



ΜΕΘΟΔΟΛΟΓΙΕΣ ΟΜΑΔΟΠΟΙΗΣΗΣ ΜΕ ΕΞΕΛΙΚΤΙΚΕΣ ΤΕΧΝΙΚΕΣ

Μιχαλόπουλος Ιωάννης

Επιβλέπων καθηγητής :Αναστάσιος Ντούνης

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΤΜΗΜΑ ΑΥΤΟΜΑΤΙΣΜΟΥ

ΜΑΡΤΙΟΣ (2014)

Περιεχόμενα

Κεφάλαιο 1 :Μεθοδολογίες ομαδοποίησης.....	2
1.1 Fuzzy c means.....	2
1.2 Μέθοδος Subtractive clustering.....	7
1.3 Μέθοδος Gustafson Kessel	12
1.4 Μέθοδος Self Organized Maps.....	17
Κεφάλαιο 2 Συστήματα Ασαφούς Λογικής.....	24
2.1 Εισαγωγή	24
2.2 Ασαφές σύνολο	24
2.3 Η έννοια του κανόνα.....	25
2.4 Ασαφή συστήματα τύπου Sugeno.....	26
2.5 Συμπεράσματα και παρατηρήσεις.....	29
Κεφάλαιο 3: Εφαρμογές σε δεδομένα ρυζιού και PMV.....	30
3.1 Δεδομένα ρυζιού-Δειγματοληψία PMV.....	30
3.2 Λειτουργία Genfis2.m.....	31
3.3 Εφαρμογή.....	32
Κεφάλαιο 4: Η ΜΕΘΟΔΟΣ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ BIG BANG BIG CRUNCH.....	46
4.1 Εισαγωγή.....	46
4.2 Big-Bang Big-Crunch (BB-BC).....	46
4.3 Παραδείγματα.....	49
4.4 Best Fit Solution.....	51
4.5 Συμπεράσματα και παρατηρήσεις	53

Κεφάλαιο 5 : Βελτιστοποίηση συστήματος ασαφούς λογικής με τη μέθοδο Big Bang – Big Crunch.....	54
5.1 Εισαγωγή	54
5.2 Λειτουργία	54
5.3 Μαθηματική ανάλυση.....	55
5.4 Εφαρμογή	57
5.5 Παρατηρήσεις.....	59
Κεφάλαιο 6: Συνδυασμός και εφαρμογή διάφορων μεθοδολογιών.....	60
6.1 Υβριδική μέθοδος GK-BBBC.....	60
6.2 Μοντελοποίηση της συνάρτησης $f(x) = \sin(2x) / e^{\frac{x}{5}}$	62
6.3 Μοντελοποίηση των μετρήσεων ρυζιού.....	64
6.4 Μοντελοποίηση PMV.....	68
6.5 Γενικές παρατηρήσεις.....	71
Παράρτημα Α.....	72
Παράρτημα Β.....	82

ΚΕΦΑΛΑΙΟ 1: Μεθοδολογίες ομαδοποίησης.

Εισαγωγή

Η ραγδαία ανάπτυξη της πληροφορικής και των τηλεπικοινωνιών έχει οδηγήσει στην απόκτηση τεράστιου όγκου πληροφοριών και κατά συνέπεια στην δημιουργία αναγκών για γρήγορη και αξιόπιστη εξόρυξη γνώσης από βάσεις δεδομένων. Αυτή η επιστήμη ονομάζεται *datamining* και οι αλγόριθμοι ομαδοποίησης είναι ένα τμήμα της.

Η ομαδοποίηση φροντίζει στο να χωριστούν τα δεδομένα σε ομάδες ανάλογα με το πόσο ταιριάζουν μεταξύ τους, συγκεκριμένα ένας αλγόριθμος ομαδοποίησης χρησιμοποιεί την απόσταση του κάθε δεδομένου από τα γύρω του και αναλόγως πόσο απομακρυσμένα είναι τα ομαδοποιεί.

1.1 Fuzzy C-means

Ο αλγόριθμος *fuzzy c-means* είναι μια επέκταση του πασίγνωστου αλγορίθμου *k-means* όπου χάρις την λειτουργικότητα και την ευελιξία του χρησιμοποιείται ευρέως σε στατιστικά μοντέλα, *machine learning* και διάφορες ευριτικές συναρτήσεις .

1.1.α Λειτουργία

Αρχικά, ο αλγόριθμος δέχεται από τον χρήστη τον αριθμό των ομάδων στα οποία θα διαχωριστούν τα δεδομένα (*cluster means*). Ανάλογα την εφαρμογή, στα κέντρα μπορεί να δοθεί τυχαία αρχική τιμή ή συγκεκριμένη από τον χρήστη .

Στη συνέχεια υπολογίζετε η ευκλείδεια απόσταση των κεντρών από κάθε σημείο ξεχωριστά.

Στο τρίτο στάδιο κάθε σημείο δέχεται έναν βαθμό συμμετοχής για κάθε cluster με κριτήριο την απόσταση του από αυτό (1 αν βρίσκεται πάνω στο κέντρο και όσο απομακρύνεται τείνει στο 0 .

Στο τέταρτο στάδιο μετακινούνται τα κέντρα ώστε να απέχουν όσο το δυνατόν λιγότερο από όλα τα δεδομένα και πραγματοποιείτε επόμενη επανάληψη στον υπολογισμό της απόστασης.

Τέλος, μετά το πέρας των επαναλήψεων κάθε δεδομένο σημείο διαθέτει μία πραγματική τιμή από [0,1] για κάθε cluster. Αυτή η τιμή αντιπροσωπεύει τον βαθμό συμμετοχής του κάθε σημείου ως προς κάθε cluster ξεχωριστά και ανάλογα με το πόσο απομακρυσμένο είναι από το κέντρο του έχει και μικρότερη τιμή.

1.1.β Μαθηματική ανάλυση

Στην ουσία ο αλγόριθμος στηρίζεται στην ελαχιστοποίηση μιας συνάρτησης κόστους (Dunn, 1974; Bezdek, 1981) η οποία είναι:

$$J(\mathbf{X};\mathbf{U};\mathbf{V})=\sum_{l=1}^c \sum_{k=1}^N (\mu_{lk})^m ||x_k - v_l||_A^2$$

Όπου $\mathbf{X}=[N \times n]$ πίνακας με τα δεδομένα προς ταξινόμηση. N είναι ο αριθμός των δεδομένων και n το σύνολο των δεδομένων ή πόσα μεγέθη μετράμε ανά μέτρηση.

$\mathbf{V}=[V_1, V_2, \dots, V_c]$ είναι πίνακας με τα κέντρα των cluster και

$D_{ikA}^2 = ||x_k - v_l||_A^2 = (\mathbf{x}_k - \mathbf{v}_l)^T \mathbf{A} (\mathbf{x}_k - \mathbf{v}_l)$ είναι ένας εσωτερικού γινομένου κανονικοποιητής απόστασης (ευκλείδεια απόσταση).

Τα τοπικά ακρότατα της παραπάνω συνάρτησης υπολογίζονται από την προηγούμενη συνάρτηση μετατρέποντας την μέσω των πολλαπλασιαστών Lagrange

(παράρτημα A1) και παραθέτοντας τον περιορισμό $\sum_{l=1}^c \mu_{lk}=1$ για το J .

$$J(\mathbf{X};\mathbf{U};\mathbf{V})= \sum_{l=1}^c \sum_{k=1}^N (\mu_{lk})^m D_{ikA}^2 + \sum_{k=1}^N \lambda_k (\sum_{l=1}^c \mu_{lk} - 1)$$

και από την παράγωγο του J ως προς το U, V και λ στο 0 τότε ...

