που βρίσκονται σε κάθε κλάση, ένα ακόμα παράδειγμα θα μπορούσε να θεωρηθεί και αυτό του κ. Ηλία Δημητρίου από το Ελληνικό Κέντρο Θαλασσίων Ερευνών – Ινστιτούτο Εσωτερικών Υδάτων, ο οποίος χρησιμοποίησε τον αλγόριθμο για να μελετήσει, με χρήση τηλεματικών προϊόντων την αποτύπωση μεταβολών χρήσεων γης και την διαχείριση των υδατικών πόρων της υδρολογικής λεκάνης της λίμνης Τριχωνίδας [6].

Στην παρακάτω εικόνα μπορούμε να διακρίνουμε σε μία απλουστευμένη μορφή τα βασικά βήματα του ISODATA. Στο πρώτο διάγραμμα διακρίνουμε την θέση των δεδομένων στον χώρο ,στο δεύτερο τοποθετούνται τυχαία κέντρα βάρους έτσι ώστε να υπολογιστούν οι αποστάσεις των σημείων από τα κέντρα αυτά και εκείνα που είναι πιο κοντά να αποδοθούν και στις αντίστοιχες κλάσεις. Στο τρίτο γράφημα μετακινούνται τα κέντρα βάρους και επαναϋπολογίζονται οι αποστάσεις των σημείων. Σε περίπτωση που μία κλάση έχει περισσότερα στοιχεία από αυτά που έχει ορίσει ο χρήστης, τότε η κλάση πρέπει να διασπαστεί σε δύο, αντίθετα αν η κλάση περιέχει λιγότερα δεδομένα από αυτά που ο χρήστης όρισε ως ελάχιστα η κλάση θα πρέπει να συγχωνευθεί με την πλησιέστερη της. Η διαδικασία αυτή συνεχίζεται έως ότου δεν υπάρξει καμία μεταβλητή στις κλάσεις (γραφήματα 4 και 5).



Εικόνα 5 Τα βήματα του αλγορίθμου ISODATA [1]

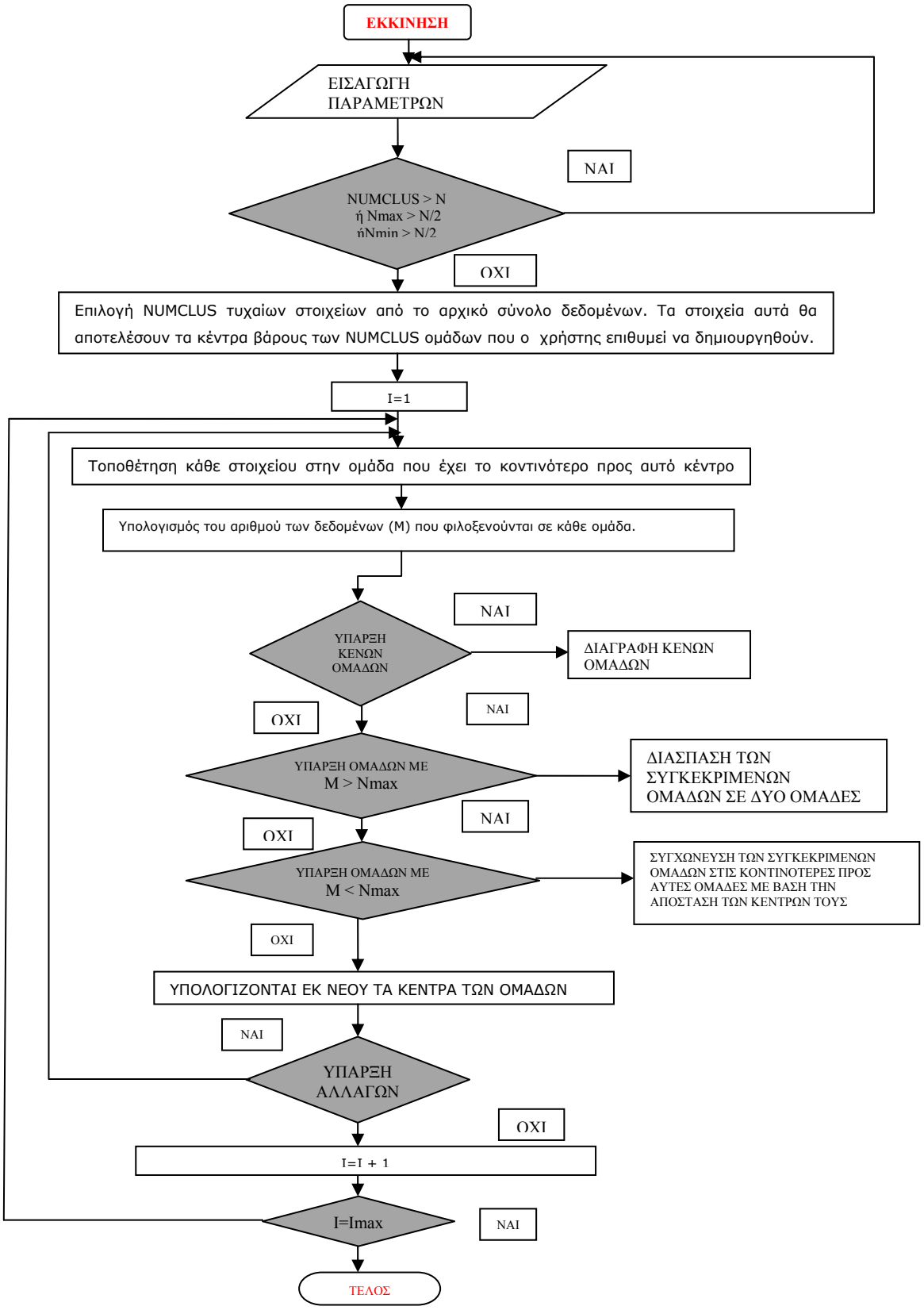
Στη συνέχεια ακολουθεί μια συνοπτική περιγραφή του αλγορίθμου ISODATA [16]

1. Καθορισμός του επιθυμητού αριθμού των ομάδων – (NUMCLUS) - , του μέγιστου – Nmax – και ελάχιστου – Nmin - αριθμού δεδομένων σε μία ομάδα, καθώς και του μέγιστου αριθμού -Imax-επαναλήψεων του αλγορίθμου. (ΠΕΡΙΟΡΙΣΜΟΙ : Ο αριθμός των ομάδων δεν πρέπει να ξεπερνά το πλήθος των προς ταξινόμηση στοιχείων, ο μέγιστος αριθμός δεδομένων το μισό του πλήθους των δεδομένων – διαφορετικά θα δημιουργηθούν λίγες ομάδες με μεγάλο αριθμό στοιχείων καταχωρημένο σε αυτές – και ο ελάχιστος αριθμός να μην ξεπερνά τόσο το μισό του πλήθους των δεδομένων όσο και τον μέγιστο αριθμό δεδομένων κάθε ομάδας.)
2. Επιλογή NUMCLUS τυχαίων στοιχείων από το αρχικό σύνολο δεδομένων. Τα στοιχεία αυτά θα αποτελέσουν τα κέντρα βάρους των NUMCLUS ομάδων που ο χρήστης επιθυμεί να δημιουργηθούν.
3. Τοποθέτηση κάθε στοιχείο στην ομάδα που έχει το κοντινότερο προς αυτό κέντρο βάρους.
4. Αν δημιουργηθούν κενές ομάδες (ομάδες που περιέχουν μόνο το κέντρο βάρους και κανένα άλλο στοιχείο), διαγράφονται και το στοιχείο-κέντρο βάρους τοποθετείται στην κοντινότερη ομάδα. Αν

δημιουργηθούν ομάδες με αριθμό δεδομένων μεγαλύτερο του Nmax, τότε θα πρέπει να χωριστούν σε δύο ομάδες. Αν δημιουργηθούν ομάδες με αριθμό δεδομένων μικρότερο του Nmin, τότε τα στοιχεία τους θα πρέπει να ενσωματωθούν στις κοντινότερες (με βάση τις αποστάσεις από τα κέντρα βάρους τους) ομάδες.

5. Τα κέντρα βάρους των ομάδων υπολογίζονται εκ νέου. Αν κάποιο άλλαξε, τότε πήγαινε στο βήμα 3. Σε διαφορετική περίπτωση ή εάν έχει φθάσει ο μέγιστος αριθμός επαναλήψεων του αλγορίθμου που έχει καθοριστεί από τον χρήστη, τερμάτισε.

Τα παραπάνω βήματα φαίνονται καλύτερα διάγραμμα ροής της εικόνας 6 που ακολουθεί.



Εικόνα 6 Διάγραμμα Ροής Αλγορίθμου ISODATA

1.6.2 Μαθηματική περιγραφή του αλγορίθμου ISODATA

Παρακάτω με την βοήθεια του διαγράμματος ροής που φαίνεται στην εικόνα 5 θα προσπαθήσουμε να αναλύσουμε τα βήματα του αλγορίθμου ISODATA και στη συνέχεια ακολουθεί η μαθηματική περιγραφή του αλγορίθμου ISODATA η οποία ξεκινά με την δήλωση όλων των παραμέτρων που ο αλγόριθμος χρησιμοποιεί εσωτερικά προκειμένου να λειτουργήσει[16]. (Εικόνα 6)

- **NUMCLUS**: ο αρχικός αριθμός των ομάδων.
- **SAMPAR**: ο ελάχιστος αριθμός δεδομένων που μπορεί να περιλαμβάνει μία ομάδα.
- **MAXITER:** ο μέγιστος αριθμός επαναλήψεων του αλγορίθμου.
- **STDV**: η μέγιστη τυπική απόκλιση των σημείων από το κέντρο βάρους της ομάδας κατά μήκος κάθε άξονα.
- **LUMP**: η ελάχιστη απαιτούμενη απόσταση μεταξύ των κέντρων βάρους δύο ομάδων.
- **MAXPAIR**: ο μέγιστος αριθμός από ζεύγη ομάδων τα οποία μπορούν να συμπεριληφθούν σε μία επανάληψη.

Θεωρούμε το σύνολο $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ των προς κατηγοριοποίηση «αμαρκάριστων» δεδομένων. Καθένα από τα σημεία $\mathbf{x}_j = \left(x_{j1}, \ldots, x_{jn}\right)$ του συνόλου **S** μπορεί να θεωρηθεί ως ένα πραγματικό διάνυσμα του χώρου $\mathbf{R}^d$. Το πλήθος των διαθέσιμων προς κατηγοριοποίηση δεδομένων θα υποδηλώνεται από την παράμετρο **n.** Στην περίπτωση κατά την οποία το αρχικό πλήθος των δεδομένων είναι αρκετά μεγάλο, τότε όλες οι επαναλήψεις του αλγορίθμου εκτός της τελευταίας μπορούν να εκτελεστούν σε ένα τυχαίο υποσύνολο του αρχικού συνόλου **S** με μικρότερη τάξη μεγέθους.

1. Αρχικά θεωρούμε πως ο αρχικός αριθμός των ομάδων **k** που θέλουμε να δημιουργήσει ο αλγόριθμος είναι ίσος με τη τιμή της παραμέτρου **NUMCLUS**. Για το σκοπό αυτό επιλέγουμε **k** τυχαία σημεία από το αρχικό σύνολο **S** των δεδομένων $Z = \{\mathbf{z}_1, \ldots, \mathbf{z}_k\}$, τα οποία θα αποτελέσουν τα κέντρα βάρους των **k** θεωρούμενων ομάδων.

2. Το κάθε σημείο του αρχικού συνόλου δεδομένων ανατίθεται στην ομάδα που αντιπροσωπεύεται από το πλησιέστερο προς αυτό κέντρο. Δηλαδή, για κάθε **i** τέτοιο ώστε $1 \le i \le k$, θωρούμε πως το $S_i \subseteq S$ είναι το υποσύνολο των σημείων που είναι πλησιέστερα προς το κέντρο $\mathbf{z}_i$. Αυτό σημαίνει πως για κάθε $\mathbf{x} \in S$ ισχύει η παρακάτω σχέση:

$$\mathbf{x} \in S_i \ \alpha\nu \ \left\|\mathbf{x} - \mathbf{z}_j\right\| < \left\|\mathbf{x} - \mathbf{z}_i\right\|, \forall\, i \neq j$$

Στην περίπτωση κατά την οποία δύο ή περισσότερα κέντρα ισαπέχουν από ένα δεδομένο σημείο τότε επιλέγεται τυχαία κάποιο από αυτά. Θεωρούμε επιπλέον πως $\mathbf{n_j}$ είναι το πλήθος των σημείων που περιέχονται σε κάθε ομάδα.

3. Εκείνα τα κέντρα βάρους για τα οποία ο αριθμός των πλησιέστερων προς αυτά σημείων του συνόλου **S** ήταν λιγότερος από **SAMPAR** θα πρέπει να διαγραφούν. Ωστόσο, τα αντίστοιχα σημεία του **S** δεν διαγράφονται αλλά αγνοούνται μέχρι την ολοκλήρωση της τρέχουσας επανάληψης. Στη συνέχεια, η τιμή της παραμέτρου **k** θα πρέπει να ενημερωθεί μιας και ο αριθμός των θεωρούμενων ομάδων μειώθηκε και αντίστοιχα θα πρέπει να ενημερωθούν και οι δείκτες των συνόλων των ομάδων $S_1, \ldots, S_k$ που έχουν απομείνει σύμφωνα με την νέα τιμή του **k**.
4. Τα κέντρα βάρους των θεωρούμενων ομάδων θα πρέπει να μετακινηθούν προς το κεντροειδές του αντίστοιχου συνόλου δεδομένων σύμφωνα με την εξίσωση:

$$\mathbf{z}_j \leftarrow \frac{1}{n_j} \sum_{\mathbf{x} \in S_j} \mathbf{x}, \ \text{όπου} \ 1 \le j \le k \quad \textbf{Εξίσωση 9}$$

Αν κατά την εκτέλεση του βήματος 3 απαιτήθηκε η διαγραφή κάποιων ομάδων τότε ο αλγόριθμος θα πρέπει να επαναλάβει το βήμα 2 στο συγκεκριμένο σημείο.

5. Υπολογίζεται η παράμετρος $\mathbf{\Delta_j}$ που αντιστοιχεί στη μέση απόσταση των σημείων του συνόλου $\mathbf{S_j}$ από το κέντρο της ομάδας $\mathbf{z}_j$. Ακολούθως υπολογίζεται η παράμετρος **Δ** που αντιστοιχεί στη μέση τιμή όλων των

προηγούμενων αποστάσεων. Οι τιμές των παραμέτρων **Δ_j** και **Δ** υπολογίζονται σύμφωνα με τις εξισώσεις:

$$\Delta_j \leftarrow \frac{1}{n_j} \sum_{x \in S_j} \left\| x - z_j \right\|, \text{ όπου } 1 \le j \le k \quad \textbf{Εξίσωση 10}$$

$$\Delta \leftarrow \frac{1}{n} \sum_{j=1}^{k} n_j \Delta_j \quad \textbf{Εξίσωση 11}$$

6. Στην περίπτωση κατά την οποία εκτελείται η τελευταία επανάληψη του αλγορίθμου, τότε η τιμή 0 εκχωρείται στην παράμετρο **LUMP (LUMP = 0)** και η εκτέλεση του αλγορίθμου συνεχίζεται από το βήμα 9. Επιπλέον, αν $2k > NUMCLUS$ και πρόκειται για άρτιο αριθμό επανάληψης του αλγορίθμου ή $k \ge 2NUMCLUS$, τότε και στην περίπτωση αυτή η εκτέλεση του αλγόριθμου θα πρέπει να συνεχίσει από το βήμα 9.
7. Για κάθε ομάδα **S_j**, υπολογίζεται ένα νέο διάνυσμα $v_j = (u_1, \ldots, u_d)$ του οποίου η ι – οστή συντεταγμένη αντιστοιχεί στην τυπική απόκλιση των ι – οστών συντεταγμένων των διανυσμάτων τα οποία κατευθύνονται από το εκάστοτε κέντρο βάρους z_j προς καθένα από τα σημεία του συνόλου **S_j** σύμφωνα με την παρακάτω εξίσωση:

$$u_{ji} \leftarrow \sqrt{\frac{1}{n_j} \sum_{x \in S_j} \left(x_i - z_{ji} \right)^2}, \text{ όπου } 1 \le j \le k \text{ και } 1 \le i \le d. \quad \textbf{Εξίσωση 12}$$

Θεωρούμε πως η μεταβλητή $v_{j,\max}$ υποδηλώνει την μέγιστη συντεταγμένη του διανύσματος v_j.

8. Για κάθε ομάδα **S_j** , στην περίπτωση κατά την οποία έχουμε $v_{j,\max} > STDV$ και ισχύει κάτι από τα παρακάτω:

$$\left(\left(\Delta_j > \Delta \right) \text{ και } \left(n_j > 2\left(n_{\min} + 1 \right) \right) \right) \text{ ή } k \le \frac{k_{init}}{2} \quad \textbf{Εξίσωση 13}$$

τότε ο αριθμός των θεωρούμενων κέντρων βάρους θα πρέπει να αυξηθεί και το σύνολο **S_j** να διασπαστεί σε δύο ομάδες. Τότε, το κέντρο βάρους της ομάδας **S_j** , θα πρέπει να αντικατασταθεί από δύο σημεία τα οποία θα βρίσκονται στην γειτονιά του z_j όπου το μέτρο

και η κατεύθυνση του διανύσματος που υποδηλώνει την μεταξύ τους απόσταση θα εξαρτάται από την παράμετρο $\mathbf{v}_{j,\max}$. Αν κατά την εκτέλεση του συγκεκριμένου βήματος κριθεί αναγκαία η διάσπαση κάποιας ομάδας τότε ο αλγόριθμος θα πρέπει να συνεχίσει με την εκτέλεση του βήματος 2.

9. Υπολογίζονται οι τιμές των αποστάσεων μεταξύ όλων των ζευγαριών των θεωρούμενων κέντρων βάρους όπως περιγράφεται από την παρακάτω εξίσωση:

$$\mathbf{d}_{ij} \leftarrow \left\| \mathbf{z}_i - \mathbf{z}_j \right\|, \text{ όπου } 1 \le i < j \le k. \quad \textbf{Εξίσωση 14}$$

10. Ταξινόμηση των αποστάσεων που υπολογίστηκαν στο προηγούμενο βήμα σε αύξουσα σειρά. Από το σύνολο αυτό επιλέγεται ένα υποσύνολο **MAXPAIR** το πολύ αποστάσεων, τα οποία αντιστοιχούν σε ζεύγη ομάδων που η μεταξύ τους απόσταση δεν υπερβαίνει την τιμή **STDV**. Για κάθε τέτοιο ζεύγος συνόλων (**S_i**, **S_j**) αν καμία από τις θεωρούμενες ομάδες δεν έχουν εμπλακεί σε κάποια διαδικασία συγχώνευσης τότε τα σύνολα των ομάδων **S_i**, **S_j** αντικαθίστανται από την συγχωνευμένη ομάδα $S_i \cup S_j$. Το κέντρο βάρους της νέας ομάδας θα υπολογίζεται σύμφωνα με την εξίσωση:

$$\mathbf{z}_{ij} \leftarrow \frac{1}{n_i + n_j}\left(n_i \mathbf{z}_i + n_j\right) \quad \textbf{Εξίσωση 15}$$

Στη συνέχεια, η τιμή της παραμέτρου **k** θα πρέπει να ενημερωθεί μιας και ο αριθμός των θεωρούμενων ομάδων μειώθηκε και αντίστοιχα θα πρέπει ναν ενημερωθούν και οι δείκτες των συνόλων των ομάδων $S_1, \ldots, S_k$ που έχουν απομείνει σύμφωνα με την νέα τιμή του **k**.

11. Αν ο αριθμός των επαναλήψεων είναι μικρότερος από τη τιμή **MAXITER** τότε θα η εκτέλεση του αλγορίθμου θα συνεχιστεί απο το βήμα 2.