αν D_{ikA}^2 για κάθε i, k και $m > 1$ που είναι ο εκθέτης βάρους. Η J ελαχιστοποιείται αν

$$\mu_{ik} = \frac{1}{\sum_{j=1}^c \left(\frac{D_{ikA}^2}{D_{jkA}^2} \right)^{\frac{2}{m-1}}} \quad (\text{ενημέρωση του πίνακα βαθμών συμμετοχής } U)$$

και

$$V_i = \frac{\sum_{k=1}^N \mu_{ik}^m x_k}{\sum_{k=1}^N \mu_{ik}^m}$$
 Όπου V_i είναι το κέντρο βάρους κάθε cluster ως προς τα υπόλοιπα σημεία, με βάρη το βαθμό συμμετοχής κάθε σημείου δηλαδή σε αυτό το στάδιο γίνεται ενημέρωση των κεντρών V .

Ο αλγόριθμος FCM κάνει τους υπολογισμούς χρησιμοποιώντας την Ευκλείδεια απόσταση καθώς ο κανονικοποιητής A είναι ο μοναδιαίος πίνακας I . Ως εκ τούτου τα clusters που εντοπίζει έχουν κοινό σχήμα και προσανατολισμό (κύκλο).

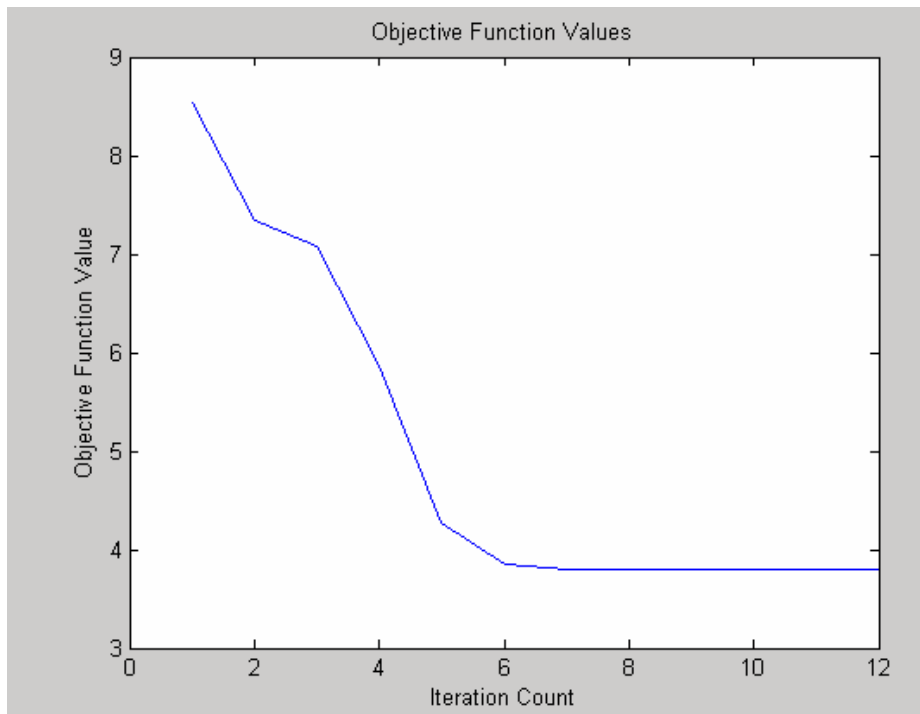
1.1.γ Matlab και παραδείγματα

Για να κατανοήσουμε καλύτερα τον αλγόριθμο προχωρούμε σε ένα παραδείγματα στο matlab ώστε να δούμε και γραφικά αποτελέσματα.

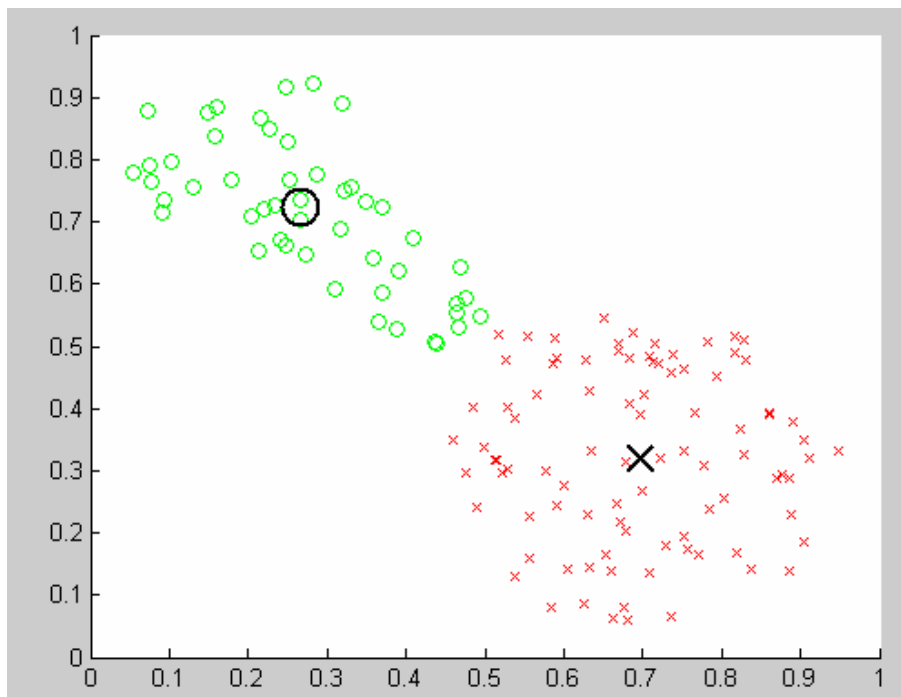
Η εντολή που χρησιμοποιείται είναι η `[center,U,objFcn] = fcm(fcndata,2);`

Όπου `center` είναι η μήτρα των κεντρών, `U` η μήτρα των βαθμών συμμετοχής και `objFcn` το αποτέλεσμα της συνάρτησης που τείνει στο ολικό ελάχιστο.

`fcndata` είναι το set δεδομένων πάνω στα οποία εφαρμόζετε η ομαδοποίηση και ο αριθμός 2 εκφράζει τον αριθμό των cluster center.



Σχήμα 1.1.γ: Αποτέλεσμα του object function ανά επανάληψη.



Σχήμα1.1.γ.2: Εδώ φαίνονται καθαρά πως ομαδοποιούνται τα δεδομένα καθώς και η θέση των κεντρών, που παρουσιάζονται με έντονο μαύρο.

1.2 Μέθοδος subtractive clustering

Η μέθοδος subtractive clustering είναι μία από του πιο διαδεδομένους αλγορίθμους ταξινόμησης καθώς είναι ιδιαίτερα αποτελεσματικός σε περιπτώσεις όπου δεν μπορεί ο χρήστης να ορίσει σε πόσα τμήματα θα διαμεριστεί ο χώρος. Άλλο ένα σημαντικό πλεονέκτημα είναι πως ανήκει στην κατηγορία no iterative αλγορίθμων πράγμα το οποίο τον κάνει ιδιαίτερα γρήγορο. Τέλος διαθέτει επεκτάσεις οι οποίες βοηθούν στην δημιουργία πολύ αξιόπιστων συστημάτων ασαφούς λογικής.

1.2.1 Λειτουργία.

Αρχικά ο αλγόριθμος υπολογίζει για κάθε data-point με βάση την ακτίνα ομαδοποίησης (cluster radius) και τα υπόλοιπα data-points που βρίσκονται σε αυτήν, πόσο κατάλληλο είναι το data-point να γίνει κέντρο ενός cluster.

Στη συνέχεια γίνεται επιλογή του σημείου που έχει την μεγαλύτερη τιμή καταλληλότητας ως το πρώτο cluster center.

Έπειτα αναθεωρήτε η καταλληλότητα όλων των υπολοίπων data-points αλλά σε σχέση με την καταλληλότητα του πρώτου cluster.

Ακολούθως γίνεται έλεγχος αν ικανοποιεί το σημείο με τη μεγαλύτερη καταλληλότητα το κριτήριο επιλογής ώστε να προαχθεί στο επόμενο cluster.

Τέλος γίνεται έλεγχος ως προς το τελευταίο κριτήριο το οποίο καθορίζει αν θα πρέπει να προαχθούν και άλλα σημεία σε cluster centers.

1.2.βΜαθηματική ανάλυση

Έστω για ένα n αριθμό μετρήσεων και m αριθμό τύπου μετρήσεων.

Αρχικά γίνεται κανονικοποίηση των δεδομένων στο διάστημα $[0,1]$

για την κανονικοποίηση χρησιμοποιείτε το παρακάτω μοντέλο :

Για $i = 1, 2, \dots, n$

Για $j = 1, 2, \dots, m$

$$X_{ij}^{norm} = \frac{X_{ij} - X_{j-min}}{X_{j-max} - X_{j-min}}$$

Έπειτα για κάθε σημείο $z_i = (x_i, y_i)$ υπολογίζεται η δυνατότητα να προαχθεί σε κέντρο P_i από τον τύπο :

$$P_i^* = \sum_{j=1}^n e^{-\frac{|x_i - x_j|^2}{r_a^2}}$$

Όπου r_a είναι μια θετική σταθερά, αντιπροσωπεύει το cluster radius και ορίζεται από τον χρήστη. Συγκεκριμένα, data-points που βρίσκονται εκτός του cluster radius ασκούν ελάχιστη επιρροή στο δυναμικό του σημείου.

Στη συνέχεια το σημείο με το μεγαλύτερο δυναμικό P_i^* προωθείτε στο πρώτο κέντρο cluster c_1 του οποίου το δυναμικό συμβολίζεται με ως $PotVal(c_1)$.

Έπειτα αναθεωρούνται τα δυναμικά όλων των υπόλοιπων σημείων αφαιρώντας από τη μήτρα P_i^* το δυναμικό του πρώτου κέντρου.

$$P_i^* = P_i^* - PotVal(c_1)\zeta$$

Όπου $\zeta = e^{-\beta|x_i - x_j|^2}$

$$\beta = \frac{4}{r_b^2}$$

$$r_b = r_a * n$$

Όπου β είναι βάρος μεταξύ του data-point και του cluster center.

Και n συμβολίζεται ο συντελεστής διαφοροποίησης (quash factor).

Έπειτα επιλέγεται το επόμενο σημείο με το μεγαλύτερο δυναμικό σαν το επόμενο cluster center και στη συνέχεια αν ικανοποιεί την παρακάτω προϋπόθεση :

$$\frac{d_{min}}{r_a} + \frac{PotVal(C_K)}{P_1^*} \geq 1 \quad \text{προάγεται.}$$

K είναι ο δίκτης του τρέχοντος σημείου που δύναται να προωθηθεί σε cluster center. Και P_1^* είναι το δυναμικό του πρώτου κέντρου cluster.

d_{min} είναι η απόσταση του πιο κοντινού cluster center από το εξεταζόμενο data point.

Αν κάποιο πιθανό cluster center δεν ικανοποιεί το παραπάνω κριτήριο τότε το δυναμικό του μηδενίζεται και στη συνέχεια ελέγχεται το αμέσως μικρότερο.

Ο αλγόριθμος σταματάει όταν ενεργοποιηθεί η παρακάτω συνθήκη :

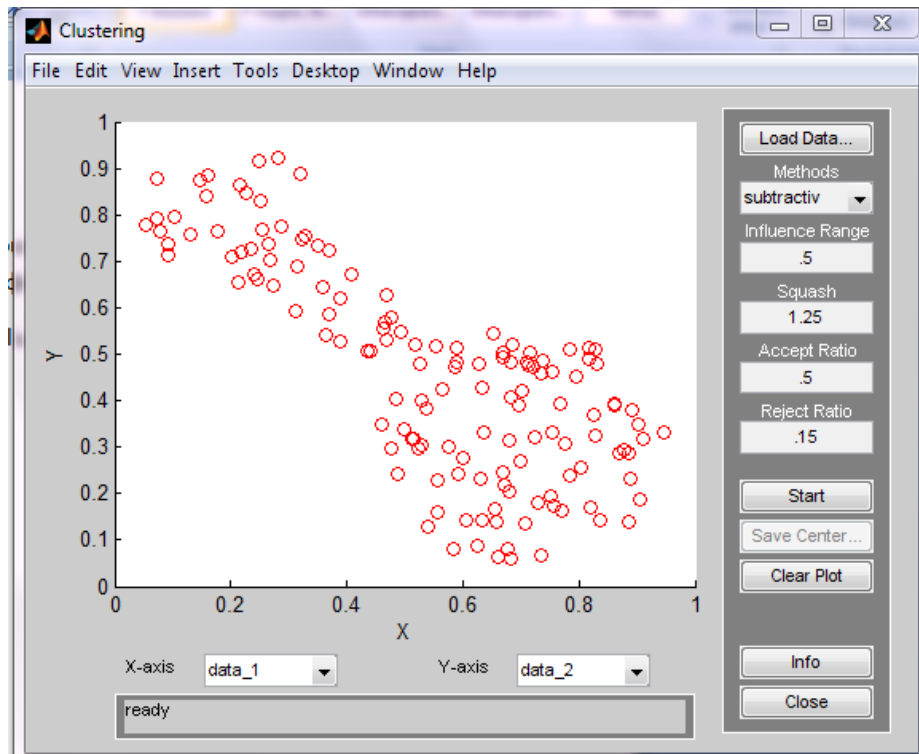
$$PotVal(C_K) < \varepsilon * P_i^*$$

Όπου ε είναι ποσοστό απόρριψης.

1.2.γ Matlab και παραδείγματα.

Στο παράδειγμα που ακολουθεί χρησιμοποιούμε το ίδιο σετ δεδομένων που χρησιμοποιήσαμε και στο προηγούμενο κεφάλαιο.

Για την ομαδοποίηση χρησιμοποιούμε το clustering GUI tool του matlab με την εντολή findcluster.



Σχήμα 1.2.1: Τα δεδομένα όπως αναπαρίστανται από το GUI

Με το πλήκτρο load data φορτώνουμε ένα σεντ δεδομένων υπό μορφή «.dat»

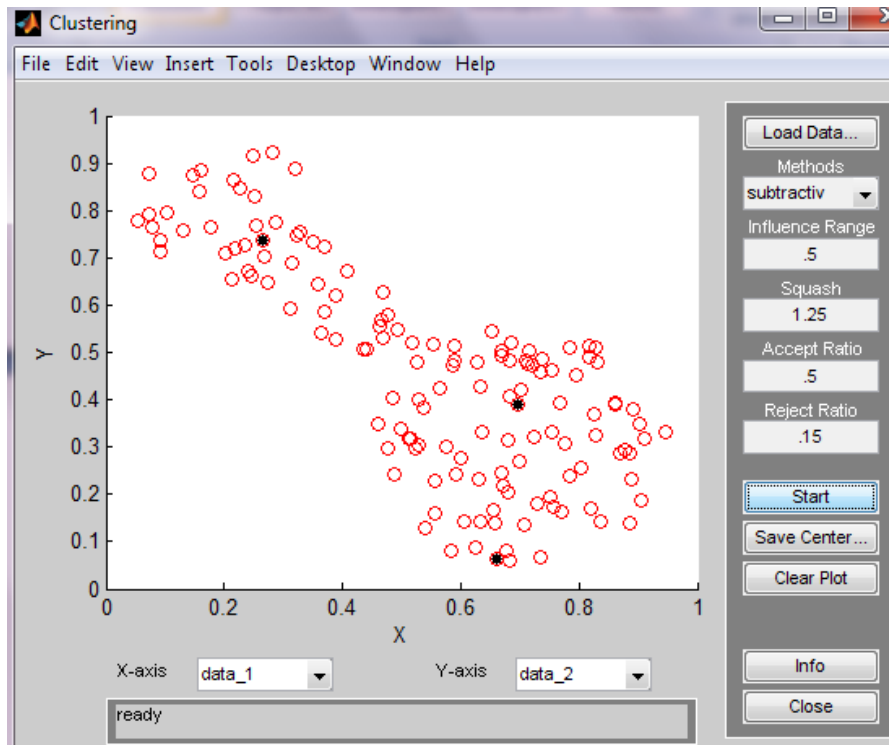
Στο πεδίο methods επιλέγεται η μέθοδος ταξινόμησης.