1.6.4 Πλεονεκτήματα και Μειονεκτήματα του αλγορίθμου ISODATA

Τα βασικότερα **πλεονεκτήματα** του αλγορίθμου ISODATA είναι τα παρακάτω:

- Έχει ενσωματωμένη την ιδιότητα της αυτοοργάνωσης.
- Διαθέτει την ευελιξία στο να καταργεί εκείνες τις συστάδες των δεδομένων με τα λιγότερα δείγματα.
- Έχει την ικανότητα να διαιρεί εκείνες τις συστάδες των δεδομένων για τις οποίες η ανομοιότητα μεταξύ των δειγμάτων τους είναι μεγαλύτερη από κάποιο προκαθορισμένο κατώφλι.
- Έχει την ικανότητα να συγχωνεύει εκείνες τις συστάδες των δεδομένων για τις οποίες η ομοιότητα μεταξύ των δειγμάτων τους είναι μεγαλύτερη από κάποιο προκαθορισμένο κατώφλι.

Τα βασικότερα **μειονεκτήματα** του αλγορίθμου ISODATA είναι τα παρακάτω:

- Τα δεδομένα θα πρέπει να είναι γραμμικώς διαχωρίσιμα. Η συγκεκριμένη απαίτηση γίνεται περισσότερο κατανοητή αν τα προς συσταδοποίηση δεδομένα θεωρηθούν ως σημεία σε κάποιο διανυσματικό χώρο ανώτερης διάστασης του δύο όπου η γραμμική διαχωριστικότητα μεταφράζεται στην ύπαρξη υπερεπιπέδων τα οποία μπορούν να ξεχωρίζουν τα στιγμιότυπα των δεδομένων που ανήκουν σε διαφορετικές συστάδες.
- Είναι δύσκολος ο προσδιορισμός των καταλληλότερων αρχικών παραμέτρων του αλγορίθμου. Ο αλγόριθμος ISODATA είναι εξαιρετικά ευαίσθητος στις τιμές των αρχικών παραμέτρων τις οποίες δέχεται ως είσοδο και κατά συνέπεια θα πρέπει να καθοριστούν από τον χρήστη. Πιο συγκεκριμένα ο αλγόριθμος ISODATA με είσοδο δυο διαφορετικά σετ αρχικών συνθηκών θα δώσει ως έξοδο δυο διαφορετικά στιγμιότυπα συσταδοποίησης του ίδιου συνόλου δεδομένων.

Κεφάλαιο 2. Ανάπτυξη Εφαρμογής

Σκοπός αυτής της πτυχιακής εργασίας είναι η συγκριτική παρουσίαση των αποτελεσμάτων συσταδοποίησης που παράγονται κατά την δράση των αλγορίθμων K–means και ISODATA επάνω σε πραγματικές συλλογές δεδομένων. Στο κεφάλαιο αυτό θα περιγραφούν οι προγραμματιστικές λεπτομέρειες που αφορούν την δημιουργία της παραθυρικής εφαρμογής η οποία αναλαμβάνει:

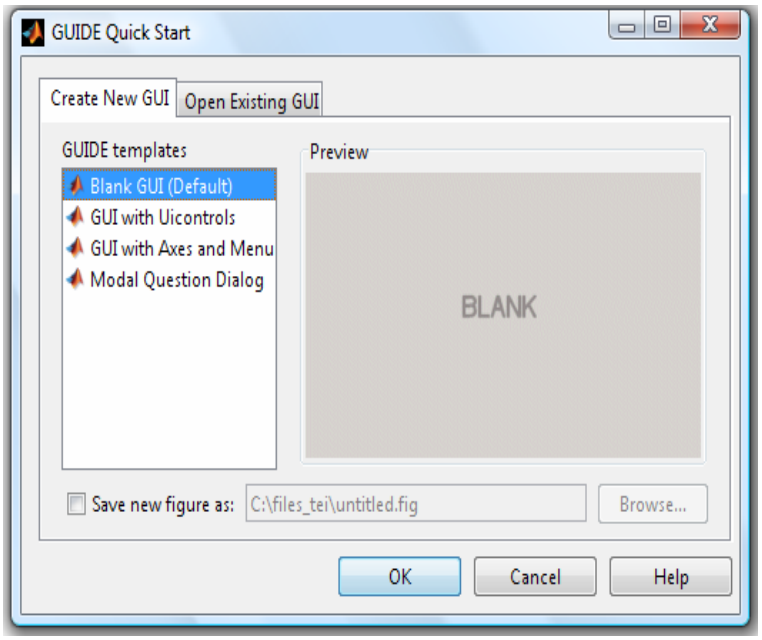
- την σύνδεση με τις βάσεις των δεδομένων στις οποίες είναι οργανωμένες οι διαθέσιμες συλλογές.
- την εισαγωγή των κατάλληλων παραμέτρων για την εκτέλεση των αλγορίθμων.
- την εκτέλεση των αλγορίθμων.
- την οπτικοποίηση των αποτελεσμάτων που παράγουν οι υπό εξέταση αλγόριθμοι συσταδοποίησης.

Επιπλέον θα παρουσιαστούν τα τμήματα κώδικα τα οποία αντιστοιχούν σε καθεμία από τις παραπάνω λειτουργίες καθώς και αυτά που αντιστοιχούν στην υλοποίηση τόσο του αλγόριθμου K–means όσο και του αλγόριθμου ISODATA. Θα παρουσιαστούν επίσης τα τμήματα κώδικα των βοηθητικών συναρτήσεων που αναπτύχθηκαν μέσα στα πλαίσια της εφαρμογής ανάμεσα στις οποίες συγκαταλέγεται και η συνάρτηση η οποία αναλαμβάνει να εφαρμόσει την τεχνική της ανάλυσης των κυρίων συνιστωσών προκειμένου να οπτικοποιηθούν τα αποτελέσματα της συσταδοποίησης.

Στην παράγραφο 2.1 εξηγούμε τον τρόπο δημιουργίας της εφαρμογής με Matlab. Στην παράγραφο 2.2 παρουσιάζουμε τις επιλογές και τα βήματα που θα ακολουθήσει ο χρήστης για να λειτουργήσει την εφαρμογής μας.

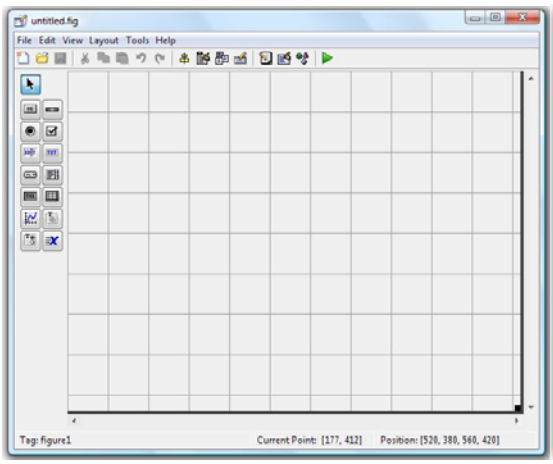
2.1 Γραφικό περιβάλλον Matlab

Για να ξεκινήσουμε να εργαζόμαστε θα πρέπει να δημιουργήσουμε ένα γραφικό περιβάλλον σε MATLAB, αυτό γίνεται είτε πληκτρολογώντας την εντολή guide, είτε επιλέγοντας από το μενού File ->New->GUI .Κάνοντας μία από αυτές τις ενέργειες θα μας εμφανιστεί το παράθυρο της εικόνας 7.

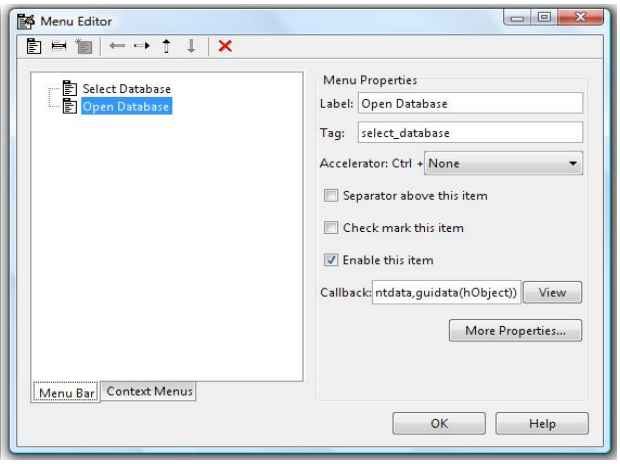


Εικόνα 7 Άνοιγμα ενός νέου GUIDE

Το παράθυρο αυτό μας δίνει τη δυνατότητα είτε να δημιουργήσουμε ένα νέο guide είτε να τροποποιήσουμε ένα ήδη υπάρχον. Στην περίπτωση μας θα επιλέξουμε Blank GUI (Default).

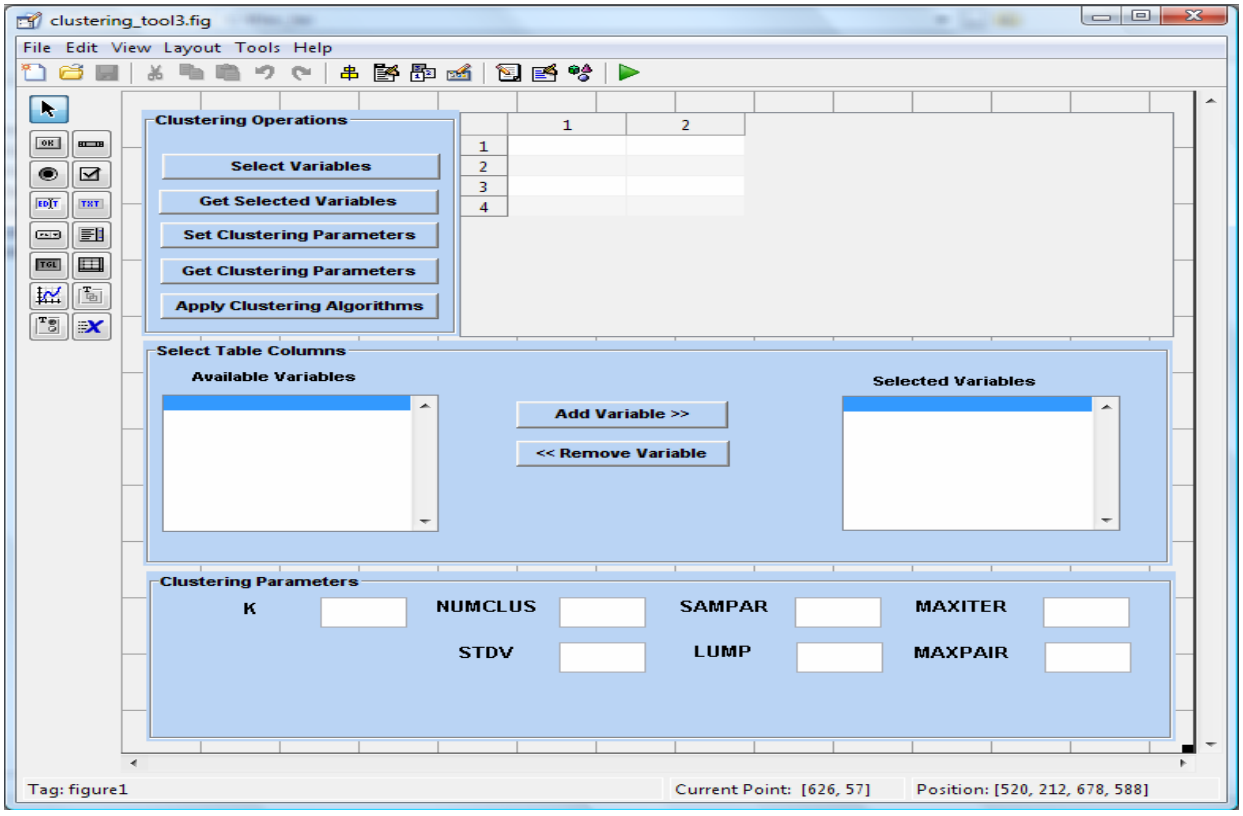


Εικόνα 8 Κενο GUI



Εικόνα 9 Menu Editor

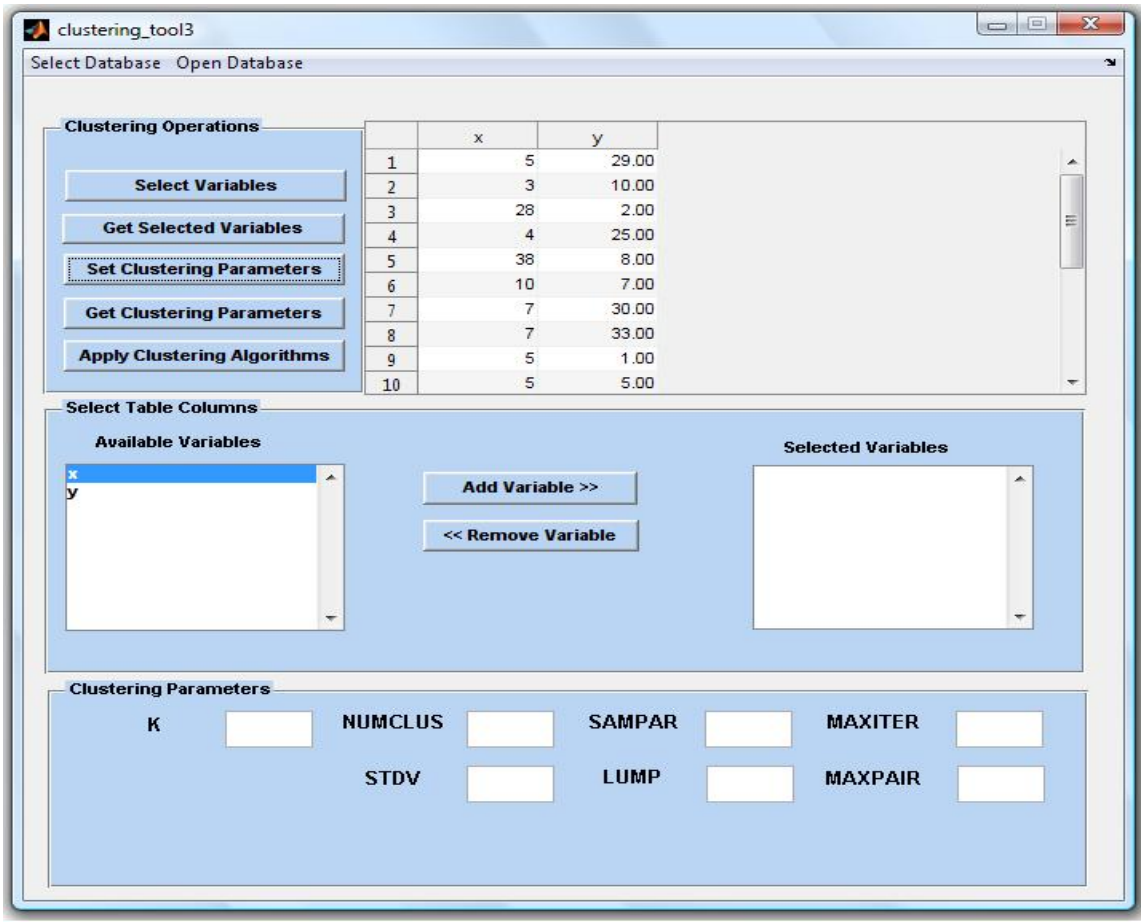
Στην εικόνα 8 βλέπουμε πως στην αριστερή πλευρά υπάρχουν διαφορά components που μπορούμε να τοποθετήσουμε στην φόρμα μας ώστε να υλοποιήσουμε την εφαρμογή. Επίσης χρησιμοποιήσαμε τον Menu Editor έτσι ώστε τοποθετήσουμε δύο Menu (στην περίπτωση μας τα Select Database και Open Database) για να μπορεί ο χρήστης να επιλέξει την βάση δεδομένων την οποία θα χρησιμοποιήσει και στην συνέχεια να την ανοίξει (εικόνα 9). Η παρούσα εφαρμογή που υλοποιήθηκε στα πλαίσια αυτής της εργασίας έχει την μορφή που απεικονίζεται στην εικόνα 9. Στην φόρμα έχουν τοποθετηθεί διάφορα components, όπως panel, table, pushbutton, listbox, και text. Κάθε component έχει μία σειρά από συναρτήσεις (functions) τις οποίες καλεί ανάλογα με τις επιλογές του χρήστη. Η τελική μορφή της φαίνεται στην εικόνα 10 που ακολουθεί.



Εικόνα 10 Τελική μορφή εφαρμογής

2.2 Βασική καρτέλα της εφαρμογής

Ας δούμε αναλυτικότερα την λειτουργία του προγράμματος για κάθε αντικείμενο που προστέθηκε στην εφαρμογή μας. Στην εικόνα 11 παρουσιάζονται οι επιλογές που έχει ο χρήστης και τα κουμπιά που πρέπει να πατήσει για να λειτουργήσει η εφαρμογή.



Εικόνα 11 Η εφαρμογή ολοκληρωμένη

Select Database: Πρόκειται για ένα Menu αντικείμενο. Εδώ ο χρήστης πατώντας το κουμπί αυτό καλείται να επιλέξει μία βάση δεδομένων από τις ήδη υπάρχουσες. Ο κώδικας που εκτελείται ανήκει στην συνάρτηση Clustering_tool3 και είναι ο εξής.

```
function open_database_Callback(hObject, eventdata, handles)
[FileName,PathName] = uigetfile({'*.mdb';'*.xls'},'Select the database to
connect');
mydata = guidata(hObject);
mydata.FileName = FileName;
guidata(hObject,mydata);
```

Με την συνάρτηση uigetfile, μπορούμε να ψάξουμε όλα τα αρχεία που βρίσκονται στον υπολογιστή, προκειμένου να βρούμε μια βάση δεδομένων για να συνδεθούμε (excel, access).