Στο πεδίο influence range ή αλλιώς cluster radius είναι η ακτίνα επιρροής που είδαμε και προηγουμένως.

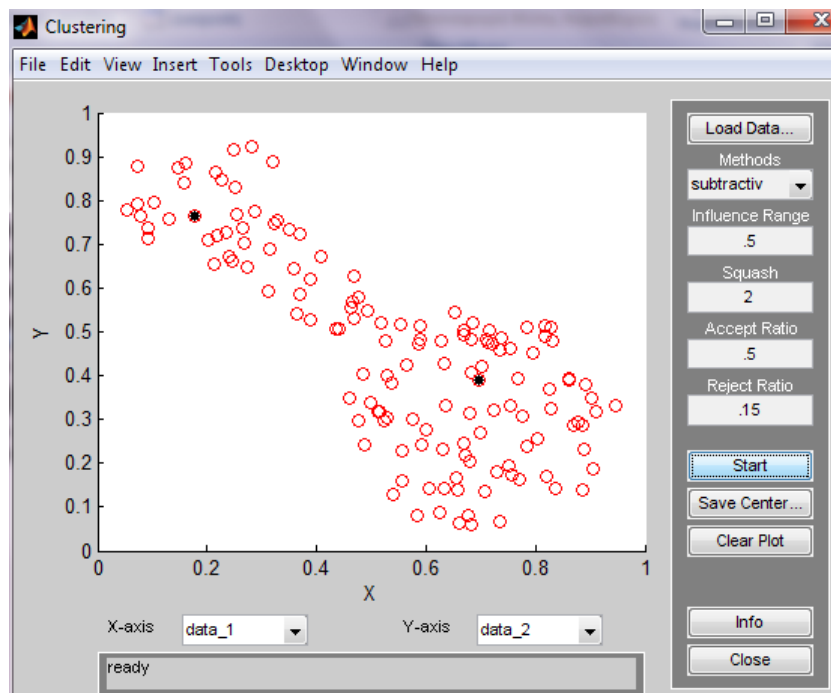
Στο πεδίο Squash είναι ο συντελεστής διαφοροποιήσεις ή quash factor που είδαμε προηγουμένως και όσο μεγαλύτερη τιμή έχει τόσο πιο απομακρυσμένα θέλουμε τα κέντρα των cluster .

Accept ratio είναι η τιμή η οποία συγκρίνεται με το κλάσμα του δυναμικού του εξεταζόμενου σημείου με του πρώτου cluster center. Αν ο λόγος τους έχει τιμή μεγαλύτερη από το accept ratio τότε το σημείο προάγεται. Το accept ratio έχει τιμές στο διάστημα $[0,1]$. (Είναι αντίστοιχο του πρώτου κριτηρίου επιλογής)

Το reject ratio, όπως είδαμε παραπάνω καθορίζει πότε θα σταματήσει ο αλγόριθμος.



Σχήμα 1.2.3: Τα κέντρα των cluster για τις προτοποθετημένες σταθερές.



Σχήμα 1.2.4 :Τα κέντρα αφού έχει μεγαλώσει το squash factor(n)

Στην εικόνα 2 έχοντας μεγαλώσει το squash factor το τρίτο cluster χάθηκε καθώς βρισκόταν πολύ κοντά στο δεύτερο cluster οπότε το δυναμικό του δεν ήταν αρκετό.

1.3 Παραλλαγή F.C.M. : Ο Αλγόριθμος Gustafson-Kessel

Οι Gustafson και Kessel το 1979 προέκτειναν τον βασικό Fuzzy C Means αλγόριθμο χρησιμοποιώντας έναν προσαρμοζόμενο κανονικοποιητή απόστασης, με σκοπό να αναγνωρίζει clusters με διαφορετική γεωμετρία στο ίδιο σετ δεδομένων.

Ο αλγόριθμος χρησιμοποιεί το \mathbf{A} σαν μήτρα κανονικοποίησης, η οποία όπως και στον κλασικό αλγόριθμο χρησιμοποιείται στον υπολογισμό της ευκλείδειας απόστασης.

$$D_{ikA}^2 = \|x_k - v_i\|_A^2 = (x_k - v_i)^T A (x_k - v_i) \quad (1)$$

Η διαφορά είναι πως η μήτρα \mathbf{A} αντί να είναι ο σταθερός μοναδιαίος πίνακας, είναι μεταβλητή και χρησιμοποιείται για βελτιστοποίηση του αλγορίθμου καθώς επιτρέπει σε κάθε cluster να αλλάζει κανονικοποιητή απόστασης (1) ανάλογα με την τοπολογία του.

Η συνάρτηση-στόχος είναι :

$$J(\mathbf{X}; \mathbf{U}; \mathbf{V}) = \sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m \|x_k - v_i\|_A^2 \quad (2)$$

Όμως η παραπάνω συνάρτηση δεν μπορεί να ελαχιστοποιηθεί ως προς το \mathbf{A} γιατί είναι γραμμική ως προς σε αυτό. Για να μπορέσουμε να καταλήξουμε σε μία λύση, το \mathbf{A} πρέπει να δεχθεί κάποιους περιορισμούς. Ο πιο συνηθισμένος τρόπος για να επιτευχθεί αυτό είναι με το να περιορίσουμε την ορίζουσα.

$$|A_i| = \rho_i \quad \rho_i > 0$$

Έτσι επιτρέποντας στην μήτρα \mathbf{A} να αλλάζει ενώ η ορίζουσα της είναι καθορισμένη, μπορούμε να αλλάζουμε το σχήμα του cluster χωρίς να αλλάζει ο όγκος του.

Μετατρέποντας τώρα με τη μέθοδο των πολλαπλασίων Lagrange το A γίνεται:

$$(3) A_i = [\rho_i \det [F_i]^{1/n} F_i^{-1}]$$

Όπου το F είναι ο πίνακας ασαφής συνδιακίμανσης του i cluster.

$$F_i = \frac{\sum_{k=1}^N (\mu_{ik})^m (z_k - v_i)(z_k - v_i)^T}{\sum_{k=1}^N (\mu_{ik})^m} \quad (4)$$

Επομένως ο αλγόριθμος αποτελείται από τα εξής βήματα, παρόμοια με την πρωτότυπη μέθοδο.

Όπου z είναι ένα σετ δεδομένων, αριθμός cluster $1 < c < N$, ο εκθέτης βάρους $m > 1$, το όριο ανοχής (termination tolerance) είναι $tol > 0$ και οι όγκοι των cluster είναι ρ_i . Η μήτρα συμμετοχής αρχικοποιείτε τυχαία, ώστε $U^{(0)} \in M_{fc}$

Συνοψίζοντας σε βήματα έχουμε :

Βήμα 1

Υπολογισμός των κεντρώων cluster:

$$V_i = \frac{\sum_{k=1}^N \mu_{ik}^m z_k}{\sum_{k=1}^N \mu_{ik}^m}, \quad 1 < i < c$$

Βήμα 2

Υπολογισμός των μητρώων συνδιακίμανσης

$$F_i = \frac{\sum_{k=1}^N (\mu_{ik})^m (z_k - v_i)(z_k - v_i)^T}{\sum_{k=1}^N (\mu_{ik})^m}$$

Βήμα 3

Υπολογισμός των αποστάσεων

$$D_{ikA}^2 = (x_k - v_i)^T [\rho_i \det [F_i]^{1/n} F_i^{-1}] (x_k - v_i)$$

$$1 < i < c, \quad 1 \leq k \leq N$$

Βήμα 4

Ενημέρωση της μήτρας συμμετοχής

for $1 \leq k \leq N$

AN $D_{ikA}^2 > 0$ για κάθε $i = 1, 2, \dots, c$

$$\mu_{ik} = \frac{1}{\sum_{j=1}^c \left(\frac{D_{ikA}^2}{D_{jkA}^2} \right)^{\frac{2}{m}-1}}$$

ΑΛΙΩΣ

AN $D_{ikA}^2 > 0$ ΤΟΤΕ $\mu_{ik}=0$ ΚΑΙ $\mu_{ik} \in [0,1]$ ΜΕ $\sum_{i=1}^c \mu_{ik}=1$

Δηλαδή εάν το σημείο έχει μηδενική απόσταση από ένα κέντρο, τότε ο βαθμός συμμετοχής του ως προς τα υπόλοιπα cluster γίνεται 0.