Open Database: Πρόκειται για ένα Menu αντικείμενο. Εδώ ο χρήστης πατώντας το κουμπί αυτό κάνει ορατή στην εφαρμογή μας την βάση που επέλεξε. Ο κώδικας που εκτελείται ανήκει στην συνάρτηση Clustering_tool3 και είναι ο εξής:

```
function select_database_Callback(hObject, eventdata, handles)
mydata = guidata(hObject);
DatabaseName = strtok(mydata.FileName,'.')
DatasourceName = strcat(DatabaseName,'_source')
```

Σύνδεση με την βάση δεδομένων.

```
conn = database(DatasourceName,'','')
```

Κάνει Ping με την βάση δεδομένων για να δει την κατάσταση την σύνδεσης.

```
ping(conn)
command = strcat(['select * from ',DatabaseName])
curs = exec(conn,command);
```

Μετατρέπει ττα περιεχόμενα της βάσης δεδομένων σε ένα cell array .

```
setdbprefs('DataReturnFormat','cellarray');
curs = fetch(curs)
colnames = columnnames(curs)
```

Επιστρέφει ένα cell array από strings το οποίο περιέχει τα ονόματα των στηλών στον πίνακα.

```
colnum = 0;
remain = colnames;
col_names = {};
while(true)
    [token,remain] = strtok(remain,',');
    if(isempty(token))
        break;
    else
        colnum = colnum + 1;
        token = token(2:1:end-1);
        col_names{colnum} = token;
    end;
end;
data = cell2mat(curs.Data)
whos curs
```

Κλείνει την σύνδεση με την βάση.

```
close(conn)

columnformat = {'numeric', 'bank', []};
set(mydata.uitable1,'ColumnFormat',columnformat);
set(mydata.uitable1,'Data',data);
set(mydata.uitable1,'ColumnName',col_names);
set(mydata.uitable1,'Visible','on');
```

Select Variables: Πρόκειται για ένα Puhsbutton αντικείμενο. Εδώ ο χρήστης πατώντας το κουμπί αυτό κάνει ορατή στην εφαρμογή μας τις διαθέσιμες λίστες από την βάση δεδομένων που διάλεξε ,έτσι ώστε να επιλέξει τις δυο από αυτές. Ο κώδικας που εκτελείται ανήκει στην συνάρτηση Clustering_tool3 και είναι ο εξής:

```
function pushbutton7_Callback(hObject, eventdata, handles)
mydata = guidata(hObject);
```

Κάνε ορατό το panel με τίτλο 'Select Table Columns'

```
set(mydata.uipanel3,'Visible','on'); col_names = mydata.col_names;
```

Παίρνει τα ονόματα της κάθε στήλης

Αρχικά αριστερά έχουμε όλα τα ονόματα

```
left_list = col_names;
```

και δεξιά τίποτα

```
right_list = {};
left_list_indices = [1:1:length(left_list)];
right_list_indices = [];
selected_variables_num = 0;
```

Αποθηκεύει τις παρακάτω μεταβλητές στο struct mydata και καλεί την guidata(),ώστε να γνωρίζουν τις αλλαγές και οι συναρτήσεις εκτός της ίδιας

```
mydata.selected_variables_num = selected_variables_num;
mydata.left_list = left_list;
mydata.right_list = right_list;
mydata.left_list_indices = left_list_indices;
mydata.right_list_indices = right_list_indices;
```

Βάζουμε στην αριστερή λίστα 'Available Variables' του συγκεκριμένου panel ότι έχουμε (δηλαδή όλα τα column names)

```
set(mydata.LeftList,'String',left_list,'Value',1);
```

Βάζουμε στην δεξιά λίστα 'Selected Variables' του συγκεκριμένου panel ότι έχουμε (δηλαδή στην ουσία τίποτα)

```
set(mydata.RightList,'String',right_list);
```

Set Clustering Parameters: Πρόκειται για ένα Puhsbutton αντικείμενο. Εδώ ο χρήστης πατώντας το κουμπί αυτό κάνει ορατή στην εφαρμογή μας τις

μεταβλητές που θα καλεστεί ο χρήστης να εισάγει. Ο κώδικας που εκτελείται ανήκει στην συνάρτηση Clustering_tool3 και είναι ο εξής:

```
function pushbutton1_Callback(hObject, eventdata, handles)
mydata = guidata(hObject);
set(mydata.uipanel2,'Visible','on');
```

Get Clustering Parameters: Πρόκειται για ένα Puhsbutton αντικείμενο. Εδώ ο χρήστης πατώντας το κουμπί αυτό αποθηκεύει τις μεταβλητές που εισήγαγε. Ο κώδικας που εκτελείται ανήκει στην συνάρτηση Clustering_tool3 και είναι ο εξής:

```
function pushbutton2_Callback(hObject, eventdata, handles)
```

Από τα edit boxes μπορούμε να διαβάσουμε μόνο συμβολοσειρές (strings). Οπότε με την συνάρτηση get() γνωρίζουμε το περιεχόμενο του
edit box, αλλά ως συμβολοσειρά. Επειδή οι συγκεκριμένοι παράμετροι είναι αριθμοί, με την str2double τους μετατρέπουμε σε αριθμούς

```
mydata = guidata(hObject);
Kstring = get(mydata.k_edit,'String');
NUMCLUSstring = get(mydata.numclus_edit,'String'
SAMPARstring = get(mydata.sampar_edit,'String');
 MAXITERstring = get(mydata.maxiter_edit,'String'
STDVstring = get(mydata.stdv_edit,'String');
LUMPstring = get(mydata.lump_edit,'String'
MAXPAIRstring = get(mydata.maxpair_edit,'String'
```

Μετατρέπει τα strings σε αριθμητικές τιμές.

```
k = str2num(Kstring);
numclus = str2double(NUMCLUSstring);
sampar = str2double(SAMPARstring);
maxiter = str2double(MAXITERstring);
stdv = str2double(STDVstring);
lump = str2double(LUMPstring);
maxpair = str2double(MAXPAIRstring);
```

Apply Clustering Parameters: Πρόκειται για ένα Puhsbutton αντικείμενο. Εδώ ο χρήστης πατώντας το κουμπί αυτό εκτελεί τους δύο αλγορίθμους και εξάγει τα τελικά αποτελέσματα σε μορφή γραφημάτων. Ο κώδικας που εκτελείται ανήκει στην συνάρτηση Clustering_tool3 και είναι ο εξής:

```
function pushbutton3_Callback(hObject, eventdata, handles)
```

Με την εντολή addpath() μπορούμε να συμπεριλάβουμεόλα τα .m files που βρίσκονται στον συγκεκριμένο φάκελο. έπειτα μπορούμε να τακαλέσουμε ώστε να εκτελεσθούν

```
addpath('isomatlab4');
mydata = guidata(hObject);
```

Κανονικά το ίδιο κάνει και το button 'Get Selected Variables' του panel 'Clustering Operations', αλλά εδώ γίνεται ξανά προς σιγουριά.

Βλέπει τους δείκτες των μεταβλητών της δεξιάς λίστας

```
right_list = mydata.right_list;
right_list_indices = mydata.right_list_indices;
patterns = mydata.data;
patterns = patterns(:,right_list_indices);
```

Διαβάζει από όλα τα δεδομένα (patterns), μόνο εκείνες τις στήλες(δηλαδή μεταβλητές)που έχει επιλέξει ο χρήστης. Να θυμίσουμε ότι τα δεδομένα είναι αποθηκευμένα σε πίνακα 2 διαστάσεων. Η πρώτη διάσταση (γραμμές) αντιστοιχεί σε διαφορετικό δεδομένο. Δηλαδή το πρώτο πρότυπο είναι στην πρώτη γραμμή, το δεύτερο στην δεύτερη κοκ Η δεύτερη διάσταση (στήλες) αντιστοιχεί σε κάθε μεταβλητή. Δηλαδή οι τιμές της πρώτης μεταβλητής(x1) είναι αποθηκευμένες στην πρώτη στήλη κοκ.Άρα π.χ. το σημείο (10,2) μας δίνει την τιμή του δέκατου προτύπου για την δεύτερη μεταβλητή στην αρχή θα εκτελεστεί ο k-means αλγόριθμος και μετά ο ISODATA, ώστε να φανούν και οι διαφορές τους.

```
K = mydata.k
[KMeansClusterIndices,Centers] = kmeans(patterns,K);
```

Εδώ καλούμε την έτοιμη συνάρτηση k-means του Matlab. Γνωρίζουμε ότι: [IDX, C] = KMEANS(X, K) επιστρέφει τα κέντρα βάρους των K κλάσεων cluster centroid locations in the K-by-P matrix C.Άρα δίνουμε στην συνάρτηση την τιμή του K και τα δεδομένα μας (patterns) και μας επιστρέφει τα κέντρα βάρους των κλάσεων και τους αντίστοιχους δείκτες ,που μας δείχνουν που ακριβώς (σε ποιό cluster δηλαδή) έχει ταξινομηθεί κάθε δεδομένο. Συγκεκριμένα, αν IDX(4)=1, σημαίνει ότι το δεδομένο που

αντιστοιχεί στην τέταρτη σειρά των δεδομένων έχει ταξινομηθεί στο cluster με κέντρο C(1)

```
mydata.KMeansClusterIndices = KMeansClusterIndices;
I = cell(1,K);

for m = 1:1:K
    I{m} = find(KMeansClusterIndices==m);
    end;
plot_class_patterns(patterns,I,KMeansClusterIndices,'K Means Clustring
Results',right_list);
```

Εκτέλεση ISODATA

Τώρα εκτελούμε τον αλγόριθμο ISODATA και παρατηρούμε τις αλλαγές σε σύγκριση με τον k-means πρέπει να βρούμε τις τιμές των παραμέτρων που έχουν διαβαστεί από τα edit boxes με χρήση άλλης συνάρτησης και έχουν αποθηκευτεί στην μεταβλητή mydata.

```
numclus = mydata.numclus
sampar = mydata.samparmaxiter = mydata.maxiterstdv = mydata.stdv;
lump = mydata.lump
maxpair = mydata.maxpair;
[Rows,Columns] = size(patterns);
Patterns = patterns';
c = numclus;
Nc = sampar;
Iterations = maxiter;
L = maxpair;
Selta_n=2;
Selta_s=1;
Selta_D=8;
Classes=ISODATA(Patterns,c,Nc,Selta_n,Selta_s,Selta_D,L,Iterations);%
ClusterPatterns = cell(1,length(Classes));
IsodataClusterIndices = zeros(Rows,1);
n = 0;
Patterns = [];
for i = 1:1:length(Classes    ClusterPatterns{i} = getPatterns(Classes(i))';
x = ClusterPatterns{i};
    [Intersection,Indices] = intersect(patterns,x,'rows');
    IsodataClusterIndices(Indices) = I;
    Patterns = [Patterns;ClusterPatterns{i}];
    n = n + size(ClusterPatterns{i},1);
end;
mydata.IsodataClusterIndices = IsodataClusterIndices;
I = cell(1,length(Classes));
Nprev = 0;
for i = 1:1:length(Classes)
    Ncurr = size(ClusterPatterns{i},1);
    I{i} = [Nprev+1:1:Nprev+Ncurr];
    Nprev = Nprev + Ncurr;
end;
guidata(hObject,mydata);
update_table(hObject);

I2 = cell(1,length(Classes));
```

```
lc=length(Classes);
if (min(IsodataClusterIndices)==0 )
        IsodataClusterIndices2=IsodataClusterIndices+1;
        lc=length(Classes)+1
end

for m = 1:1:lc
    I2{m} = find(IsodataClusterIndices2==m);
end;
plot_class_patterns(patterns,I2,IsodataClusterIndices2,'ISODATA Clustering
Results',right_list)
```

Add Variable: Πρόκειται για ένα Puhsbutton αντικείμενο. Εδώ ο χρήστης πατώντας το κουμπί αυτό διαλέγει τις στήλες που θα χρησιμοποιήσει. Ο κώδικας που εκτελείται ανήκει στην συνάρτηση Clustering_tool3 και είναι ο εξής:

```
function pushbutton4_Callback(hObject, eventdata, handles)
mydata = guidata(hObject);
selected_index = get(mydata.LeftList,'Value'
s1 = get(mydata.LeftList,'String');
s2=size(s1,1);
```

Σημείωση: η μεταβλητή mydata περιέχει πληροφορίες που έχουν αποθηκευτεί από άλλες συναρτήσεις

```
selected_variables_num =0;
if(selected_variables_num <= s2)
    left_list = mydata.left_list;
    right_list = mydata.right_list;
    left_list_indices = mydata.left_list_indices;
    right_list_indices = mydata.right_list_indices;
    selected_variables_num = selected_variables_num + 1
[right_list,left_list] = UpdateLists(right_list,left_list,selected_index);
    [right_list_indices,left_list_indices] =
UpdateListsIndices(right_list_indices,left_list_indices,selected_index);
    mydata.left_list = left_list;
    mydata.right_list = right_list;
    mydata.right_list_indices = right_list_indices
    mydata.left_list_indices = left_list_indices
    mydata.selected_variables_num = selected_variables_num;
    set(mydata.LeftList,'String',left_list,'Value',1);
    set(mydata.RightList,'String',right_list,'Value',1);
    s1 = get(mydata.LeftList,'String');
    s2=size(s1,1);
    guidata(hObject,mydata);
end;
guidata(hObject,mydata);
```

Remove Variable: Πρόκειται για ένα Puhsbutton αντικείμενο. Εδώ ο χρήστης πατώντας το κουμπί αυτό σβήνει τις στήλες που επέλεξε. Ο κώδικας που εκτελείται ανήκει στην συνάρτηση Clustering_tool3 και είναι ο εξής:

```
function pushbutton5_Callback(hObject, eventdata, handles)
mydata = guidata(hObject);
```

```
selected_index = get(mydata.RightList,'Value');
selected_variables_num = size( get(mydata.RightList,'String'),1);
if(selected_variables_num > 0)
    left_list = mydata.left_list;
    right_list = mydata.right_list;
    left_list_indices = mydata.left_list_indices;
    right_list_indices = mydata.right_list_indices;
    [left_list,right_list] =
UpdateLists(left_list,right_list,selected_index);
    [left_list_indices,right_list_indices] =
UpdateListsIndices(left_list_indices,right_list_indices,selected_index);
    selected_variables_num = selected_variables_num - 1;
       mydata.right_list = right_list;
    mydata.left_list_indices = left_list_indices
    mydata.right_list_indices = right_list_indices
    mydata.selected_variables_num = selected_variables_num;
    set(mydata.LeftList,'String',left_list,'Value',1);
    set(mydata.RightList,'String',right_list,'Value',1);

    guidata(hObject,mydata);
end;
guidata(hObject,mydata);
```

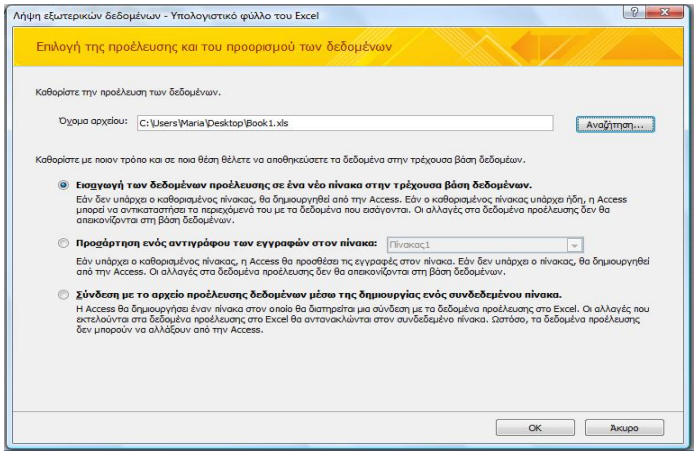
Το σπουδαιότερο στάδιο κατά το σχεδιασμό ενός αλγορίθμου μηχανικής μάθησης εκτός από αυτό που αφορά την μαθηματική διατύπωση του αλγορίθμου είναι ο έλεγχος της συμπεριφοράς του όταν τροφοδοτείται με πραγματικά δεδομένα. Μέσα στα πλαίσια αυτής της πτυχιακής εργασίας, στόχος είναι η συγκριτική μελέτη της απόδοσης δύο διαφορετικών αλγορίθμων συσταδοποίησης, του αλγορίθμου K – Means και του αλγορίθμου ISODATA. Η σύγκριση δύο αλγορίθμων μπορεί να γίνει με βάση την μαθηματική διατύπωσή τους προκειμένου να υπολογιστούν οι πολυπλοκότητες των αλγορίθμων οι οποίες αποτελούν και το βασικότερο κριτήριο αποτίμησης της υπολογιστικής ισχύος που απαιτεί ένας συγκεκριμένος αλγόριθμος. Ωστόσο, στην περίπτωση που έχουμε να κάνουμε με προβλήματα συσταδοποίησης πραγματικών δεδομένων για τα οποία δεν έχουμε κάποια εκ των προτέρων πληροφορία η συγκριτική αποτίμηση των δύο αλγορίθμων απαιτεί την εξέταση της συμπεριφοράς τους σε πραγματικά δεδομένα προκειμένου να προσδιοριστεί ο καταλληλότερος αλγόριθμος συσταδοποίησης για κάποιο συγκεκριμένο σύνολο δεδομένων. Θα πρέπει να σημειωθεί πως η συγκριτική μελέτη των δυο αλγορίθμων συσταδοποίησης μέσω της καταγραφής της απόδοσής τους σε πραγματικά δεδομένα δεν γίνεται να καταλήξει σε κάποιο τελικό συμπέρασμα όσο αφορά το ποιος είναι ο βέλτιστος αλγόριθμος. Η απάντηση στο ερώτημα που αφορά τον βέλτιστο αλγόριθμο δεν υπάρχει καθώς δεν υπάρχει κάποιος αλγόριθμος

συσταδοποίησης που τα αποτελέσματα που παρέχει να είναι καλύτερα σε σχέση με αυτά όλων των υπόλοιπων αλγορίθμων για κάθε δυνατό σύνολο προς ταξινόμηση δεδομένων. Το ερώτημα που μπορεί να απαντηθεί είναι το ποιος αλγόριθμος είναι καταλληλότερος για ένα συγκεκριμένο πρόβλημα συσταδοποίησης το οποίο καθορίζεται απόλυτα από τα προς ταξινόμηση δεδομένα.

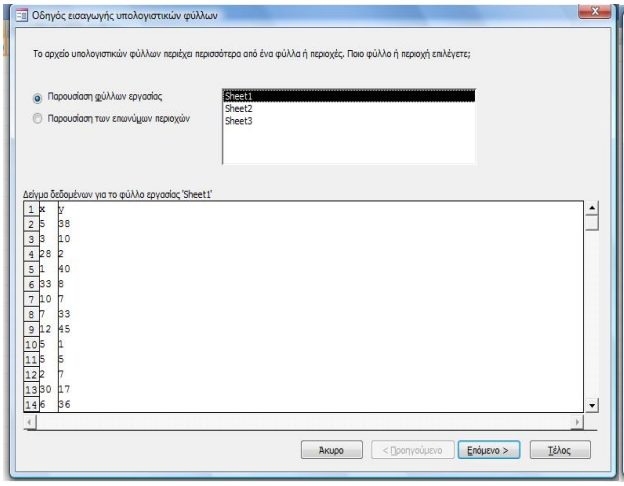
2.3 Σύνδεση με βάση δεδομένων

2.3.1 Δημιουργία Βάσης Access από αντίστοιχο αρχείο Excel

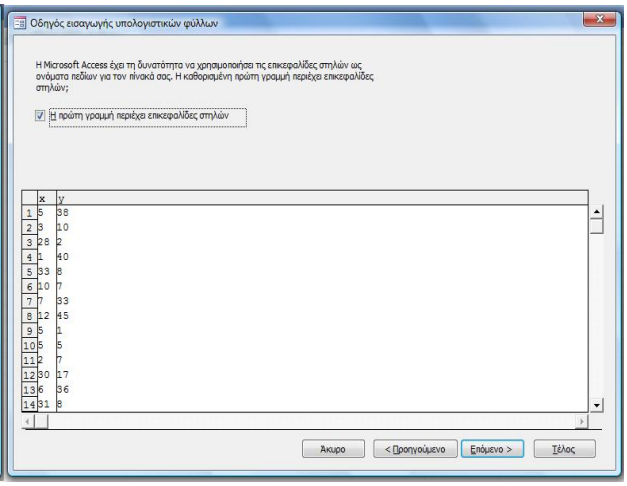
Για να δημιουργήσουμε μία νέα βάση δεδομένων πρώτα από όλα θα πρέπει να ανοίξουμε ένα νέο αρχείο της Access και εκεί να επιλέξουμε την δημιουργία μίας κενής βάσης δεδομένων. Στη συνέχεια από την καρτέλα Εξωτερικά δεδομένα επιλέγουμε το κουμπί Excel όπου θα εμφανιστεί και η παρακάτω εικόνα.



Εικόνα 12 Δημιουργία Βάσης Access από αντίστοιχο αρχείο Excel

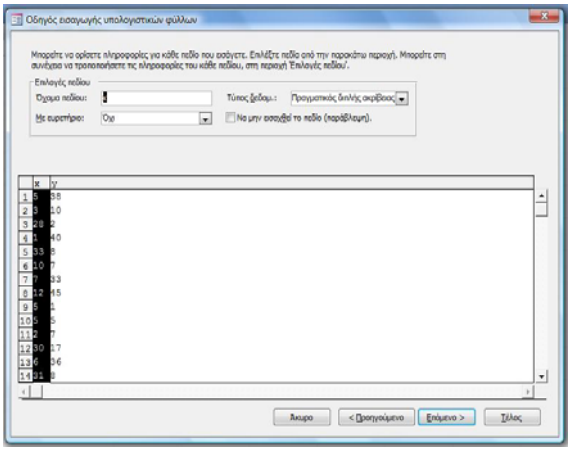


Εικόνα 13 Δήλωση υπολογιστικού φύλλου

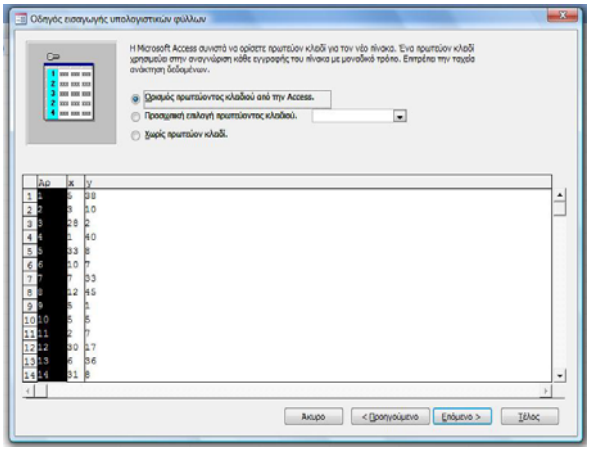


Εικόνα 14 Δήλωση της πρώτης γραμμής

Στις εικόνες 12 και 13 καλείται ο χρήστης να δηλώσει ποιο υπολογιστικό φύλλο επιθυμεί να ανοίξει και πατόντας το κουμπί επόμενο δηλώνουμε πως η πρώτη γραμμή από τα δεδομένα του πίνακα είναι τα ονόματα των στηλών .

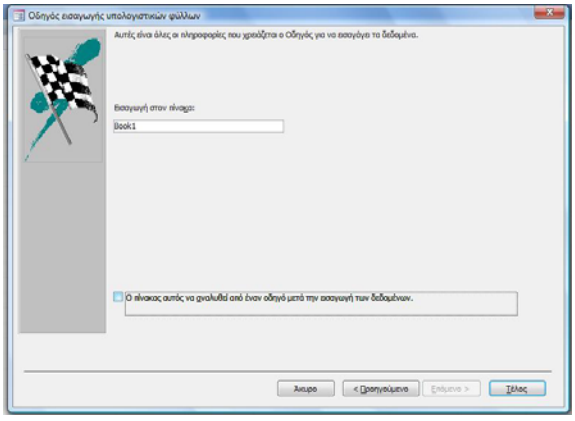


Εικόνα 15 Επιλογή των ονομάτων των στηλών

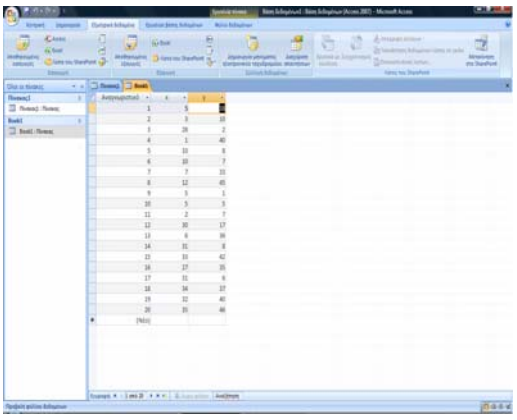


Εικόνα 16 Ορισμός πρωτεύοντος κλειδιού

Στις εικόνες 14 και 15 δηλώνουμε τα ονόματα των πεδίων και στη συνέχεια έχουμε την δυνατότητα να ορίσουμε εμείς πρωτεύον κλειδί, να μην επιλέξουμε κανένα ή να ορίσει κάποιο το πρόγραμμα από μόνο του. Στην περίπτωση μας επιλέγει για εμάς το πρόγραμμα. Για να ολοκληρωθεί η διαδικασία μένει μόνο να ορίσουμε το όνομα του πίνακα.



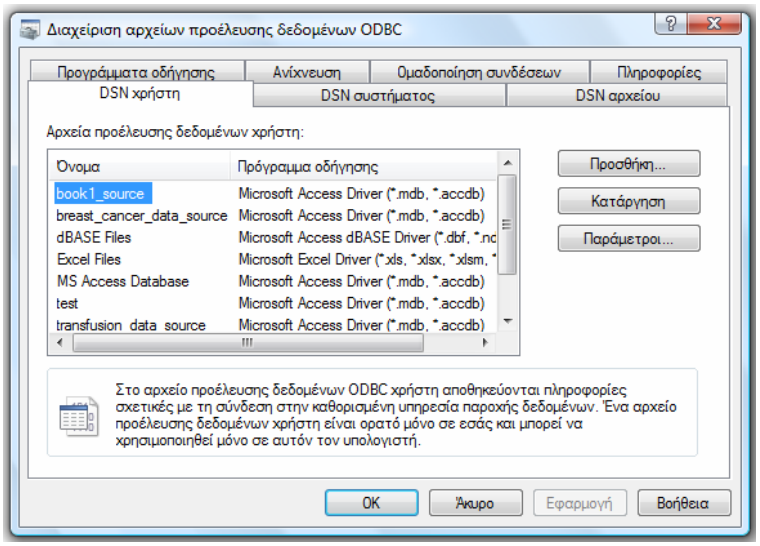
Εικόνα 17 Ορισμός ονόματος πίνακα



Εικόνα 18 Η βάση δεδομένων που δημιουργήσαμε

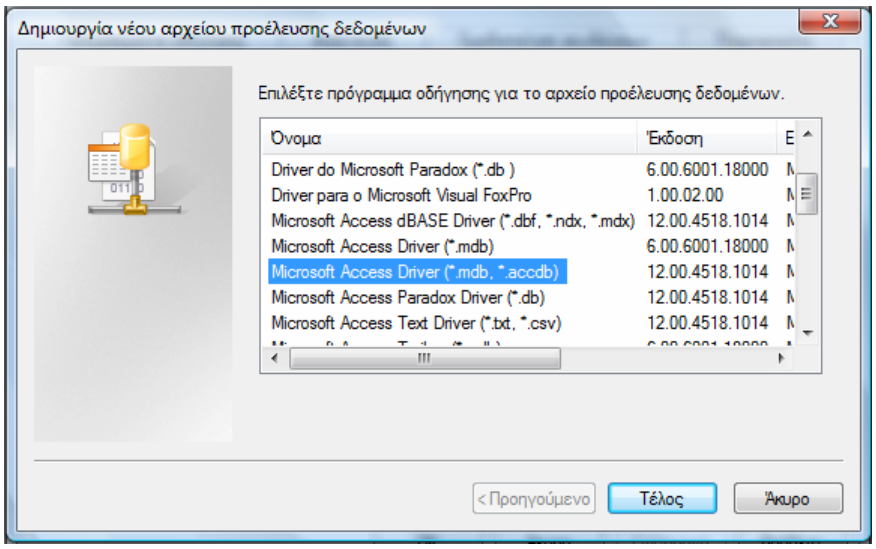
Η τελική αναπαράσταση των διαθέσιμων συλλογών δεδομένων γίνεται μέσω του περιβάλλοντος του ODBC Driver. Για να γίνει η σύνδεση των

βάσεων της Access με το Matlab θα πρέπει να πάμε στον Πίνακα Ελέγχου και να επιλέξουμε τα εργαλεία διαχείρισης, από εκεί στη συνέχεια επιλέγουμε το Διαχείριση αρχείων προέλευσης δεδομένων ODBC . Κάνοντας αυτή τη διαδικασία θα εμφανιστεί το ακόλουθο παράθυρο της Εικόνας 19.



Εικόνα 19 Αρχεία προέλευσης δεδομένων

Το επόμενο και τελευταίο βήμα είναι να διαλέξουμε το πρόγραμμα οδήγησης για το αρχείο προέλευσης δεδομένων, αυτό γίνεται πατώντας το κουμπί Προσθήκη. Κάνοντας αυτή τη διαδικασία θα εμφανιστεί το παράθυρο της εικόνας 20.



Εικόνα 20 Τύπος αρχείου προελεύσεως

Στο σημείο αυτό θα μπορούσαμε να πούμε ότι η εφαρμογή είναι

ευέλικτη όσο αναφορά στις βάσεις δεδομένων και αυτό γιατί με τις επιλογές της εικόνας 20 , μπορεί ο χρήστης να επιλέξει οποιοδήποτε ODBC Driver για να αποθηκεύσει τα δεδομένα.

Κεφάλαιο 3. Αποτελέσματα

Στη πτυχιακή αυτή επιλέξαμε σύνολα δεδομένων με βάση τα οποία θα πραγματοποιηθεί η μελέτη της συμπεριφοράς των δύο υπό εξέταση αλγορίθμων. Η πηγή των δεδομένων που χρησιμοποιήθηκε ήταν το UCI Machine Learning Repository(12) ένας δικτυακός τόπος που παρέχει μια μεγάλη ποικιλία από συλλογές δεδομένων κατάλληλες για προβλήματα μηχανικής μάθησης.

Στις παραγράφους που ακολουθούν θα περιγραφούν διεξοδικά τα σύνολα δεδομένων που χρησιμοποιήθηκαν μέσα στα πλαίσια αυτής της πτυχιακής εργασίας. Πρόκειται για σύνολα δεδομένων που ο συγκεκριμένος δικτυακός τόπος παρέχει αποκλειστικά για προβλήματα συσταδοποίσησης. Τα σύνολα δεδομένων αυτού του τύπου αποτελούνται από εγγραφές οι οποίες είναι οργανωμένες σε πίνακες. Οι στήλες του πίνακα αυτού αποτελούν τα επιμέρους χαρακτηριστικά για κάθε ένα από τα αντικείμενα που συνιστούν το σύνολο των προς συσταδοποίηση δεδομένων. Με άλλα λόγια η κάθε στήλη του πίνακα των δεδομένων αποτελεί την μαθηματική περιγραφή ενός σημείου σε κάποιο πολυδιάστατο διανυσματικό χώρο. Η αναπαράσταση αυτών των σημείων σε χώρους μικρότερης διάστασης όπως είναι το καρτεσιανό επίπεδο πραγματοποιείται με χρήση της τεχνικής της Ανάλυσης Κύριων Συνιστωσών (Principal Components Analysis).

3.1 Πραγματικά Δεδομένα (Wine Data Set) (12)

Η συλλογή δεδομένων περιλαμβάνει τα αποτελέσματα της χημικής ανάλυσης κρασιών τα οποία καλλιεργούνται στην ίδια περιοχή της Ιταλίας αλλά προέρχονται από 3 διαφορετικές καλλιέργειες. Η χημική ανάλυση προσδιόρισε 13 διαφορετικά χημικά συστατικά τα οποία αποτελούν τα

χαρακτηριστικά της συγκεκριμένης συλλογής δεδομένων, η γενική περιγραφή της οποίας παρουσιάζεται στον παρακάτω πίνακα:

Wine Data Set	
Χαρακτηριστικά Συνόλου Δεδομένων	**Πολυμετάβλητα**
Χαρακτηριστικά Γνωρισμάτων	**Πραγματικές Τιμές**
Σχετιζόμενο Πρόβλημα Μηχανικής Μάθησης	**Συσταδοποίηση**
Αριθμός Εγγραφών	**178**
Αριθμός Γνωρισμάτων	**13**
Ελλιπής Τιμές	**Όχι**

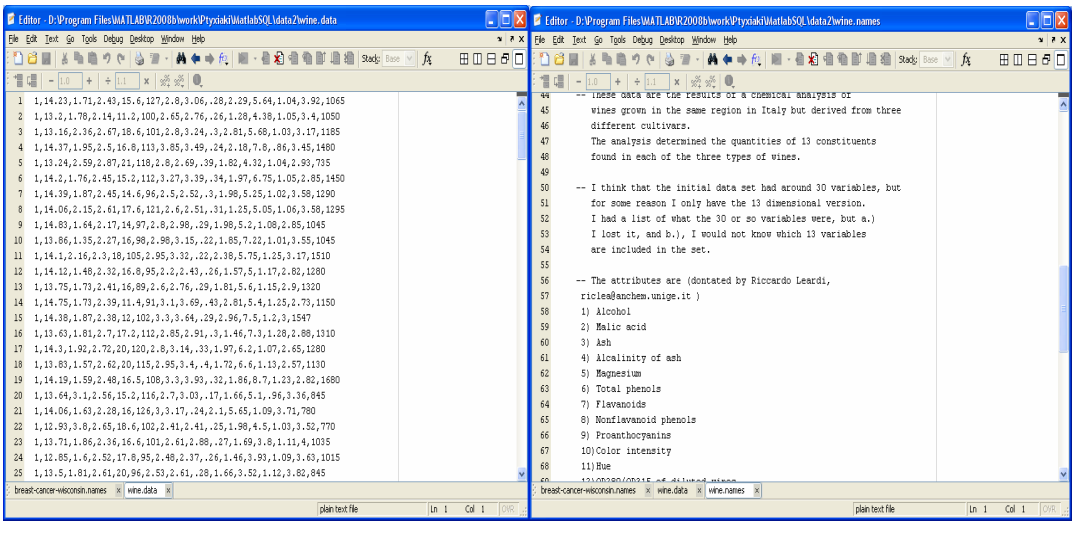
Κάθε εγγραφή της συγκεκριμένης συλλογής δεδομένων αποτελείται από τα 13 παρακάτω χαρακτηριστικά που αφορούν την χημική σύσταση μιας δοσμένης καλλιέργειας κρασιού και οι τιμές τους είναι πραγματικές:

1. Αλκοόλη: Αναφέρεται στην περιεκτικότητα αλκοόλης της συγκεκριμένης ποικιλίας κρασιού.
2. Μηλικό Οξύ: Αναφέρεται στην περιεκτικότητα μηλικού οξέος της συγκεκριμένης ποικιλίας κρασιού.
3. Τέφρα: Αναφέρεται στην περιεκτικότητα τέφρας της συγκεκριμένης ποικιλίας κρασιού.
4. Αλκαλικότητα της τέφρας
5. Μαγνήσιο: Αναφέρεται στην περιεκτικότητα μαγνησίου της συγκεκριμένης ποικιλίας κρασιού.
6. Φαινόλες: Αναφέρεται στην συνολική περιεκτικότητα φαινόλων της συγκεκριμένης ποικιλίας κρασιού.
7. Flavanoids: Αντιοξειδωτικοί παράγοντες.
8. Οξειδωτικές Φαινόλες
9. Προανθοκυάνες
10. Χρωματική Ένταση
11. Χρωματικός Τόνος

12. Προλίνη: Αναφέρεται στην περιεκτικότητα του αμινοξέως προλίνη της συγκεκριμένης ποικιλίας κρασιού.

3.2 Μετατροπή Δεδομένων

Η συλλογή των δεδομένων που χρησιμοποιήθηκε μέσα στα πλαίσια αυτής της πτυχιακής εργασίας ήταν αναγκαίο να υποστεί μια διαδικασία μετατροπής προκειμένου να οργανωθεί σε μία βάση δεδομένων τόσο της Access όσο και της SQL. Η διαδικασία αυτή της μετατροπής είναι αναγκαία προκειμένου να μετασχηματιστούν τα δεδομένα σε μορφή κατάλληλη η οποία να επιτρέπει την σύνδεσή της με το προγραμματιστικό περιβάλλον του MatLab και την ακόλουθη εφαρμογή των υπό εξέταση αλγορίθμων συσταδοποίησης. Ωστόσο, η αρχική μορφή των δεδομένων όπως αυτά παρέχονται από τον δικτυακό τόπο του UCI Machine Learning Repository (12) επιβάλει ένα αρκετά πολύπλοκο σχήμα μετατροπής των δεδομένων μεταξύ πολλών και διαφορετικών τρόπων αναπαράστασής τους. Η αναπαράσταση αυτή χρησιμοποιείται από τον δικτυακό τόπο του UCI Machine Learning Repository και είναι δεσμευτική για το σχήμα μετασχηματισμού των δεδομένων που θα ακολουθήσει. Η συλλογή δεδομένων που χρησιμοποιήθηκε στην παρούσα πτυχιακή εργασία αντιστοιχεί σε δύο αρχεία κειμένου, όπως φαίνεται και στην εικόνα 20, όπου το πρώτο περιέχει τις εγγραφές της συλλογής δεδομένων και το δεύτερο τα χαρακτηριστικά γνωρίσματα που αντιστοιχούν στο σύνολο των εγγραφών. Η αρχική μορφή των δεδομένων είναι η ακόλουθη:



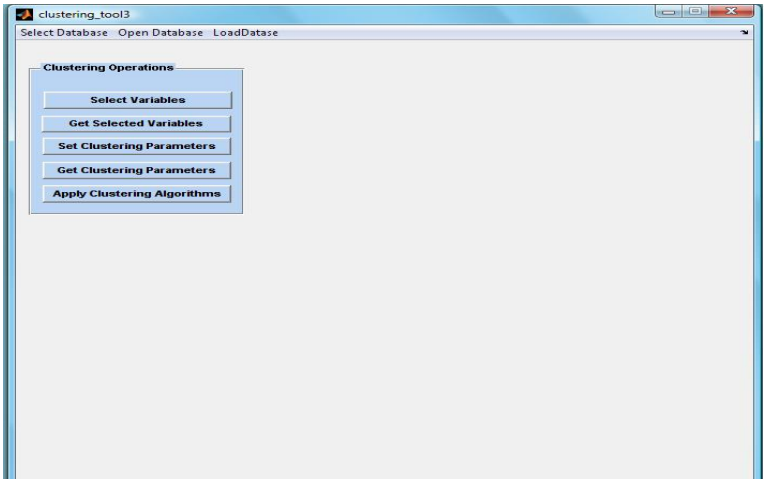
Εικόνα 21 Αρχείο με τα δεδομένα και τα ονόματα της βάσης

- wine.data: Σε κάθε γραμμή του αρχείου και μια διαφορετική εγγραφή.
- wine.names: Συνοδευτικό αρχείο περιγραφής της συλλογής δεδομένων όπου μεταξύ άλλων αναφέρει και τα χαρακτηριστικά γνωρίσματα των εγγραφών, τα ονόματα των στηλών δηλαδή στον πίνακα της βάσης που θα προκύψει στο τέλος.

  Η μετατροπή των αρχικών αρχείων κειμένων σε αρχεία .mat πραγματοποιείται μέσω του προγραμματιστικού περιβάλλοντος του MatLab το οποίο παρέχει την δυνατότητα ανάγνωσης των αρχείων .data και την μετατροπή τους σε αρχεία .mat. Η συγκεκριμένη διαδικασία γίνεται μέσω της λειτουργίας import.

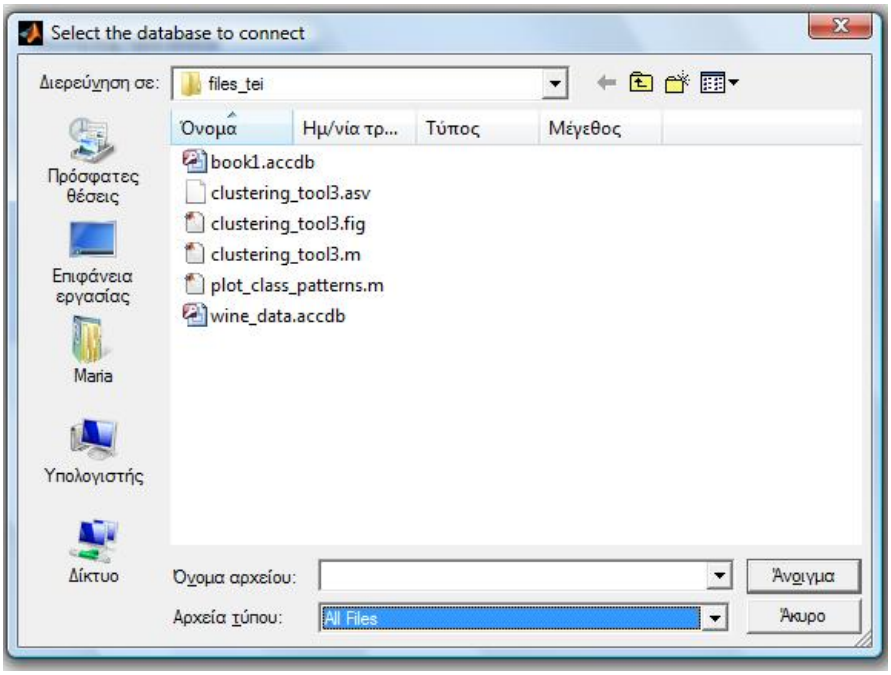
3.3 Εκτέλεση Εφαρμογής

Αφότου εκτελέσουμε την εφαρμογή θα έχουμε το ακόλουθο παράθυρο.