Όλα τα παραπάνω βήματα επαναλαμβάνονται μέχρι να ικανοποιηθεί το κριτήριο

$||U^{(l)} - U^{(l-1)}|| < tol$ l δείχνει σε ποια επανάληψη βρίσκεται ο αλγόριθμος. Επίσης σε όλα τα βήματα χρησιμοποιούμε τους βαθμούς συμμετοχής από τη προηγούμενη επανάληψη εκτός βέβαια από το βήμα 4 όπου γίνεται η ενημέρωση.

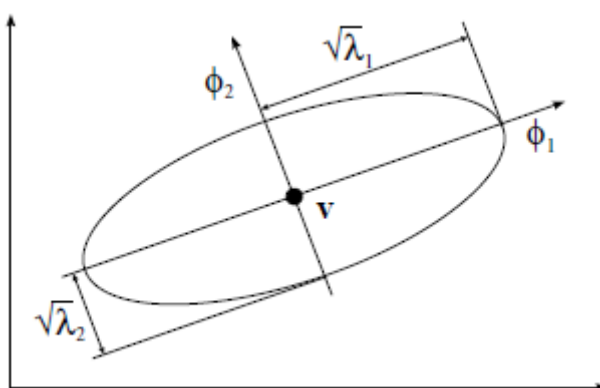
1.3.2 Παράμετροι

Όπως είδαμε και πριν, σχεδόν όλοι οι παράμετροι είναι περίπου ίδιοι με τον αλγόριθμο FCM. Επιπλέον παράμετρος είναι ο όγκος των cluster ρ_i όπου χωρίς να έχουμε προηγούμενη γνώση μπορούμε να θέσουμε την τιμή του ως 1. Επειδή όμως χρησιμοποιείτε σαν περιορισμός στην (3) το ελάττωμα που παρουσιάζεται από αυτή τη ρύθμιση είναι πως εντοπίζει σχεδόν ίδια cluster.

1.3.3 Ερμηνεία του πίνακα συνδιακίμανσης.

Η ιδιοδομή του πίνακα συνδιακίμανσης μας προσφέρει πληροφορίες σχετικά με τη δομή και τον προσανατολισμό των cluster . Οι κατευθύνσεις των αξόνων δίνονται από τις ιδιοτιμές του πίνακα F_i .

Το ποσοστό από τα μήκη των αξόνων δίνεται από το ποσοστό της τετραγωνικής ρίζας των ιδιοτιμών.



Σχήμα 1.3.1 : Γραφική παράσταση της συνάρτησης $(\mathbf{Z}-\mathbf{V})\mathbf{F}^{-1}(\mathbf{X}-\mathbf{V})=1$.

ϕ_1 και ϕ_2 είναι τα ιδιοδιανύσματα, \mathbf{V} το κέντρο του cluster και λ_1, λ_2 ιδιοτιμές.

Το ιδιοδιάνισμα που αντιστοιχεί στην μικρότερη ιδιοτιμή είναι η μεσοκάθετος της ελλειπτικής συνάρτησης και δείχνει τον προσανατολισμό του cluster.

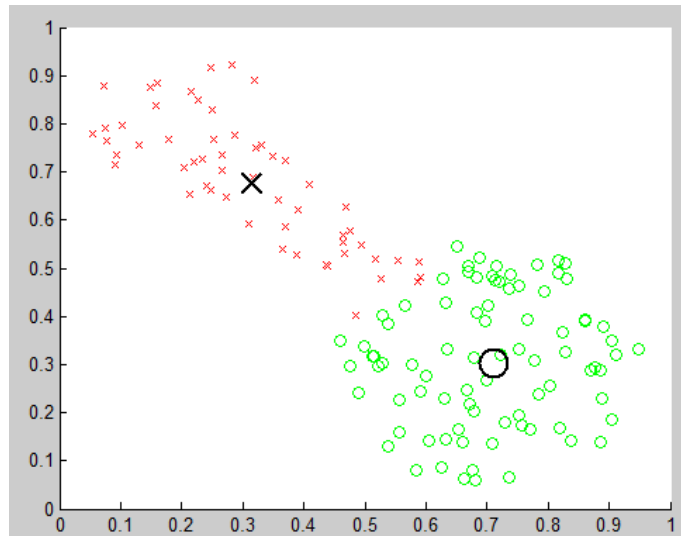
1.3.5 Matlab και παραδείγματα.

Στο παράδειγμα που ακολουθεί χρησιμοποιούμε το ίδιο σετ δεδομένων με προηγουμένως.

Όπου $c=2$ αριθμός cluster

$m=2$ εκθέτης βάρους

Ο αλγόριθμος βρίσκεται στο παράρτημα ()



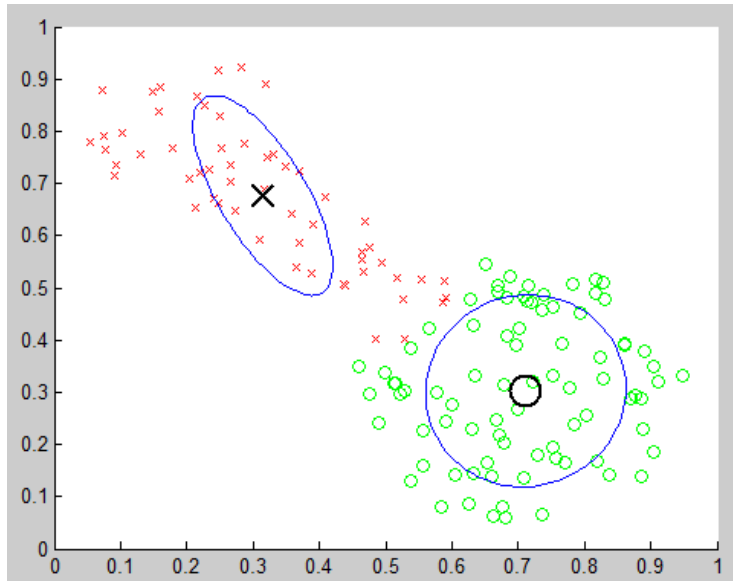
Σχήμα 1.3.2: Εδώ φαίνονται τα 2 cluster που εντόπισε ο αλγόριθμος.

Από τον πίνακα F ο οποίος έχει διαστάσεις 2×2 για κάθε cluster (είναι στην ουσία ένας $2 \times 2 \times 2$ πίνακας) υπολογίζουμε τις ιδιοτιμές και τα ιδιοδιανίσματα.

(X) CLUSTER	$\lambda_1=0.0048$	$\theta_1=0.6655$
	$\lambda_2=0.0429$	
(O)CLUSTER	$\lambda_1=0.0124$	$\theta_1 = 0.9443$
	$\lambda_2=0.0184$	

Από τις ιδιοτιμές βλέπουμε πως το (O)CLUSTER είναι περίπου σφαιρικό ενώ από τη διαφορά των ιδιοτιμών το (X)CLUSTER είναι ελλειπτικό.

Το ιδιοδιάνυσμα που αντιστοιχεί στην ελάχιστη ιδιοτιμή για κάθε cluster εκφράζει τη μεσοκάθετο και ως εκ τούτου τον προσανατολισμό του.



Σχήμα 1.4.4: Εδώ φαίνεται η γεωμετρία των cluster με βάση τις τιμές του πίνακα (1)

1.4 Μέθοδος Self Organized Maps (SOM) –Αυτοπροσαρμοζόμενοι Χάρτες Kohonen

Ο συγκεκριμένος αλγόριθμος χρησιμοποιεί έναν δυσδιάστατο «χάρτη» από νευρώνες, οι οποίοι εκπαιδεύονται και μετακινούνται ώστε να καταλάβουν μία θέση στο χώρο που να ικανοποιεί όσο το δυνατόν περισσότερες μετρήσεις.