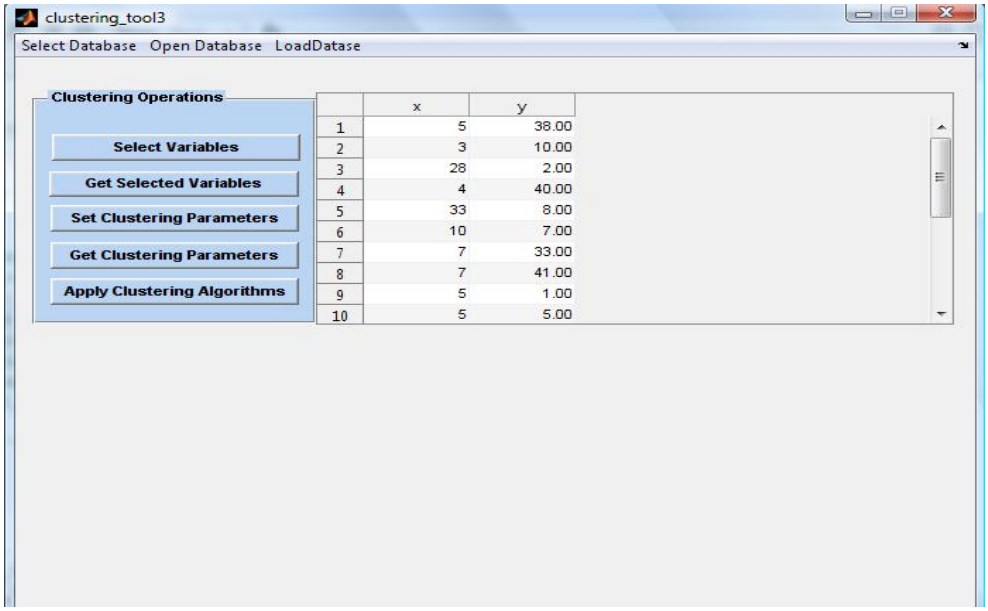
Εικόνα 22 Η αρχική μορφή της εφαρμογής

Στην εφαρμογή δεν έχει φορτωθεί κάποια βάση και επομένως δεν υπάρχουν δεδομένα προς επεξεργασία. Το πρώτο πράγμα που πρέπει να κάνουμε είναι να πιέσουμε το κουμπί Select Database που θα ανοίξει την αντίστοιχη εφαρμογή ώστε να φορτώσουμε δεδομένα από μία βάση της επιλογής μας .



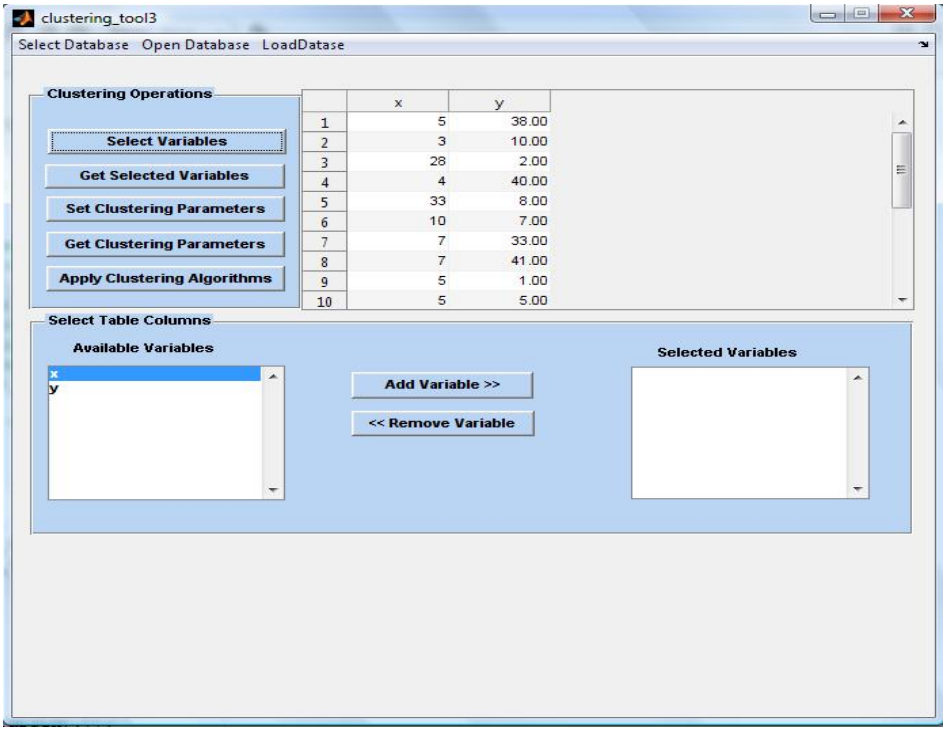
Εικόνα 23 Αναζήτηση μίας βάσης δεδομένων

Στη συνέχεια επιλέγουμε μία βάση δεδομένων από τις ήδη υπάρχουσες και πιέζουμε άνοιγμα. Αφού γίνει αυτή η διαδικασία θα πρέπει να πιέσουμε το κουμπί Open Database όπου πλέον η εφαρμογή μας θα περιέχει τα δεδομένα της βάσης που επιλέξαμε.



Εικόνα 24 Η εφαρμογή με την βάση δεδομένων που επιλέξαμε

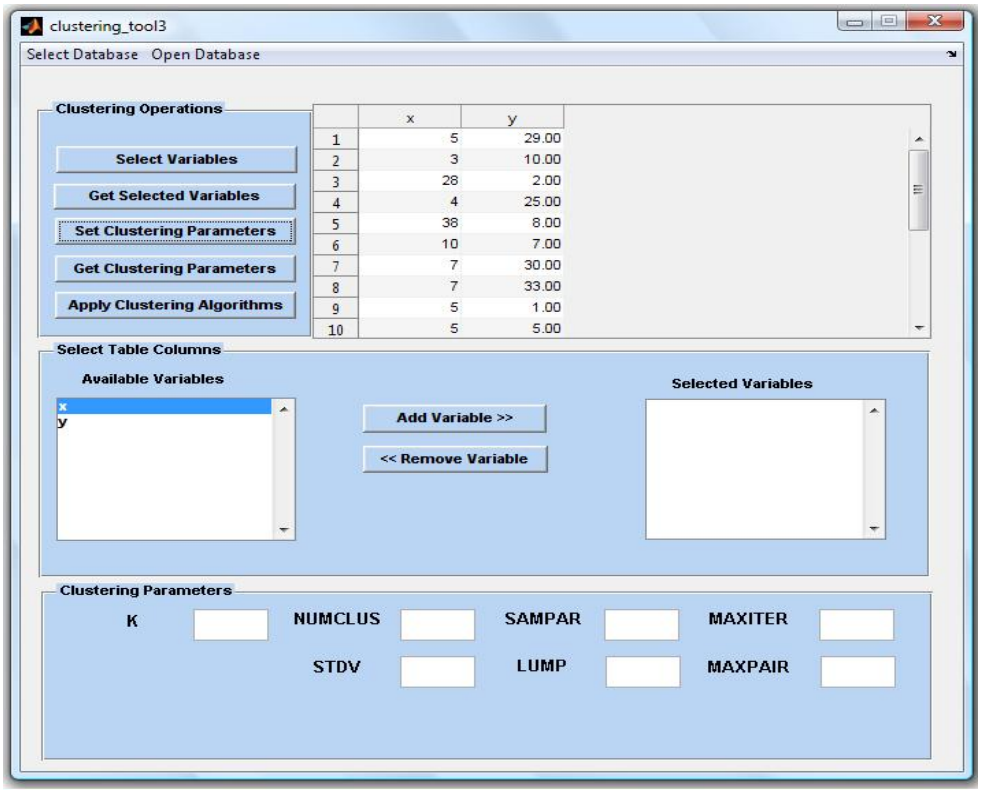
Το επόμενο βήμα που πρέπει να κάνει ο χρήστης είναι να πιέσει το κουμπί Select Variables έτσι ώστε να εμφανιστούν οι διαθέσιμες στήλες τις οποίες θα χρησιμοποιήσει στην εφαρμογή. Πιέζοντας το κουμπί Add Variable επιλέγει τις μεταβλητές που θέλει να χρησιμοποιήσει.



Εικόνα 25 Εισαγωγή στηλών

Στη συνέχεια ο χρήστης θα πρέπει να εισάγει τις μεταβλητές που θα χρειαστούν ώστε να λειτουργήσουν οι δύο αλγόριθμοι(10) , δηλαδή τον αριθμό των κλάσεων για τον αλγόριθμο k-means (K), έναν προτεινόμενο

αριθμό κλάσεων για τον αλγόριθμο Isodata (NUMCLUS), τον ελάχιστο αριθμό δεδομένων που μπορεί να περιλαμβάνει μία ομάδα (SAMPAR), τον μέγιστο αριθμό επαναλήψεων του αλγορίθμου (MAXITER), τη μέγιστη τυπική απόκλιση των σημείων από το κέντρο βάρους της ομάδας κατά μήκος κάθε άξονα (STDV), την ελάχιστη απαιτούμενη απόσταση μεταξύ των κέντρων βάρους δύο ομάδων (LUMP), ο μέγιστος αριθμός από ζεύγη ομάδων τα οποία μπορούν να συγχωνευθούν σε μία επανάληψη (MAXPAIR). Αφού εισάγει ο χρήστης τις κατάλληλες μεταβλητές, πρέπει να πιέσει το κουμπί Get Clustering Parameters έτσι ώστε να αποθηκεύσει ο αλγόριθμος τις μεταβλητές που χρειάζεται.



Εικόνα 26 Εισαγωγή κατάλληλων μεταβλητών

Αφού γίνουν όλες οι παραπάνω ενέργειες ο χρήστης καλείται να πατήσει το κουμπί Apply Clustering Algorithms για να εμφανιστούν τα αποτελέσματα στην οθόνη του.

3.4 Αποτελέσματα από συνθετικά δεδομένα

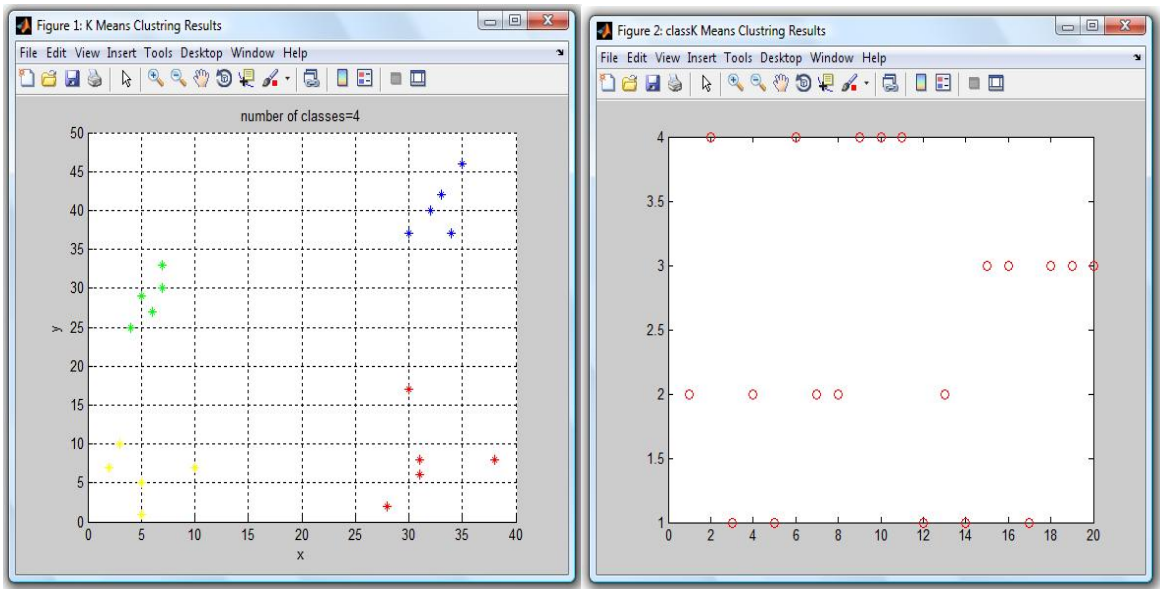
Για τις ανάγκες της πτυχιακής δημιουργήσαμε μία βάση δεδομένων Book1 με τυχαία στοιχεία έτσι ώστε να μπορέσουμε να δούμε την λειτουργία των δύο

αλγορίθμων. Η βάση αυτή περιέχει τα δεδομένα που φαίνονται στην εικόνα 26.

x	y
5	38
3	10
28	2
1	40
33	8
10	7
7	33
12	45
5	1
5	5
2	7
30	17
6	36
31	8
33	42
27	35
31	6
34	37
32	40
35	46

Εικόνα 27 Στοιχεία συνθετικής βάσης δεδομένων

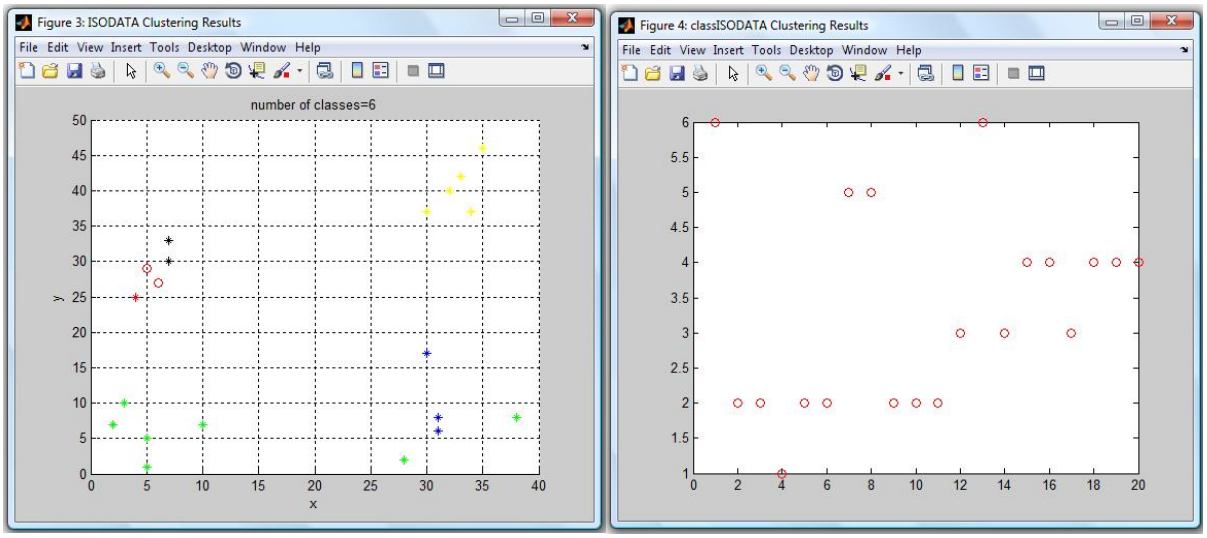
Εισάγουμε τα δεδομένα αυτά στην εφαρμογή μας όπως περιγράψαμε στην προηγούμενη ενότητα, στο συγκεκριμένο παράδειγμα ζητήσαμε από το αλγόριθμο K-means να δημιουργήσει τέσσερις κλάσεις. Τα αποτελέσματα της εφαρμογής όσο αναφορά στον K-means φαίνονται στην εικόνα 27.



Εικόνα 28 Γραφικά αποτελέσματα του αλγορίθμου k-means

Στο πρώτο γράφημα βλέπουμε τον αριθμό των κλάσεων με διαφορετικό χρώμα η κάθε μία ενώ στο δεύτερο γράφημα ο χρήστης μπορεί να δει σε ποιά κλάση ανήκει κάθε ζευγάρι δεδομένων, για παράδειγμα το πρώτο ζευγάρι ανήκει στην δεύτερη κλάση, το δεύτερο στην τέταρτη κλάση κοκ.

Αντίθετα από τον αλγόριθμο isodata ζητήσαμε να δημιουργήσει τέσσερις κλάσεις (NUMCLUS) με τουλάχιστον δύο στοιχεία στην κάθε μια από αυτές (SAMPAR),να επαναληφθεί δέκα φορές (MAXITER),η μέγιστη τυπική απόκλιση (STDV)των σημείων από το κέντρο βάρους της ομάδας κατά μήκος κάθε άξονα να είναι περίπου το 10% της ελάχιστη απαιτούμενης απόστασης μεταξύ των κέντρων βάρους δύο ομάδων στην οποία ζητήσαμε να είναι ένα (LUMP), τέλος ο μέγιστος αριθμός από ζεύγη ομάδων τα οποία μπορούν να συμπεριληφθούν σε μία επανάληψη ζητήσαμε να είναι δύο (MAXPAIR).Τα αποτελέσματα φαίνονται στην εικόνα 28 που ακολουθεί.



Εικόνα 29 Γραφικά αποτελέσματα του αλγορίθμου Isodata

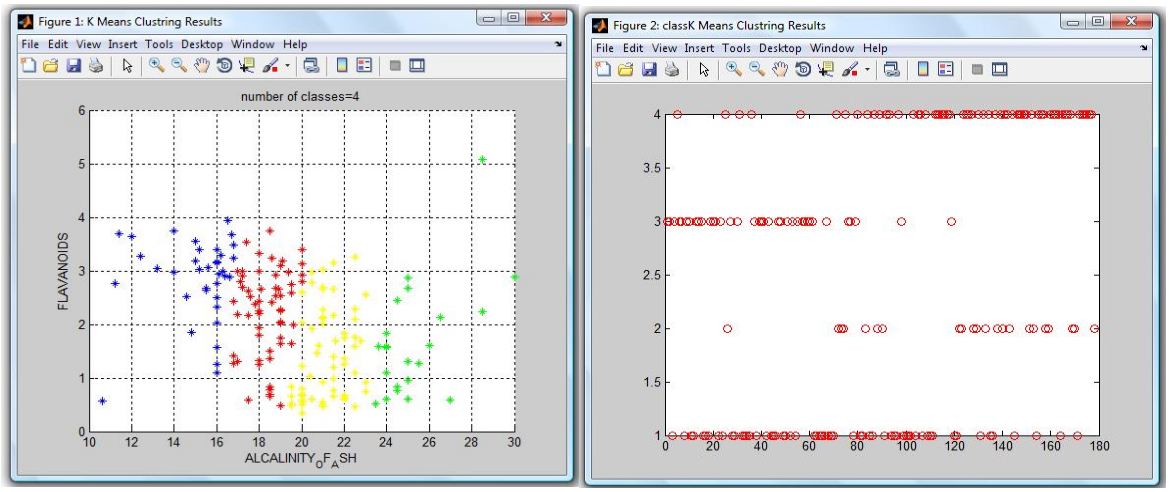
Στα παραπάνω γραφήματα βλέπουμε πως για τις συγκεκριμένες μεταβλητές που δώσαμε στον αλγόριθμο, ο isodata χώρισε τα δεδομένα σε έξι κλάσεις .

3.5 Αποτελέσματα από πραγματικά δεδομένα

Στα πραγματικά δεδομένα οι τιμές είναι δεδομένα που πήραμε δικτυακό τόπο του UCI Machine Learning Repository και αναφέρονται σε συστατικά κάποιας ποικιλίας κρασιού. Εφαρμόζοντας τους δύο αλγορίθμους έχουμε τα αποτελέσματα που φαίνονται στις παρακάτω εικόνες.

Εισάγουμε τα δεδομένα αυτά στην εφαρμογή μας όπως περιγράψαμε στην προηγούμενη ενότητα, στο συγκεκριμένο παράδειγμα ζητήσαμε από το αλγόριθμο K-means να δημιουργήσει τέσσερις κλάσεις και επιλέξαμε τυχαία

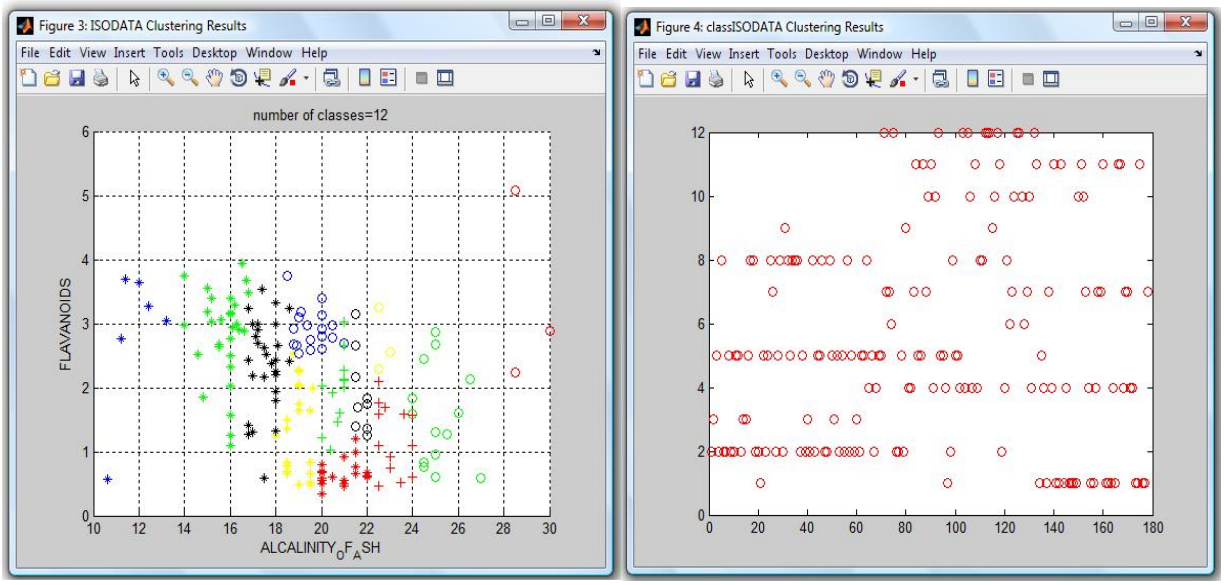
δύο από τις 13 μεταβλητές την ALCANICITY OF ASH και την FLAVANOIDS. Τα αποτελέσματα της εφαρμογής όσο αναφορά στον K-means φαίνονται στην εικόνα 29.



Εικόνα 30 Γραφικά αποτελέσματα για τον αλγόριθμο k-means

Εδώ βλέπουμε πως ο αλγόριθμος k-means χώρισε τα δεδομένα μας σε αυστηρά τέσσερις κλάσεις.

Αντίθετα από τον αλγόριθμο isodata ζητήσαμε να δημιουργήσει τέσσερις κλάσεις (NUMCLUS) με τουλάχιστον τρία στοιχεία στην κάθε μια από αυτές (SAMPAR),να επαναληφθεί δέκα φορές (MAXITER),η μέγιστη τυπική απόκλιση (STDV)των σημείων από το κέντρο βάρους της ομάδας κατά μήκος κάθε άξονα να είναι περίπου το 10% της ελάχιστη απαιτούμενης απόστασης μεταξύ των κέντρων βάρους δύο ομάδων στην οποία ζητήσαμε να είναι πέντε (LUMP), τέλος ο μέγιστος αριθμός από ζεύγη ομάδων τα οποία μπορούν να συμπεριληφθούν σε μία επανάληψη ζητήσαμε να είναι δύο (MAXPAIR).Τα αποτελέσματα φαίνονται στην εικόνα 31 που ακολουθεί.



Εικόνα 31 Γραφικά αποτελέσματα του αλγορίθμου isodata

Στα παραπάνω γραφήματα βλέπουμε πως για τις συγκεκριμένες μεταβλητές που δώσαμε στον αλγόριθμο, ο isodata χώρισε τα δεδομένα σε δώδεκα κλάσεις.

4. Συμπεράσματα

Με την εφαρμογή που υλοποιήθηκε στα πλαίσια αυτής της πτυχιακής εργασίας, δίνεται η δυνατότητα στο χρήστη να κατανοήσει κατά κύριο λόγο τις μεθόδους συσταδοποίησης και τα είδη των αλγορίθμων. Η εφαρμογή αυτή μπορεί να προβάλει γραφικά την συσταδοποίηση των δυο αλγορίθμων έτσι ώστε να μπορεί ο χρήστης να τους συγκρίνει. Η δοκιμή των αλγορίθμων συσταδοποίησης που χρησιμοποιήθηκαν μας έδειξε ότι ο isodata είναι πιο ευέλικτος σε σχέση με τον k-means καθώς δεν χρειάζεται να ξέρουμε εκ των προτέρων τον αριθμό των κλάσεων που θα πρέπει να δημιουργηθούν απλά να του δώσουμε έναν ενδεικτικό αριθμό και από εκεί και ύστερα ο αλγόριθμος θα κρίνει ποιος είναι ο καταλληλότερος αριθμός κλάσεων, σε αντίθεση με τον k-means ο οποίος θα δημιουργήσει αυστηρά όσες κλάσεις του ζητήσει χρήστης. Επίσης Ένα ακόμα αρνητικό στοιχείο του ISODATA είναι και το γεγονός ότι είναι εμφανώς πιο αργός από τον k-means. Συμπεραίνοντας θα μπορούσαμε να πούμε ότι ο ISODATA είναι πολύ χρήσιμος σε προβλήματα που δεν γνωρίζουμε εκ των προτέρων τον αριθμό

των κλάσεων αλλά είναι δύσκολος στην χρήση του καθώς χρειάζεται πολύ μεγάλη προσοχή στην επιλογή των μεταβλητών εισόδου προκειμένου να βγει το σωστό αποτέλεσμα. Κλείνοντας θα πρέπει να επισημάνουμε για ακόμη μια φορά πως η παρούσα εφαρμογή κατασκευάστηκε να λειτουργεί για έναν δυσδιάστατο διανυσματικό χώρο, σε περίπτωση που οι διαστάσεις είναι περισσότερες θα πρέπει να εφαρμοστεί η Ανάλυση Κυρίων Συνιστωσών (PCA- Principal Components Analysis) κάτι που ξεφεύγει από τα πλαίσια της παρούσας πτυχιακής.

Παράρτημα Α

Ο αλγόριθμος ISODATA

```
function Classes=ISODATA(Patterns,c,Nc,Selta_n,Selta_s,Selta_D,L,I)


%Μεταβλητές εισόδου:
%     Patterns:πίνακας 2 διαστάσεων που περιέχει τα προς ταξινόμιση
δεδομένα. % Η κάθε στήλη αντιστοιχεί σε ένα δεδομένο(πρότυπο). Η κάθε γραμμή
%     αντιστοιχεί στις τιμές συγκεκριμένης μεταβλητής σε όλα υα πρότυπα.
%    Nc: ο αρχικός αριθμός των κλάσεων.
%    c : ο αρχικός αριθμός των κλάσεων.
%    L: ο μέγιστος αριθμός απο ζεύγη ομάδων τα οποία μπορούν να συγχωνευθούν
%     σε μια επανάληψη.
%    I:  ο μέγιστος αριθμός επαναλήψεων του αλγορίθμου.
%Μεταβλητές εξόδου:
%    Classes: οι ομάδες όπως προέκυψαν απο τον isodata

Dim=size(Patterns,1); %ο αριθμός των μεταβλητών
N=size(Patterns,2);   %ο αριθμός των προτύπων που έχουμε προς ταξινόμηση.
Partition=0; % flag , αρχικοποιείται με 0 και αλλάζει σε 1 αν χρειαστεί.
bCombine=0;   % flag , αρχικοποιείται με 0 και αλλάζει σε 1 αν χρειαστεί.

Classes=[];

% ΒΗΜΑ 1 : κατασκευάζουμε στην αρχή τόσες κλάσεις όσες είναι ο αρχικός
%αριθμός NUMCLUS.Επιλέγουμε Nc τυχαία σημεία από το αρχικό σύνολο των
% δεδομένων που θα αποτελέσουν τα  κέντρα βάρους των αρχικών ομάδων.
for iPattern=1:Nc
    p=Classf(Patterns(:,iPattern));  % κατασκευή της κλάσης p.
%Προσοχή: δίνουμε ως κέντρο βάρους της ομάδας την τιμή του iPattern
    Classes=[Classes,p];             % ενσωμάτωση της κλάσης p στην
μεταβλητή %Classes,που έχει αποθηκευμένες  όλες τις προηγούμενες κλάσεις
end
%Η μεταβλητή classes έχει πλέον πολλές στήλες. Κάθε στήλη περιέχει και
% μια μεταβλητή που απεικονίζει μια κλάση 'Classf'
% οπότε η εντολή c = Classes(2) θα  επιστρέψει στην μεταβλητή c μία δομή
% δεδομένων που θα περιέχει το κέντρο βάρους(c.CenterPattern)και τα
% δεδομένα (c.Patterns) της δεύτερης κλάσης.
iStep=0;
while iStep<I % δεν πρέπει να κάνουμε περισσότερες επαναλήψεις από ότι
%προσδιορίζει η μεταβλητή  I (max iterations)
    %*************************************************************


    for iClass=1:Nc
        Classes(iClass)=empty(Classes(iClass)); % αδειάζουμε τις κλάσεις από
%δεδομένα.
    end

    % ΒΗΜΑ 2 : το κάθε σημείο του αρχικού συνόλου ανατίθεται στην ομάδα
    % που  αντιπροσωπεύεται από το πλησιέστερο προς αυτό κέντρο
    % Η παρακάτω for εντολή βρίσκει για κάθε ένα πρότυπο την ομάδα που
    % έχει το πλησιέστερο προς αυτό κέντρο.

    for iPattern=1:N % επανάληψη για όλα τα πρότυπα που έχουμε διαθέσιμα.
        dMin=getDistance(Classes(1),Patterns(:,iPattern)); %βρες την
απόσταση %της πρώτης κλάσης από το  συγκεκριμένο πρότυπο
 % και θεώρησε ότι αυτή είναι  η ελάχιστη δυνατή.
```

```
        iClassMin=1; %Η μεταβλητή αυτή περιέχει την κλάση(1,2,...) το κέντρο
%βάρος της οποίας έχει την ελάχιστη απόσταση με το συγκεκριμένο πρότυπο  που
%εξετάζουμε.
% στην αρχή έχουμε θεωρήσει οτι η κλάση 1 είναι η 'πλησιέστερη'
        % για κάθε μια από τις υπόλοιπες διαθέσιμες κλάσεις, βρες την
        % απόσταση του συγκεκριμένου προτύπου από το κέντρο βάρους των
        % κλάσεων .Αν η απόσταση είναι μικρότερη από αυτήν που μέχρι τώρα
        % θεωρούμε ελάχιστη, τότε θα πρέπει να αποθηκεύσουμε την νέα
        % ελάχιστη απόσταση και την νέα κλάση που μας δίνει την dMin
        for iClass=2:length(Classes)
            d=getDistance(Classes(iClass),Patterns(:,iPattern));
%υπολογίζουμε την απόσταση του iPattern και του κέντρου βάρους της iClass
            if dMin>d  %αν βρεθεί μικρότερη από αυτή που μέχρι τώρα θεωρούμε
%dMin
                dMin=d;  %αποθήκευσε ως dMin την νέα απόσταση.
                iClassMin=iClass; %αποθήκευσε και την κλάση που μας δίνει
την %dMin
            end
        end

Classes(iClassMin)=addPattern(Classes(iClassMin),Patterns(:,iPattern)); %θα
πρέπει στο τέλος να προσθέσουμε το iPattern στην ομάδα(κλάση-cluster)
iClassMin, που μας έδωσε την ελάχιστη απόσταση από το κέντρο βάρους της.
    end
    %*************************************************************

    %*************************************************************
    % ΒΗΜΑ 3 :Εκείνα τα κέντρα βάρους για τα οποία ο αριθμός των
πλησιέστερων
    % προς αυτών σημείων είναι λιγότερος από SAMPAR πρέπει να διαγραφούν.
    for iClass=1:length(Classes)
        if getSize(Classes(iClass))<Selta_n
            Classes(iClass)=[];
            bCombine=1;
            break;
        end
    end

    if bCombine==1
        Nc=Nc-1;
        break;
    end
    %*************************************************************


    d=[];
    d_mean=0.0;

    % ΒΗΜΑ 4 :Τα κέντρα βάρους των θεωρούμενων ομάδων πρέπει να
    % μετακινηθούν προς το κέντρο της κλάσης.
    % Το συγκεκριμένο βήμα εκτελείται αυτόματα, αφού η getCenter μας
    % επιστρέφει τον μέσο όρο  των δεδομένων που περιέχει μια κλάση.

    % ΒΗΜΑ 5 :Υπολογίζεται η παράμετρος Dj που αντιστοιχεί στην μέση
    % απόσταση των σημείων του συνόλου Sj από το κέντρο ομάδας Zj.
    % Υπολογίζεται η παράμετρος D που αντιστοιχεί στην μέση τιμή όλων των
    % προηγούμενων αποστάσεων
    for iClass=1:Nc  % Για κάθε μία κλάση υπολόγισε το Dj
        d_tmp=getMeanDistance(Classes(iClass));% Βρες τον μέσο όρο των
αποστάσεων όλων των δεδομένων της κλάσης iClass από το κέντρο βάρους της.
        d=[d,d_tmp]; % πρόσθεσε τον μέσο όρο στο διάνυσμα d, που έχει τους
μέσους όρους των προηγούμενων κλάσεων.
```

```
        d_mean=d_mean+d_tmp*getSize(Classes(iClass)); % Afto einai to D
    end
    d_mean=d_mean/N;  % η τελική τιμή του D
    %*********************************************************

    % ΒΗΜΑ 6 : Αν είμαστε στην τελευταία επανάληψη του αλγορίθμου, τότε
    % πάμε στο Βήμα 9.Επιπλέον ,αν 2k>NUMCLUS και πρόκειται για άρτιο
    % αριθμό επανάληψης ή k>=2NUMCLUS,τότε πάμε στο Βήμα 9
    %*********************************************************
    % Η παρακάτω if εντολή ελέγχει αν ισχύουν οι παραπάνω συνθήκες
    if (iStep~=I)&&(Nc<=c/2||(Nc>c/2&&Nc<2*c&&(mod(iStep,2)==0)))
        %*****************************************************
        for iClass=1:Nc
          % ΒΗΜΑ 7 : Για κάθε ομάδα Sj υπολογίζεται ένα νέο διάνυσμα vj του
          %οποίου i i-οστή συντεταγμένη αντιστοιχεί στην τυπική απόκλιση των
          % i-οστών συντεταγμένων διανυσμάτων τα οποία κατευθύνονται από
          % το εκάστοτε κέντρο βάρους zj προς κάθε ένα από τα σημεία του
            % συνόλου Sj
            % Το Βήμα 7 εκτελείται από την εντολή getStdVector
            delta=getStdVector(Classes(iClass));

            [delta_max,index]=max(delta); % Βρες την μέγιστη τιμή

            % ΒΗΜΑ 8 : Για κάθε ομάδα Sj,στην περίπτωση που έχουμε
            % vj,max>STDV και ισχύει κάτι από τα παρακάτω
            % ( (Dj>D) και (nj>2(nmin+1)) ) ή k<=kinit/2
            % τότε ο αριθμός των θεωρούμενων κέντρων βάρους θα πρέπει να
            % αυξηθεί και το σύνολο Sj να διασπαστεί σε δυο ομάδες. Τότε,
            % το κέντρο βάρους της ομάδας Sj θα πρέπει να αντικατασταθεί
            % από δύο σημεία που θα βρίσκονται στην γειτονία του zj όπου το
            % μέτρο και η κατεύθυνση του διανύσματος θα εξαρτάται από
            % την vj,max. Αν κριθεί απαραίτητη η διάσπαση ,θα πρέπει να
            % επιστρέψουμε στο Βήμα 2
            % Το συγκεκριμένο βήμα εκτελείται σε 2 μέρη
            % Σε πρώτη φάση ελέγχουμε αν ισχύει η συνθήκη που περιγράφεται
            % ανωτέρω με την βοήθεια της παρακάτω if εντολής.
            % Αν ισχύει, τότε θέτουμε bPartition=1 και εκτελούμε την
            % διάσπαση στο αμέσως επόμενο block εντολών, όπου ελέγχουμε την
            % τιμή της Partition
            if
(delta_max>Selta_s)&&((d(iClass)>d_mean&&getSize(Classes(iClass))>2*(Selta_n
+1))||Nc<=c/2)
                bPartition=1;
                break;
            end
        end
    end

    k=0.5;
    if 1==bPartition %αν ισχύει η συνθήκη διάσπασης, θα πρέπει να
διασπάσουμε
        centerPattern=getCenter(Classes(iClass)); % Βρες το κέντρο της
iClass
        Classes(iClass)=[];                       % Διέγραψε την iClass
        Nc=Nc+1;      % Αύξησε τον αριθμό των κλάσεων κατά 1, εφόσον
διασπάμε
        centerPatternPlus=centerPattern;
        centerPatternPlus(index)=centerPattern(index)+k*delta_max;  % το
%κέντρο βάρους της πρώτης κλάσης είναι ότι είχα συν k*vj,max
        centerPatternMinus=centerPattern;
        centerPatternMinus(index)=centerPattern(index)-k*delta_max; % το
%κέντρο βάρους της πρώτης κλάσης είναι οτι είχα μείον k*vj,max
```

```
        p=Classf(centerPatternPlus);  % μια νέα κλάση p,με κέντρο βάρους ότι
%είχα συν το k*vj,max
        Classes=[Classes,p];          % πρόσθεσέ την σε αυτές που έχω ήδη
        p=Classf(centerPatternMinus); % μία νέα κλάση p,με κέντρο βάρους ότι
%είχά συν το k*vj,max
        Classes=[Classes,p];          % πρόσθεσέ την σε αυτές που έχω ήδη
        iStep=iStep+1;                % αυξάνω τα steps κατά 1
        continue; % με την εντολή αυτή επιστρέφουμε στο ΒΗΜΑ 2,δηλαδή στην
%αρχή της while που εμπεριέχεται η continue
    end

    %*********************************************************

    if iStep==I
        Selta_D=0;
    end

    % ΒΗΜΑ 9 : Υπολογίζονται οι τιμές των αποστάσεων μεταξύ όλων των
    % ζευγαριών των θεωρούμενων κέντρων βαρών
    % ΒΗΜΑ 10 : Ταξινόμηση των αποστάσεων που υπολογίσθηκαν στο
    % προηγούμενο βήμα σε αύξουσα σειρά. Από το σύνολο αυτό επιλέγεται ένα
    % υποσύνολο  MAXPAIR to poly apostasewn, που αντιστοιχούν σε ζεύγη
    % ομάδων που η μεταξύ τους απόσταση  δεν υπερβαίνει την τιμή STDV.Για
    % κάθε τέτοιο ζεύγος συνόλων (Si,Sj) αν καμία από τις θεωρούμενες
    % ομάδες δεν έχουν εμπλακεί σε κάποια διαδικασία συγχώνευσης ,τότε τα
    % σύνολα των ομάδων Si,Sj αντικαθιστώνται από την συγχωνευμένη ομάδα
    % Ανανέωση της μεταβλητής k και όλων των δεικτών
    % στην αρχή υπολόγισε την απόσταση των κέντρων βάρους των κλάσεων 1,2
    % και θεώρησε αυτήν ως ελάχιστη.
    d=getDistance(Classes(1),getCenter(Classes(2)));
    i=1;
    j=2;
    for iClass=1:Nc-1
        for jClass=iClass+1:Nc
           % έπειτα υπολόγισε την απόσταση των κέντρων βάρους των
           % κλάσεων i,j
           d_ij=getDistance(Classes(iClass),getCenter(Classes(jClass)));

           % αν η νέα υπολογισμένη απόσταση είναι μικρότερη από την μέχρι
           % τώρα ελάχιστη, θεώρησέ την 1 ως ελάχιστη και αποθήκευσε
           % και τους δείκτες i,j
           % Με τον τόπο αυτό στην ουσία ταξινομεί τις αποστάσεις και
           % επιλέγω την ελάχιστη από αυτές
           if d>d_ij
              d=d_ij;
              i=iClass;
              j=jClass;
           end

        end
    end

    % έλεγχος αν η ελάχιστη απόσταση υπερβαίνει την προκαθορισμένη τιμή
    if d_ij>Selta_D
        iStep=iStep+1;
    end

    % στο παρακάτω κομμάτι του κώδικα κάνω συγχώνευση των δυο κλάσεων i,j
    iCenter=getCenter(Classes(i)); % το κέντρο βάρους της κλάσης i
    jCenter=getCenter(Classes(j)); % το κέντρο βάρους της κλάσης j

    Ni=getSize(Classes(i));        % ο αριθμός των δεδομένων της κλάσης i
    Nj=getSize(Classes(j));        % ο αριθμός των δεδομένων της κλάσης j
```

```
    newCenter=(iCenter*Ni+jCenter*Nj)/(Ni+Nj);  % το νέο κέντρο βάρους
    Classes(i)=[]; % σβήνω την κλάση i
    Classes(j)=[]; % σβήνω την κλάση j
    p=Classf(newCenter);%κατασκευάζω νέα κλάση με κέντρο βάρους το
%υπολογισμένο
  Classes=[Classes,p];  % και την προσθέτω σε αυτές που έχω ήδη
    Nc=Nc-1;              % μειώνω τον αριθμό των κλάσεων κατα 1
    iStep=iStep+1;

end
```