1.4.1 Λειτουργία

Αρχικά ορίζεται ο αριθμός των βαρών υπό μορφή πίνακα.

Κάθε νευρώνας διαθέτει αριθμό βαρών αντίστοιχου του αριθμού του τύπου των δεδομένων ανά μέτρηση. Επομένως για k τύπους δεδομένων ο κάθε νευρώνας διαθέτει k εισόδους με τα αντίστοιχα βάρη τους.

Έπειτα στην διαδικασία επιλογής οι νευρώνες στο δίκτυο Kohonen λειτουργούν ανταγωνιστικά. Συγκεκριμένα, μόλις εξετάζετε ένα set εισόδων εντοπίζεται ο νευρώνας ο οποίος βρίσκεται πιο «κοντά» του από τον υπολογισμό της ευκλείδειας απόστασης και ανακηρύσσεται νικητής.

Στη συνέχεια ακολουθεί η εκπαίδευση του νευρώνα.

Συνηθίζεται να εκπαιδεύονται και οι νευρώνες μέσα σε μία ακτίνα σ_0 γύρω από τον νικητή. Η οποία μειώνεται εκθετικά με την πάροδο των επαναλήψεων.

Στη συνέχεια ακολουθεί η ενημέρωση των βαρών του νευρώνα ως προς τη διαφορά του με την εξεταζόμενη είσοδο. Εδώ γίνεται χρήση μιας μεταβλητής $L(n)$ η οποία με την πάροδο των επαναλήψεων μειώνεται.

Για την περίπτωση που ο αλγόριθμος εκπαιδεύει και τους γείτονες νευρώνες τότε καλό θα είναι να μην εκπαιδεύονται όλοι το ίδιο. Οπότε στη διαδικασία της ενημέρωσης λαμβάνεται υπ όψιν και μία τελευταία μεταβλητή $\theta(n)$ που επηρεάζει την εκμάθηση του νευρώνα ανάλογα με το πόσο μακριά βρίσκεται από τον νικητή.

1.4.2 Μαθηματική ανάλυση

Ο αριθμός των νευρώνων, επειδή αποτελούν δυδιάστατο πλέγμα ορίζεται υπό μορφή πίνακα. Για $nw = (nw1 * nw2)$, όπου nw ο αριθμός των νευρώνων.

Αρχικά γίνεται η αρχικοποίηση των βαρών των νευρώνων συνήθως με τυχαίες τιμές.

Έπειτα εξετάζονται ένα-ένα όλα τα σετ εισόδων ως προς όλους τους νευρώνες και επιλέγεται ο νικητής. Με βάση την απόσταση $dist$.

$$dist = \sqrt{\sum_{i=1}^n (V_i - W_i)^2}$$

Όπου n αριθμός μετρήσεων-νευρώνων, V_i η τιμή της επιλεγμένης εισόδου για τον συγκεκριμένο τύπο δεδομένων (i) και W_i το βάρος του νευρώνα που απευθύνεται στον συγκεκριμένο τύπο εισόδου. Ο πιο κοντινός νευρώνας είναι ο νικητής.

Στη συνέχεια γίνεται ο υπολογισμός της ακτίνας επιρροής σ_τ

$$\sigma_\tau = \sigma_0 \exp\left(-\frac{\tau}{\lambda}\right), \quad \tau=1,2,3,\dots$$

Όπου σ_0 είναι αρχική τιμή της ακτίνας. Το τ αντιπροσωπεύει την τρέχων επανάληψη. Και λ είναι μία σταθερά χρόνου εξαρτάται από την τιμή της ακτίνας σ και από τον αριθμό των επαναλήψεων.

$$\lambda = \text{αριθμός επαναλήψεων} / \log(\sigma)$$

Στη συνέχεια η ενημέρωση των βαρών των νευρώνων πραγματοποιείται βάση της παρακάτω εξίσωσης.

$$W_{\tau+1} = W_\tau + L_\tau (V_\tau - W_\tau)$$

Στη περίπτωση που εκπαιδεύουμε και τους γύρο νευρώνες, υπολογίζεται και ο συντελεστής L_τ

$$W_{\tau+1} = W_\tau + \Theta_\tau L_\tau (V_\tau - W_\tau)$$

Το L_τ είναι ο ρυθμός εκμάθησης άλλες φορές μπορεί να είναι μια σταθερά αλλά μπορεί και να μειώνεται με την πάροδο των επαναλήψεων.

$$L_\tau = L_0 \exp\left(-\frac{\tau}{\lambda}\right)$$

Όπου L_0 η αρχική τιμή του ρυθμού εκμάθησης.

Το θ_τ δίνεται από την εξίσωση:

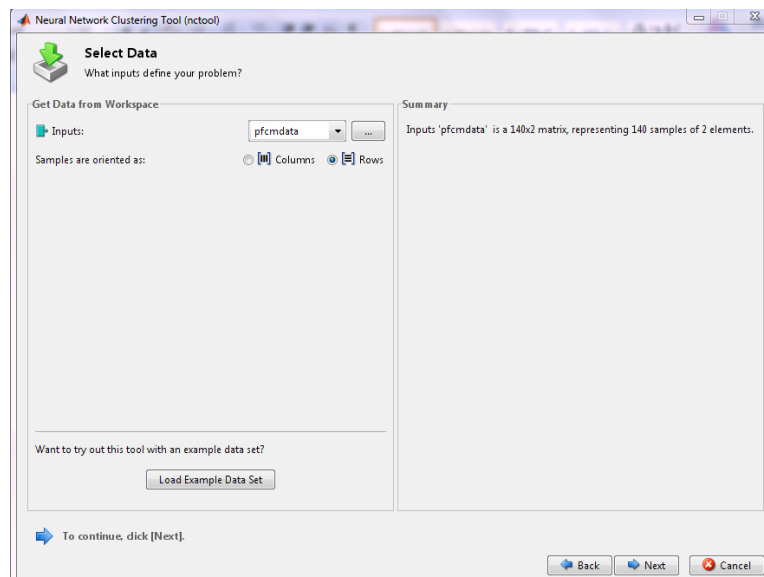
$$\theta_\tau = \exp\left(-\frac{\delta^2}{2\sigma_\tau^2}\right)$$

Όπου δ είναι η απόσταση του νευρώνα από τον νικητή.

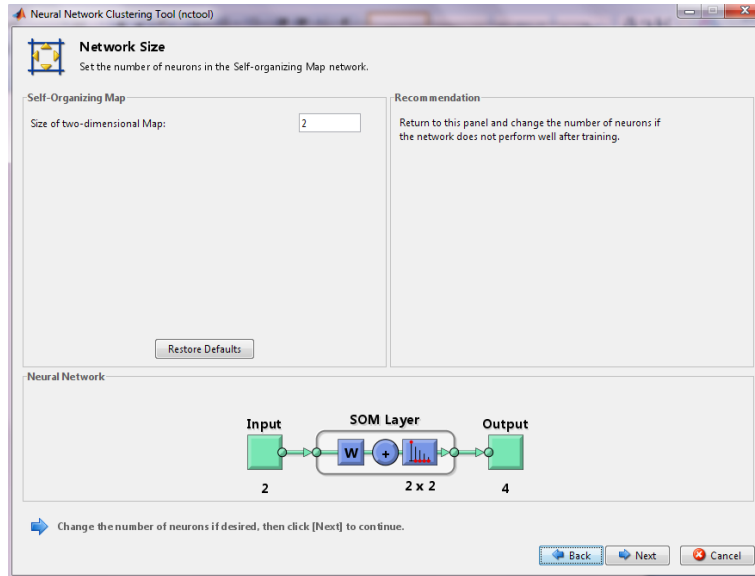
Εδώ παρατηρούμε πως ο συντελεστής θ_τ εφόσον επηρεάζεται άμεσα από την ακτίνα, η οποία με τη σειρά της μικραίνει με την πάροδο των επαναλήψεων, συμπεραίνουμε πως και το θ_τ κάθε φορά μικραίνει.