Συνάρτηση plot_class_patterns

```
function plot_class_patterns(X,I,Y,NameString,Labels)
 % Δεδομένα εισόδου της συνάρτησης  :
%                   I : Είναι ένα κελί μήκους N,όπου N είναι ο αριθμός
%                   των κλάσεων που υπάρχουν. Κάθε κελί αντιστοιχεί σε
%                   μία κλάση και περιέχει τους δείκτες των δεδομένων που
%                   ανήκουν στην κλάση αυτή. Δηλαδή X{3} = [1 4 5] σημαίνει
%                   οτι οι γραμμές 1 4 και 5 των δεδομένων ανήκουν στην
%                   κλάση 3
%                   X : τα δεδομένα σε πίνακα 2 διαστάσεων. Γραμμές ->
%                   πρότυπα και στήλες -> μεταβλητές
%                   NameString : είναι η συμβολοσειρά που θα εμφανιστεί ως
%                   τίτλος του figure εξόδου
% Η συνάρτηση αυτή τυπώνει τα δεδομένα I ανάλογα με την κλάση που ανήκουν.
% Η κλάση προσδιορίζεται από το X. Κάθε κλάση έχει συγκεκριμένο
% χρώμα(ένα εκ των κόκκινο, πράσινο, μπλε, κίτρινο και μαύρο) και σύμβολο
% (αστεράκι, σταυρό, κύκλο, τετράγωνο και διαμάντι),ώστε να ξεχωρίζει από
% τις υπόλοιπες .
% Επειδή χρησιμοποιούμε 5 χρώματα και 5 σύμβολα, μπορούμε να απεικονίσουμε
% συνολικά 25 διαφορετικές κλάσεις το πολύ.

ColorSpecifier = {'r','g','b','y','k'};
MarkerSpecifier = {'*','o','+','s','d'};
Specifier = cell(5,5);
for k = 1:1:5
    for m = 1:1:5
        Specifier{k,m} = strcat(ColorSpecifier{k},MarkerSpecifier{m});
    end;
end;
% με την reshape μετατρέπουμε την μεταβλητή Specifier από 5x5 σε 1x25
%Specifier = Specifier';
Specifier = reshape(Specifier,1,25);

% n είναι ο αριθμός των διαφορετικών κλάσεων που έχουμε

n = length(I);
k=0;
for i=1:n
    if(size(I{i},1)~=0)
    k=k+1
    end
end
```

```
Length = size(X,2); % ο αριθμός των μεταβλητών ,δηλαδή οι στήλες του πίνακα
%δεδομένων

if(Length>2)
    yX = X(:,[1,2]);
else
    yX = X;
end;
yX = X(:,[1:2]); % επιλέγουμε τις δυο πρώτες στήλες ,δηλαδή τις τιμές όλων
%των δεδομένων για τις δυο πρώτες μεταβλητές.

figure('Name',NameString) % Δίνουμε στο παράθυρο που ανοίγουμε την ονομασία
%που περιέχεται στην NameString μεταβλητή εισόδου
hold on % με την εντολή hold on διατηρούμε στο παράθυρο εξόδου τα δεδομένα
που έχουν ήδη απεικονισθεί και τυπώνουμε εκεί και τα νέα
% αν δεν την είχαμε καλέσει ,κάθε φορά που γίνεται plot, θα σβηνόντουσαν
% τα παλιά δεδομένα

for l = 1:1:k%για κάθε μία από τις δυνατές κλάσεις που μπορούμε να
%απεικονίσουμε (δλδ 25)
    % τύπωσε τα δεδομένα που ανήκουν στην κλάση αυτή
    % I{l}: έχει τους δείκτες των δεδομένων που αντιστοιχούν στην 1 κλάση
    % Για την 1 κλάση χρησιμοποίησε τον Specifier{l},που είναι μοναδικός
    % μεταξύ των 25 συνδυασμών

    plot(yX(I{l},1),yX(I{l},2),Specifier{l}),grid on
end;

%εμφάνισε τον αριθμό των κλάσεων
xlabel(Labels{1});
ylabel(Labels{2});
title(['number of classes=' num2str(k)]);

figure('Name',['class' NameString])
plot(Y,'ro')

hold off
```