1.4.3 Matlab και παραδείγματα

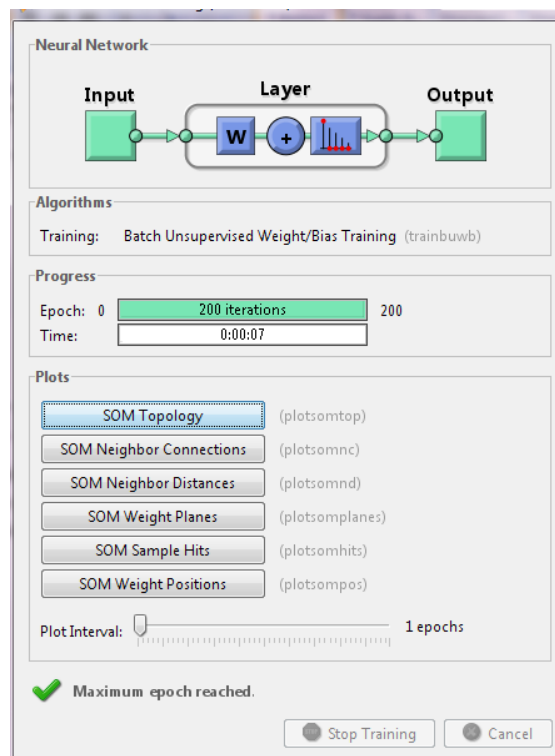
Στο matlab χρησιμοποιούμε το GUI `nctool`, στο οποίο δίνουμε τα ίδια δεδομένα εισόδου με τα προηγούμενα παραδείγματα.



Σχήμα 1.4.1 : Η αρχική σελίδα του `nctool` όπου στο πεδίο `inputs` ορίζουμε τα δεδομένα πάνω στα οποία θα εκπαιδευτεί το νευρωνικό πλέγμα.



Σχήμα 1.4.2 : Στην επόμενη σελίδα ορίζουμε τον αριθμό των νευρώνων υπό μορφή δυσδιάστατου πίνακα, στο πεδίο “size of two-dimensional map”. Δηλαδή για τιμή 2 έχουμε έναν 2×2 πίνακα άρα 4 νευρώνες.

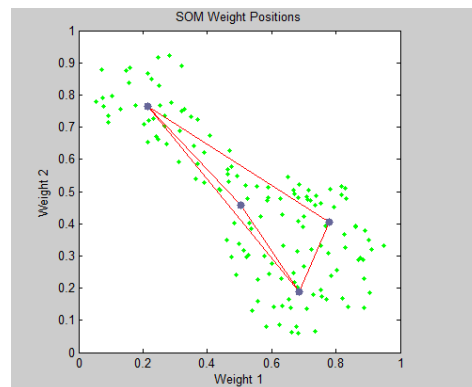
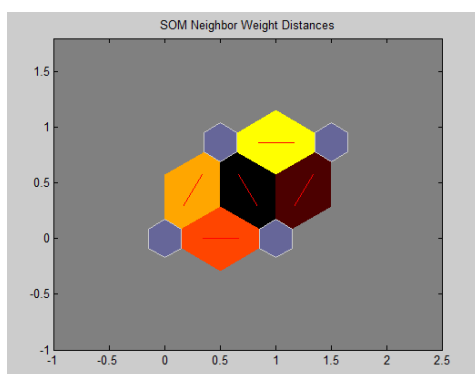


Σχήμα 1.4.3 : Το πλήκτρο SOM Topology εμφανίζει τη τοπολογία του χάρτη. Το πλήκτρο Neighbor Connections εμφανίζει σε διάγραμμα πως είναι δομημένες οι γειτονιές.

Το πλήκτρο Weight planes εμφανίζει τόσα διαγράμματα όσα και ο αριθμός των εισόδων που δέχεται το πλέγμα. Δείχνει για κάθε είσοδο ξεχωριστά πόσες συνδέσεις είχαν με τα βάρη των νευρώνων. Το χρώμα στον κάθε νευρώνα όσο πιο ανοιχτό είναι τόσες περισσότερες συνδέσεις είχαν τα βάρη του με τις εισόδους.

Το πλήκτρο sample hits δείχνει πόσες φορές νίκησε ο κάθε νευρώνας.

Τέλος τα δύο πιο σημαντικά διαγράμματα, το Weight positions που εμφανίζει τα δεδομένα και την εκπαιδευμένη θέση των νευρώνων, και το Neighbor distances που δείχνει την απόσταση του κάθε νευρώνα από τους γύρω του.



Στο διάγραμμα των αποστάσεων όσο πιο σκουρόχρωμη είναι η γραμμή τόσο πιο απομακρυσμένοι είναι οι νευρώνες. Άρα παρατηρούμε πως ο κάτω δεξιά νευρώνας του πρώτου διαγράμματος, στο διάγραμμα των θέσεων είναι αυτός που βρίσκεται πιο ψηλά. Αντίστοιχα ο πάνω αριστερά του διαγράμματος αποστάσεων είναι ο χαμηλότερος νευρώνας καθώς συνδέεται με άλλους τρεις εκ των οποίων οι δύο είναι κοντά και ο τρίτος μακριά.

1.5 Συμπεράσματα και παρατηρήσεις

Έχοντας δει τα χαρακτηριστικά των παραπάνω αλγορίθμων καταλήγουμε στα εξής συμπεράσματα. Ο πιο γρήγορος από τους παραπάνω αλγορίθμους μπορούμε να πούμε πως είναι η μέθοδος Subtractive clustering καθώς είναι ο μόνος που είναι Non iterative και καταναλώνει τη λιγότερη υπολογιστική ισχύ. Από την άλλη, η παραλλαγή Gustafson Kessel του αλγορίθμου fcm μπορεί να εκτελεί τους περισσότερους υπολογισμούς αλλά μπορούμε να πούμε πως παράγει καλύτερα αποτελέσματα λόγω της ευέλικτης γεωμετρίας των cluster. Η μέθοδος Som μας προσφέρει περισσότερες πληροφορίες για τη θέση των cluster means.

ΚΕΦΑΛΑΙΟ 2: ΣΥΣΤΗΜΑΤΑ ΑΣΑΦΟΥΣ ΛΟΓΙΚΗΣ

2.1 ΕΙΣΑΓΩΓΗ

Τα συστήματα ασαφούς λογικής ανήκουν στην κατηγορία των ευφυών συστημάτων και βρίσκουν εφαρμογή σε πολλά πρακτικά προβλήματα.

Όπως φαίνεται και από το τίτλο ένα τέτοιο σύστημα συνδυάζει την γνώση είτε είναι ποσοτική (μέτρηση) είτε είναι γλωσσική (πολύ- λίγο κλπ) με έναν ασαφή μηχανισμό συμπεράσματος, ο οποίος με την ασαφή λογική ξεπερνάει τα πλαίσια της στενής λογικής και έρχεται πιο κοντά στον ανθρώπινο συλλογισμό.

Για να παρουσιάσουμε ακριβώς την λειτουργία ενός συστήματος ασαφούς λογικής πρέπει πρώτα να διευκρινίσουμε τις έννοιες ασαφούς συνόλου και της συνάρτησης συμμετοχής.

2.2 Ασαφές σύνολο

Λέγοντας ασαφές σύνολο περιγράφουμε ένα πλήθος τιμών κατανεμημένες σε ένα σύνολο με βάση ένα κοινό χαρακτηριστικό. Στη συνέχεια ανάλογα με το πόσο πλησιάζει σε αυτό το χαρακτηριστικό η κάθε τιμή του συνόλου παίρνει μία τιμή στο διάστημα $[0,1]$ που ονομάζεται βαθμός συμμετοχής.

Οπότε έστω ότι το υπερσύνολο αναφοράς U περιλαμβάνει τους αριθμούς 3,4,5,6,7 και 8. Στο U ορίζουμε το κλασικό-σαφές μονοσύνολο $A=\{1/5\}$ το οποίο περιλαμβάνει τον αριθμό 5 με βαθμό συμμετοχής 1. Εάν χρησιμοποιήσουμε την έκφραση «οι ακέραιοι αριθμοί κοντά στο 5» τότε μπορούμε να ορίσουμε ένα ασαφές σύνολο A το οποίο να περιλαμβάνει και τους ακέραιους αριθμούς 3,4,6,7 που είναι κοντά στο 5 με βαθμούς συμμετοχής μικρότερους από το 1.

$$A = \{0.5/3 + 0.9/4 + 1/5 + 0.9/6 + 0.3/7 + 0/8\}$$

Το ασαφές σύνολο A μπορεί να θεωρηθεί ως ένας ασαφής αριθμός (fuzzy number) που περιγράφεται με τη γλωσσική τιμή «περίπου 5».