Συνάρτηση clustering_tool3

```
function varargout = clustering_tool3(varargin)
% CLUSTERING_TOOL3 M-file for clustering_tool3.fig
%      CLUSTERING_TOOL3, by itself, creates a new CLUSTERING_TOOL3 or raises
the existing
%      singleton*.
%
%      H = CLUSTERING_TOOL3 returns the handle to a new CLUSTERING_TOOL3 or
the handle to
%      the existing singleton*.
%
%      CLUSTERING_TOOL3('CALLBACK',hObject,eventData,handles,...) calls the
local
%      function named CALLBACK in CLUSTERING_TOOL3.M with the given input
arguments.
%
%      CLUSTERING_TOOL3('Property','Value',...) creates a new
CLUSTERING_TOOL3 or raises the
%      existing singleton*.  Starting from the left, property value pairs
are
%      applied to the GUI before clustering_tool3_OpeningFcn gets called.
An
```

```
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to clustering_tool3_OpeningFcn via
varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @clustering_tool3_OpeningFcn, ...
                   'gui_OutputFcn',  @clustering_tool3_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before clustering_tool3 is made visible.
function clustering_tool3_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to clustering_tool3 (see VARARGIN)

% Choose default command line output for clustering_tool3
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes clustering_tool3 wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = clustering_tool3_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --------------------------------------------------------------------
function open_database_Callback(hObject, eventdata, handles)
% hObject    handle to open_database (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
[FileName,PathName] = uigetfile({'*.mdb';'*.xls'},'Select the database to
connect');
mydata = guidata(hObject);
mydata.FileName = FileName;
guidata(hObject,mydata);

% --- Executes on button press in pushbutton1 (K means Clustering).
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%------------------
%Η συγκεκριμένη συνάρτηση εκτελείται όταν πιεστεί το  button 'Set
% Clustering Parameters' του panel 'Clustering Operations)
%----------------------
mydata = guidata(hObject);
set(mydata.uipanel2,'Visible','on');

% --- Executes during object creation, after setting all properties.
function uitable1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to uitable1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called


% ------------------------------------------------------------------------
function select_database_Callback(hObject, eventdata, handles)
% hObject    handle to select_database (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
mydata = guidata(hObject);
DatabaseName = strtok(mydata.FileName,'.')
DatasourceName = strcat(DatabaseName,'_source')
% Σύνδεση με την βάση δεδομένων.
conn = database(DatasourceName,'','')
% Κάνει Ping με την βάση δεδομένων για να δει την κατάσταση την σύνδεσης.
ping(conn)
% Define the query command.
command = strcat(['select * from ',DatabaseName])
% Execute the SQL query command.
%curs = exec(conn,'select * from breast_cancer_data');
curs = exec(conn,command);
% Μετατρέπει τα περιεχόμενα της βάσης δεδομένων σε ένα cell array .
setdbprefs('DataReturnFormat','cellarray');
curs = fetch(curs)
colnames = columnnames(curs)
%Επιστρέφει ένα cell array από strings το οποίο περιέχει τα ονόματα των
%στηλών στον πίνακα.
colnum = 0;
remain = colnames;
col_names = {};
while(true)
    [token,remain] = strtok(remain,',');
    if(isempty(token))
        break;
    else
        colnum = colnum + 1;
        token = token(2:1:end-1);
        col_names{colnum} = token;
    end;
end;
data = cell2mat(curs.Data)
%data = unique(data,'rows');
whos curs
```

```
% Κλείνει την σύνδεση με την βάση.
close(conn)
% Set the data table.
columnformat = {'numeric', 'bank', []};
set(mydata.uitable1,'ColumnFormat',columnformat);
set(mydata.uitable1,'Data',data);
set(mydata.uitable1,'ColumnName',col_names);
set(mydata.uitable1,'Visible','on');
% Αποθηκεύουμε όλες τις χρήσιμες μεταβλητές στην δομή mydata.
mydata.DatabaseName = DatabaseName;
mydata.DatasourceName = DatasourceName;
mydata.data = data;
mydata.col_names = col_names;
guidata(hObject,mydata);



function k_edit_Callback(hObject, eventdata, handles)
% hObject    handle to k_edit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of k_edit as text
%        str2double(get(hObject,'String')) returns contents of k_edit as a
double


% --- Executes during object creation, after setting all properties.
function k_edit_CreateFcn(hObject, eventdata, handles)
% hObject    handle to k_edit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Ο παρακάτω κώδικας διαβάζει τις παραμέτρους που χρειάζονται από το panel
% με ονομασία 'Clustering Parameters'.
% Κάνει πρόσβαση στα edit boxes που υπάρχουν ,για να διαβάσει τις τιμές των
% παραμέτρων
% Σημείωση : Από τα edit boxes μπορούμε να διαβάσουμε μόνο συμβολοσειρές
% (strings). Οπότε με την συνάρτηση get() γνωρίζουμε το περιεχόμενο του
% edit box, αλλά ως συμβολοσειρά. Επειδή οι συγκεκριμένοι παράμετροι είναι
% αριθμοί, με την str2double τους μετατρέπουμε σε αριθμούς
mydata = guidata(hObject);
Kstring = get(mydata.k_edit,'String'); %  K: ο επιθυμητός αριθμός των
κλάσεων για τον k-means
NUMCLUSstring = get(mydata.numclus_edit,'String');% ο αρχικός αριθμός των
κλάσεων για τον isodata (clusters)
SAMPARstring = get(mydata.sampar_edit,'String'); % ο ελάχιστος αριθμός
δεδομένων που  μπορεί να περιλαμβάνει μια ομάδα
```

```
MAXITERstring = get(mydata.maxiter_edit,'String');% ο μέγιστος αριθμός των
επαναλήψεων
STDVstring = get(mydata.stdv_edit,'String'); %η μέση τυπική απόκλιση των
σημείων από το κέντρο βάρους της ομάδας
LUMPstring = get(mydata.lump_edit,'String');% η ελάχιστη απαιτούμενη
απόσταση μεταξύ των κέντρων βάρους των δυο ομάδων
MAXPAIRstring = get(mydata.maxpair_edit,'String');% ο μέγιστος αριθμός από
ζεύγη ομάδων, που μπορούν να  συγχωνευθούν σε μια επανάληψη
% Μετατρέπει τα strings σε αριθμητικές τιμές.
k = str2num(Kstring);
numclus = str2double(NUMCLUSstring);
sampar = str2double(SAMPARstring);
maxiter = str2double(MAXITERstring);
stdv = str2double(STDVstring);
lump = str2double(LUMPstring);
maxpair = str2double(MAXPAIRstring);
% Αποθηκεύει όλες τις χρήσιμες μεταβλητές στην δομή mydata.

% Πλέον θα πρέπει να αποθηκεύσουμε όλες τις παραμέτρους διαβάσαμε στην
% μεταβλητή mydata, ώστε να εκτελεσθεί η guidata() και να είναι εμφανείς οι
% αλλαγές και στις άλλες συναρτήσεις του συγκεκριμένου project
mydata.k = k;
mydata.numclus = numclus;
mydata.sampar = sampar;
mydata.maxiter = maxiter;
mydata.stdv = stdv;
mydata.lump = lump;
mydata.maxpair = maxpair;
guidata(hObject,mydata);

function numclus_edit_Callback(hObject, eventdata, handles)
% hObject    handle to numclus_edit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of numclus_edit as text
%        str2double(get(hObject,'String')) returns contents of numclus_edit
as a double


% --- Executes during object creation, after setting all properties.
function numclus_edit_CreateFcn(hObject, eventdata, handles)
% hObject    handle to numclus_edit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function sampar_edit_Callback(hObject, eventdata, handles)
% hObject    handle to sampar_edit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of sampar_edit as text
%        str2double(get(hObject,'String')) returns contents of sampar_edit
as a double
```

```
% --- Executes during object creation, after setting all properties.
function sampar_edit_CreateFcn(hObject, eventdata, handles)
% hObject    handle to sampar_edit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function maxiter_edit_Callback(hObject, eventdata, handles)
% hObject    handle to maxiter_edit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of maxiter_edit as text
%        str2double(get(hObject,'String')) returns contents of maxiter_edit
as a double


% --- Executes during object creation, after setting all properties.
function maxiter_edit_CreateFcn(hObject, eventdata, handles)
% hObject    handle to maxiter_edit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function stdv_edit_Callback(hObject, eventdata, handles)
% hObject    handle to stdv_edit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of stdv_edit as text
%        str2double(get(hObject,'String')) returns contents of stdv_edit as
a double


% --- Executes during object creation, after setting all properties.
function stdv_edit_CreateFcn(hObject, eventdata, handles)
% hObject    handle to stdv_edit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function lump_edit_Callback(hObject, eventdata, handles)
% hObject    handle to lump_edit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of lump_edit as text
%        str2double(get(hObject,'String')) returns contents of lump_edit as
a double


% --- Executes during object creation, after setting all properties.
function lump_edit_CreateFcn(hObject, eventdata, handles)
% hObject    handle to lump_edit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function maxpair_edit_Callback(hObject, eventdata, handles)
% hObject    handle to maxpair_edit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of maxpair_edit as text
%        str2double(get(hObject,'String')) returns contents of maxpair_edit
as a double


% --- Executes during object creation, after setting all properties.
function maxpair_edit_CreateFcn(hObject, eventdata, handles)
% hObject    handle to maxpair_edit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function database_detour(hObject)
mydata = guidata(hObject);
DatabaseName = strtok(mydata.FileName,'.');
filename = strcat(DatabaseName,'.xls');
[data,colnames] = xlsread(filename);
% Set the data table.
```

```
columnformat = {'numeric', 'bank', []};
set(mydata.uitable1,'ColumnFormat',columnformat);
set(mydata.uitable1,'Data',data);
set(mydata.uitable1,'ColumnName',colnames);
set(mydata.uitable1,'Visible','on');
mydata.data = data;
guidata(hObject,mydata);

function update_table(hObject)
mydata = guidata(hObject);
data = mydata.data;% Φόρτωσε όλα τα αρχικά δεδομένα
col_names = mydata.col_names;% τα ονόματα των στηλών
KMeansClusterIndices = mydata.KMeansClusterIndices;% φόρτωσε τους δείκτες
%που δείχνουν πως ταξινομήθηκαν τα δεδομένα με τον k-means
IsodataClusterIndices = mydata.IsodataClusterIndices;% φόρτωσε τους δείκτες
%που δείχνουν πως ταξινομήθηκαν τα δεδομένα με τον Isodata
L = length(col_names);
[Rows,Columns] = size(data);
col_names{L+1} = 'KMeansClusterIndices';% βάλε τα ονόματα των 2 επόμενων
%στηλών, η μία στήλη για k-means και η άλλη για isodata
col_names{L+2} = 'IsodataClusterIndices';
data(:,Columns+1) = KMeansClusterIndices;% βάλε και τους αντίστοιχους
%δείκτες στις 2 στήλες
data(:,Columns+2) = IsodataClusterIndices;
% ξαναγράψε τα νέα δεδομένα στον πίνακα
columnformat = {'numeric', 'bank', []};
set(mydata.uitable1,'ColumnFormat',columnformat);
set(mydata.uitable1,'Data',data);
set(mydata.uitable1,'ColumnName',col_names);
% σώσε τις αλλαγές στην mydata και τρέξε την guidata(), ώστε να γνωρίζουν
% οι άλλες συναρτήσεις τις αλλαγές που έγιναν  στην μεταβλητή αυτή εντός της
% συγκεκριμένης συνάρτησης
mydata.data = data;
mydata.col_names = col_names;
guidata(hObject,mydata);
% --------------------------------------------------------------------
function load_database_Callback(hObject, eventdata, handles)
% hObject    handle to load_database (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
database_detour(hObject);


% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%-----------------
% Η συγκεκριμένη συνάρτηση εκτελείται όταν πιεστεί το button 'Apply
% Clustering Algorithms' του panel 'Clustering Operations'
%-----------------
% με την εντολή addpath() μπορούμε να συμπεριλάβουμε όλα τα  .m files
% που βρίσκονται στον συγκεκριμένο φάκελο. έπειτα μπορούμε να τα καλέσουμε
% ώστε να εκτελεσθούν
% κάνουμε addpath τον φάκελο που περιέχει τα .m αρχεία με τον Isodata
% αλγόριθμο
addpath('isomatlab4');

mydata = guidata(hObject);% φόρτωσε την μεταβλητή mydata, που περιέχει όλες
τις πληροφορίες που αποθηκεύτηκαν από άλλες συναρτήσεις
% πρέπει να δει ποιές μεταβλητές έχει επιλέξει ο χρήστης από την λίστα
% 'Selected Variables'
```

```
% Σημείωση : κανονικά το ίδιο κάνει και το  button 'Get Selected Variables'
% του panel 'Clustering Operations', αλλά εδώ γίνεται ξανά προς σιγουριά
% Βλέπει τους δείκτες των μεταβλητών της δεξιάς λίστας,
right_list = mydata.right_list;
right_list_indices = mydata.right_list_indices;
%διαβάζει όλα τα δεδομένα
patterns = mydata.data;
% και επιλέγει μόνο αυτά που αντιστοιχούν στις επιλεγμένες μεταβλητές
patterns = patterns(:,right_list_indices);
% Σημείωση
% Η εντολή π.χ.
% A = B(:,3)
% πηγαίνει στον πίνακα 2 διαστάσεων B, και παίρνει την στήλη No3 και
% όλες τις γραμμές (από το ':')
% άρα στην ουσία η εντολή
% patterns = patterns(:,right_list_indices);
% διαβάζει από όλα τα δεδομένα (patterns), μόνο εκείνες τις στήλες(δηλαδή
% μεταβλητές)που έχει επιλέξει ο χρήστης
%
% Να θυμίσουμε οτι τα δεδομένα είναι αποθηκευμένα σε πίνακα 2 διαστάσεων.
% Η πρώτη διάσταση (γραμμές) αντιστοιχεί σε διαφορετικό δεδομένο.
% Δηλαδή το πρώτο πρότυπο είναι στην πρώτη γραμμή, το δεύτερο στην δεύτερη
% κοκ
% Η δεύτερη διάσταση (στήλες) αντιστοιχεί σε κάθε μεταβλητή. Δλδ οι τιμές
% της πρώτης μεταβλητής(x1) είναι αποθηκευμένες στην πρώτη στήλη κοκ
% Άρα π.χ. το σημείο (10,2) μας δίνει την τιμή του δέκατου προτύπου για
% την δεύτερη μεταβλητή
% στην αρχή θα εκτελεστεί ο k-means αλγόριθμος και μετά ο isodata, ώστε
% να φανούν και οι διαφορές τους

% εκτέλεση του k-means
K = mydata.k;% βρες ποιά είναι η τιμή του K που έδωσε ο χρήστης στο panel
'Clustering Parameters'

[KMeansClusterIndices,Centers] = kmeans(patterns,K);
% Εδώ καλούμε την έτοιμη συνάρτηση k-means του Matlab. Γνωρίζουμε ότι:
%   [IDX, C] = KMEANS(X, K) επιστρέφει τα κέντρα βάρους των K κλάσεων
cluster centroid locations in
%   the K-by-P matrix C.
% Άρα δίνουμε στην συνάρτηση την τιμή του K και τα δεδομένα μας
% (patterns) και μας επιστρέφει τα κέντρα βάρους των κλάσεων και τους
αντίστοιχους
% δείκτες ,που μας δείχνουν που ακριβώς (σε ποιό cluster δηλαδή)έχει
% ταξινομηθεί κάθε δεδομένο.
% Συγκεκριμένα, αν IDX(4)=1, σημαίνει οτι το δεδομένο που αντιστοιχεί στην
% τέταρτη σειρά των δεδομένων έχει ταξινομηθεί στο cluster με κέντρο C(1)
% αποθήκευσε τους δείκτες που δείχνουν την ταξινόμηση των δεδομένων με
% βάση τον αλγόριθμο k-means στην μεταβλητή mydata, ώστε να γνωρίζουν και
% οι άλλες συναρτήσεις.
mydata.KMeansClusterIndices = KMeansClusterIndices;
% φτιάξε μια μεταβλητή I, που να έχει τόσα κελιά όσες και οι K κλάσεις που
% έχουμε
% Σε κάθε κελί θα αποθηκεύσουμε τους δείκτες που αντιστοιχούν στα
% δεδομένα μίας κλάσης. Με τον τρόπο αυτό θα γνωρίζουμε ποία ακριβώς
% δεδομένα έχουν ταξινομηθεί σε ποιά κλάση, και θα μπορούμε να τα
% τυπώσουμε με την plot_class_patterns
I = cell(1,K);

for m = 1:1:K
    I{m} = find(KMeansClusterIndices==m);
    % στο m κελί αποθήκευσε τους δείκτες που έχουν την τιμή m
end;
plot_class_patterns(patterns,I,KMeansClusterIndices,'K Means Clustring
Results',right_list);
```

```
% Τώρα εκτελούμε τον αλγόριθμο ISODATA και παρατηρούμε τις αλλαγές σε
% σύγκριση με τον k-means
% πρέπει να βρούμε τις τιμές των παραμέτρων που έχουν διαβαστεί από τα
% edit boxes με χρήση άλλης συνάρτησης και έχουν αποθηκευτεί στην
% μεταβλητή mydata
numclus = mydata.numclus;% ο αρχικός αριθμός των κλάσεων
sampar = mydata.sampar;% ο ελάχιστος αριθμός των δεδομένων που μπορεί να
περιλαμβάνει μια ομάδα
maxiter = mydata.maxiter;% ο μέγιστος αριθμός επαναλήψεων του αλγορίθμου
stdv = mydata.stdv; % η μέγιστη τυπική απόκλιση των σημείων από
%το κέντρο βάρους της ομάδας κατά μήκος κάθε άξονα
lump = mydata.lump;% η ελάχιστη απαιτούμενη απόσταση μεταξύ του κέντρου
βάρους δύο διαφορετικών ομάδων
maxpair = mydata.maxpair;% ο μέγιστος αριθμός από ζεύγη ομάδων  τα οποία
μπορούν να συγχωνευθούν σε μία επανάληψη
[Rows,Columns] = size(patterns);
Patterns = patterns';% απλώς άλλαξε τα δεδομένα σου, δημιουργώντας τον
ανάστροφο πίνακα,
%γιατί η isodata() θέλει σε στήλες τα δεδομένα και σε  σειρές τις μεταβλητές
% Parameters mapping.
c = numclus;
Nc = sampar;
Iterations = maxiter;
L = maxpair;
Selta_n=2;
Selta_s=1;
Selta_D=8;
Classes=ISODATA(Patterns,c,Nc,Selta_n,Selta_s,Selta_D,L,Iterations);%
εκτέλεση του αλγορίθμου isodata.Μας επιστρέφει ταξινομημένα τα δεδομένα
ClusterPatterns = cell(1,length(Classes));% φτιάξε ένα κελί με τόσο μέγεθος
όσα και τα clusters-omades
IsodataClusterIndices = zeros(Rows,1);% ένα διάνυσμα με δείκτες,κάθε γραμμή
δείχνει την ομάδα που ταξινομήθηκε το αντίστοιχο δεδομένο-pattern
n = 0;% το πλήθος των εντέλει ταξινομημένων δεδομένων
Patterns = [];
for i = 1:1:length(Classes)% για κάθε μία από τις 4 κλάσεις-ομάδες, γέμισε
τες με τα αντίστοιχα δεδομένα
    ClusterPatterns{i} = getPatterns(Classes(i))';% βάλε τα δεδομένα της
κλάσης
    x = ClusterPatterns{i};
    [Intersection,Indices] = intersect(patterns,x,'rows');% βρες σε ποιές
σειρές των αρχικών δεδομένων ταιριάζουν
    %τα δεδομένα της συγκεκριμένης κλάσης, δλδ στην ουσία τα δεδομένα ποιών
    %γραμμών(των αρχικών data) ταξινομήθηκαν στην συγκεκριμένη κλάση
    IsodataClusterIndices(Indices) = i;% και βάλε στο διάνυσμα δεικτών τις
αντίστοιχες σειρές να είναι ίσες
    %με τον αριθμό της συγκεκριμένης κλάσης
    Patterns = [Patterns;ClusterPatterns{i}];% ανανέωσε τον πίνακα με όλα υα
ταξινομημένα πρότυπα
    n = n + size(ClusterPatterns{i},1);% όπως επίσης και το πλήθος όλων των
ταξινομημένων προτύπων
end;
mydata.IsodataClusterIndices = IsodataClusterIndices;% αποθήκευσε τους
δείκτες που δείχνουν την ταξινόμηση που έγινε από τον isodata
%στην μεταβλητή mydata, ώστε να γνωρίζουν και υπόλοιπες συναρτήσεις την
ταξινόμηση
% Φτιάξε μια μεταβλητή I, που να έχει τόσα κελιά όσες και οι K klaseis που
έχουμε
% σε κάθε κελί θα αποθηκεύσουμε τους δείκτες που αντιστοιχούν στα
% δεδομένα μιας κλάσης. Με τον τρόπο αυτό θα γνωρίζουμε ποιά ακριβώς
% δεδομένα ακριβώς έχουν ταξινομηθεί σε ποιά κλάση, και θα μπορούμε να τα
% τυπώσουμε με την plot_class_patterns
I = cell(1,length(Classes));
Nprev = 0;
```

```
for i = 1:1:length(Classes)
    Ncurr = size(ClusterPatterns{i},1);% το πλήθος των δεδομένων που
ανοίκουν στην κλάση i
    I{i} = [Nprev+1:1:Nprev+Ncurr];
    Nprev = Nprev + Ncurr;
end;
guidata(hObject,mydata);
update_table(hObject);% στον πίνακα που απεικονίζεται στην έξοδο(clustering
tool) θα εμφανίσουμε
%τα αποτελέσματα των ταξινομήσεων, δλδ κάθε δεδομένα(γραμμή του πίνακα)θα
πρέπει να ξέρουμε σε ποιά κλάση ταξινομήθηκε με την k-means
%και με την isodata, οπότε έχουμε άλλες 2 επιπλέον στήλες, αυτήν την δουλεία
την κάνει η update_table
% τύπωσε στο figure εξόδου τα ταξινομημένα δεδομένα

I2 = cell(1,length(Classes));
lc=length(Classes);
if (min(IsodataClusterIndices)==0 )
        IsodataClusterIndices2=IsodataClusterIndices+1;
        lc=length(Classes)+1
end

for m = 1:1:lc
    I2{m} = find(IsodataClusterIndices2==m);
end;
plot_class_patterns(patterns,I2,IsodataClusterIndices2,'ISODATA Clustering
Results',right_list)


% --- Executes on selection change in LeftList.
function LeftList_Callback(hObject, eventdata, handles)
% hObject    handle to LeftList (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns LeftList contents as cell
array
%        contents{get(hObject,'Value')} returns selected item from LeftList


% --- Executes during object creation, after setting all properties.
function LeftList_CreateFcn(hObject, eventdata, handles)
% hObject    handle to LeftList (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: listbox controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on selection change in RightList.
function RightList_Callback(hObject, eventdata, handles)
% hObject    handle to RightList (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns RightList contents as cell
array
%        contents{get(hObject,'Value')} returns selected item from RightList
```

```
% --- Executes during object creation, after setting all properties.
function RightList_CreateFcn(hObject, eventdata, handles)
% hObject    handle to RightList (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: listbox controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% ---
% Η συγκεκριμένη εκτελείται όταν πιεστεί το button 'Add variable
% ' του panel 'Select table columns'
% ---
mydata = guidata(hObject);
selected_index = get(mydata.LeftList,'Value');% Δες ποιά γραμμή της
αριστερής λίστας έχει επιλεχθεί

s1 = get(mydata.LeftList,'String');
s2=size(s1,1);
% Σημείωση: η μεταβλητή mydata περιέχει πληροφορίες που έχουν αποθηκευτεί
% από άλλες συναρτήσεις
selected_variables_num =0;
if(selected_variables_num <= s2)
    left_list = mydata.left_list;
    right_list = mydata.right_list;
    left_list_indices = mydata.left_list_indices;
    right_list_indices = mydata.right_list_indices;
    selected_variables_num = selected_variables_num + 1;% αύξησε τις
επιλεγμένες μεταβλητές κατα μία
    [right_list,left_list] =
UpdateLists(right_list,left_list,selected_index);% κάλεσε την
UpdateLists(),για να ενημερωθούν οι 2 λίστες
    [right_list_indices,left_list_indices] =
UpdateListsIndices(right_list_indices,left_list_indices,selected_index);
    % αποθήκευσε τις αλλαγές στην μεταβλητή mydata
    mydata.left_list = left_list;
    mydata.right_list = right_list;
    mydata.right_list_indices = right_list_indices
    mydata.left_list_indices = left_list_indices
    mydata.selected_variables_num = selected_variables_num;
    % εμφάνισε στις δυο λίστες τις αλλαγές που υπολόγισες
    set(mydata.LeftList,'String',left_list,'Value',1);
    set(mydata.RightList,'String',right_list,'Value',1);
   % κάλεσε την guidata(),ώστε ναι ισχύουν οι αλλάγες και στις άλλες
συναρτήσεις
    s1 = get(mydata.LeftList,'String');
    s2=size(s1,1);
    guidata(hObject,mydata);
end;
% κάλεσε την guidata(),ώστε να αποθηκευτούν οι αλλαγές που έγιναν στις
% μεταβλητές εντός της συγκεκριμένης συνάρτησης
guidata(hObject,mydata);
```

```
% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% ---
% Η συγκεκριμένη συνάρτηση εκτελείται όταν πιεστεί το button 'Remove
variable
% ' του panel 'Select table columns'
% ---


mydata = guidata(hObject);
selected_index = get(mydata.RightList,'Value');% Δες ποιά γραμμή της δεξιάς
λίστας έχει επιλεχθεί
% Σημείωση: η μεταβλητή mydata περιέχει πληροφορίες που έχουν αποθηκευτεί
% από άλλες συναρτήσεις
selected_variables_num = size( get(mydata.RightList,'String'),1);
if(selected_variables_num > 0)% μπορώ να αφαιρέσω μόνο αν έχω ήδη επιλέξει
κάτι
    left_list = mydata.left_list;% τα στοιχεία της αριστερής λίστας ,οπου
μπορώ να προσθέσω 1
    right_list = mydata.right_list;% τα στοιχεία της δεξιάς λίστας ,οπου
μπορώ να αφαιρέσω 1
    left_list_indices = mydata.left_list_indices;
    right_list_indices = mydata.right_list_indices;
    [left_list,right_list] =
UpdateLists(left_list,right_list,selected_index);% κάλεσε την updateLists()
    %με την αντίστροφη φορά των left-right λιστών
    [left_list_indices,right_list_indices] =
UpdateListsIndices(left_list_indices,right_list_indices,selected_index);
    selected_variables_num = selected_variables_num - 1;

    %αποθήκευσε τις αλλάγες στην μεταβλητή mydatamydata.left_list =
left_list;
    mydata.right_list = right_list;
    mydata.left_list_indices = left_list_indices
    mydata.right_list_indices = right_list_indices
    mydata.selected_variables_num = selected_variables_num;
    % εμφάνισε στις δύο λίστες τις αλλαγές που υπολόγισες
    set(mydata.LeftList,'String',left_list,'Value',1);
    set(mydata.RightList,'String',right_list,'Value',1);
    % κάλεσε την guidata(),ώστε να ισχύουν οι αλλαγές και στις άλλες
    % συναρτήσεις
    guidata(hObject,mydata);
end;
% κάλεσε την guidata(),ώστε να αποθηκευτούν οι αλλαγές που έγιναν στις
% μεταβλητές εντός της συγκεκριμένης συνάρτησης
guidata(hObject,mydata);

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
mydata = guidata(hObject);
set(mydata.uipanel3,'Visible','on');% κάνε ορατό το panel με τίτλο 'Select
Table Columns'
col_names = mydata.col_names; % πάρε τα ονόματα της κάθε στήλης
left_list = col_names;% αρχικά αριστερά έχεις όλα τα ονόματα
right_list = {}; % και δεξιά τίποτα
left_list_indices = [1:1:length(left_list)];
right_list_indices = [];
selected_variables_num = 0;%ο αριθμός των επιλεγμένων μεταβλητών
% Αποθήκευσε τις παρακάτω μεταβλητές στο struct mydata και κάλεσε την
% guidata(),ώστε να γνωρίζουν τις αλλάγές  και οι συναρτήσεις εκτός της
% ίδιας
mydata.selected_variables_num = selected_variables_num;
mydata.left_list = left_list;
mydata.right_list = right_list;
mydata.left_list_indices = left_list_indices;
mydata.right_list_indices = right_list_indices;
set(mydata.LeftList,'String',left_list,'Value',1);
% βάλε στην αριστερή λίστα 'Available Variables' του συγκεκριμένου panel
% ο,τι έχεις (δηλαδή όλα τα column names)
set(mydata.RightList,'String',right_list);
% βαλε στην δεξιά λίστα 'Selected Variables' του συγκεκριμένου panel
% ο,τι έχεις(δηλαδή στην ουσία τίποτε)
guidata(hObject,mydata);
% αποθήκευση αλλαγών


% --- Executes on button press in pushbutton8.
function pushbutton8_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% ---
% Η συγκεκριμένη συνάρτηση εκτελείται όταν πιεστεί το button 'Plot
% variables' του panel 'Select table columns'
% ---

mydata = guidata(hObject);% φόρτωση της μεταβλητής mydata
selected_variables_num = mydata.selected_variables_num;
if(selected_variables_num==2)
    data = mydata.data;
    right_list_indices = mydata.right_list_indices;% βρές τα ονόματα των
επιλεγμένων μεταβλητών
    right_list = mydata.right_list;% βρες τους δείκτες των επιλεγμένων
μεταβλητών
    data = data(:,right_list_indices); % βρες τις τιμές των δεδομένων για
τις επιλεγμένες μεταβλητές
    plot_patterns(data,right_list);% τύπωσε τα επιλεγμένα δεδομένα με την
βοήθεια της plot_patterns()
end;
guidata(hObject,mydata);

function [AddList,RemoveList] =
UpdateLists(AddList,RemoveList,selected_index)
% ---
% Η συγκεκριμένη συνάρτηση καλείται όταν θέλουν να ενημερωθούν οι δύο
% λίστες του panel 'Select Table Columns'
% ---

L = length(RemoveList);
RemainingIndices = [[1:1:selected_index-1],[selected_index+1:1:L]];
AddList{end+1} = RemoveList{selected_index};
RemoveList = RemoveList(RemainingIndices);
```

```
function [AddListIndices,RemoveListIndices] =
UpdateListsIndices(AddListIndices,RemoveListIndices,selected_index)
% ---
% Η συγκεκριμένη συνάρτηση είναι όμοια με την UpdateLists(), αλλά δεν
% ενημερώνει τις συμβολοσειρές ,αλλά τους δείκτες.
% ---

L = length(RemoveListIndices);
AddListIndices = [AddListIndices,RemoveListIndices(selected_index)];
RemoveListIndices = RemoveListIndices([[1:1:selected_index-
1],[selected_index+1:1:L]]);
```

Παράρτημα Β

Ευρετήριο Εικόνων

Βιβλιογραφία

[1]Αργιαλάς Δ. Π. (1998), Ψηφιακή Τηλεπισκόπιση

[2]A. Jain and R. Dubes. *"Algorithms for Clustering Data"*. Prentice-Hall, New Jersey,1988.

[3]Bertrand, and B. Burtsch, editors, *"New Approaches in Classification and Data Analysis"*, pages 3—24. Springer-Verlag, 1994.

[4]B. Everitt. *"Cluster Analysis"*. John Wiley & Sons, New York, 1974

5]Computer processing of remotely sensed images: an introduction Από τον/την Paul M. Mather

[6] Ηλίας Δημητρίου «Χρήση προϊόντων Τηλεπισκόπησης για την αποτύπωση μεταβολών χρήσεων γης και για την διαχείριση των υδατικών πόρων της υδρολογικής λεκάνης της λίμνης Τριχωνίδας.». Ελληνικό Κέντρο Θαλασσίων Ερευνών – Ινστιτούτο Εσωτερικών Υδάτων.

[7]Guojun Gan,Chaogun Ma and Jianhong Wu,"Data Clustering Theory,Algorithms and Applications, Asa siam

[8]H. Bock. "Classification and Clustering: Problems for the Future". In E. Diday, Y. Lechevallier, M. Schader, P. T.Michel, Machine Learning.The McGraw-Hill Companies ,Inc,1997

[9]L.Breiman J.H. Friedman R.A Olshen & C.J Stone .Classification and Regression Trees.Report Technique, Wadsworth International , Monterey ,CA 1984

[10]M. Anderberg. *"Cluster Analysis for Applications"*. Academic Press, New York, 1973.

[11]N. Venkateswarlu and P. Raju. "Fast ISODATA Clustering Algorithms". *PatternRecognition,* 25(3):335—342, 1992.

[12]Pavel Berkhin,Survey of clustering Data mining Techniques, Technical Report, Accrue Software 2002

[13]http://proceedings.esri.com/library/userconf/proc00/professional/papers/pap694/p694.htm

[14]http://www.optimaldesign.com/ArrayMiner/KMeans/KMeans1.html

[15]http://archive.ics.uci.edu/ml/

[16]http://www.cs.umd.edu/~mount/Projects/ISODATA/igarss03.pdf