Στο προηγούμενο παράδειγμα ο βαθμός συμμετοχής δόθηκε καθαρά από τον άνθρωπο. Σε πολλές εφαρμογές όμως είναι απαραίτητο ο βαθμός συμμετοχής να μοιράζεται με διαφορετική ένταση ή πολύ απλά να αποτελεί κομμάτι ενός αλγορίθμου. Γι αυτό τον λόγο ορίζονται οι συναρτήσεις συμμετοχής. Η συνάρτηση συμμετοχής $(\mu(x))$ αντιστοιχεί σε κάθε στοιχείο του υπερσυνόλου A έναν βαθμό συμμετοχής. Επομένως το A γράφεται ως :

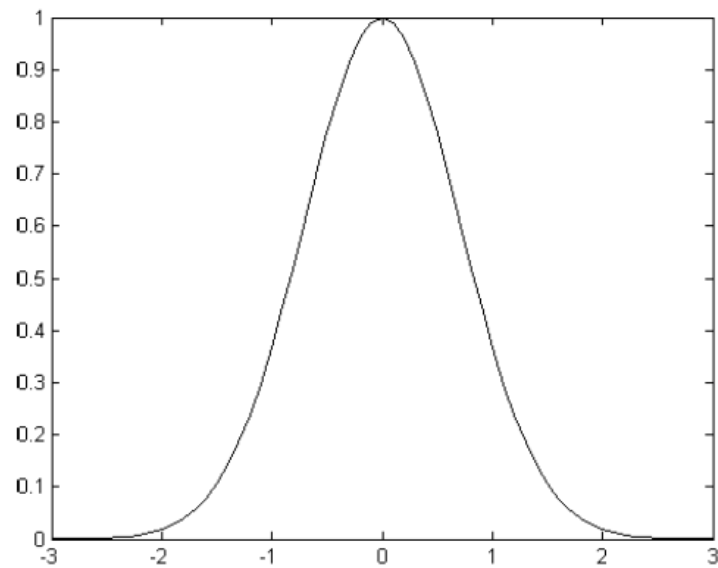
$$A = \left\{ \frac{\mu_A(x_1)}{x_1} + \frac{\mu_A(x_2)}{x_2} + \dots \right\}$$

Υπάρχουν πολλά είδη συναρτήσεων συμμετοχής. Στη παρούσα εργασία θα παρουσιαστεί η γκαουσιανή συνάρτηση συμμετοχής καθώς χρησιμοποιείται ευρέως σε συνδυασμό με την μεθοδολογία ομαδοποίησης Subtractive clustering που εφαρμόζεται στην συνάρτηση genfis2.m του Matlab.

Η γκαουσιανή συνάρτηση συμμετοχής ορίζεται από 2 παραμέτρους (σ, c).

$$\text{Gaussian}(x; \sigma, c) = e^{-\left[\frac{(x-c)}{\sigma}\right]^2}$$

Όπου c είναι το κέντρο της συνάρτησης συμμετοχής και το σ καθορίζει το εύρος της. Ένα παράδειγμα γκαουσιανής συνάρτησης με $c=0$ και τυπική απόκλιση $\sigma=1$.



Εικόνα 2.1: Γκαουσιανή ή ακτινική συνάρτηση ($y = e^{-x^2}$)

Είναι φανερό πως οι τιμές που βρίσκονται κοντά στο κέντρο έχουν βαθμό κοντά στο 1 και όσο απομακρύνονται ο βαθμός συμμετοχής πλησιάζει στο 0. Μπορούμε να πούμε πως το παράδειγμα εξυπηρετεί πολύ καλά ένα ασαφές σύνολο «Κοντά στο 0»

2.3 Η έννοια του κανόνα.

Τα συστήματα ασαφούς λογικής βασίζονται σε γνώση όχι μόνο στην σύνθεση των συναρτήσεων συμμετοχής αλλά και στον ορισμό των ασαφών κανόνων που δημιουργούν το συμπέρασμα. Ένα παράδειγμα ασαφούς κανόνα είναι :

Εάν η ταχύτητα του αυτοκινήτου είναι μεγάλη Τότε εφαρμόσε μικρή επιτάχυνση (1)

Όπου οι λέξεις μεγάλη και μικρή μοντελοποιούνται με ασαφή σύνολα.

Οι τύποι των συστημάτων ασαφούς λογικής είναι τρεις: 1) θεωρητικά ασαφή συστήματα, 2) Tagaki-Sugeno-Kang ασαφή συστήματα και 3) ασαφή συστήματα με ασαφοποιητή και αποασαφοποιητή. Ανάλογα με τον τύπο του συστήματος διαφέρει ο τρόπος σύλληψης του συμπεράσματος καθώς αλλάζει η σχέση μεταξύ του «**Εάν**» και του «**Τότε**» καθώς και η συσχέτιση περισσότερων κανόνων μεταξύ τους. Στην παρούσα εργασία γίνεται ανάλυση των συστημάτων Tagaki-Sugeno-Kang εφόσον χρησιμοποιούνται αποκλειστικά στις εφαρμογές.

2.4 Ασαφή συστήματα τύπου Sugeno

Η βασική διαφορά της μεθόδου Sugeno με τους υπόλοιπους τύπους συστημάτων ασαφούς λογικής είναι πως δεν υπάρχει το στάδιο της αποασαφοποίησης καθώς η έξοδος του κάθε κανόνα είναι είτε μία τιμή (0ης τάξης Sugeno) ή ένα πολυώνυμο n βαθμού (n- τάξης Sugeno).

Οι κανόνες τώρα είναι της μορφής :

Εάν η ταχύτητα x του αυτοκινήτου είναι μεγάλη **Τότε** η επιτάχυνση είναι $y=cx$ (2)

όπου c είναι μια σταθερά. Παρατηρούμε ότι το αποτέλεσμα του κανόνα τώρα είναι μια μαθηματική σχέση και δεν περιέχει ανθρώπινη γνώση. Ένα άλλο πρόβλημα με τους TSK είναι ότι δεν υπάρχει η δυνατότητα εφαρμογής διαφορετικών μεθοδολογιών ασαφούς συλλογισμού. Στον αντίποδα αυτών το TSK συνδυάζει ευκολότερα τους κανόνες.

Συγκεκριμένα ένας κανόνας για ένα σύστημα το οποίο περιέχει περισσότερες από μία εισόδους και κανόνες γράφεται ως:

Εάν x_1 είναι $\mu(\chi_1)_i^1$ και x_2 είναι $\mu(\chi_2)_i^2$ και..... και x_n είναι $\mu(\chi_n)_i^n$ **Τότε** $z = f(x_1, x_2, \dots, x_n)$

Ανάλογα με τον βαθμό της εξίσωσης z το σύστημα χαρακτηρίζεται ως 1^ο βαθμού 2^ο κλπ.

2.4.α Συστήματα T.S.K. μηδενικού βαθμού.

Στη περίπτωση που η εξίσωση της εξόδου του κανόνα (z) είναι ένα μηδενικής τάξης πολυώνυμο δηλαδή μια σταθερά οι κανόνες είναι της μορφής :

Εάν x_1 είναι $\mu(\chi_1)_i^1$ και x_2 είναι $\mu(\chi_2)_i^2$ και..... και x_n είναι $\mu(\chi_n)_i^n$ **Τότε** $z = c$

Για τον υπολογισμό του βαθμού πυροδότησης στον όρο και χρησιμοποιούμε τους τελεστές (min-product) π.χ. : $\min(\mu(\chi_1)_i^1, \mu(\chi_2)_i^2) = w_i$

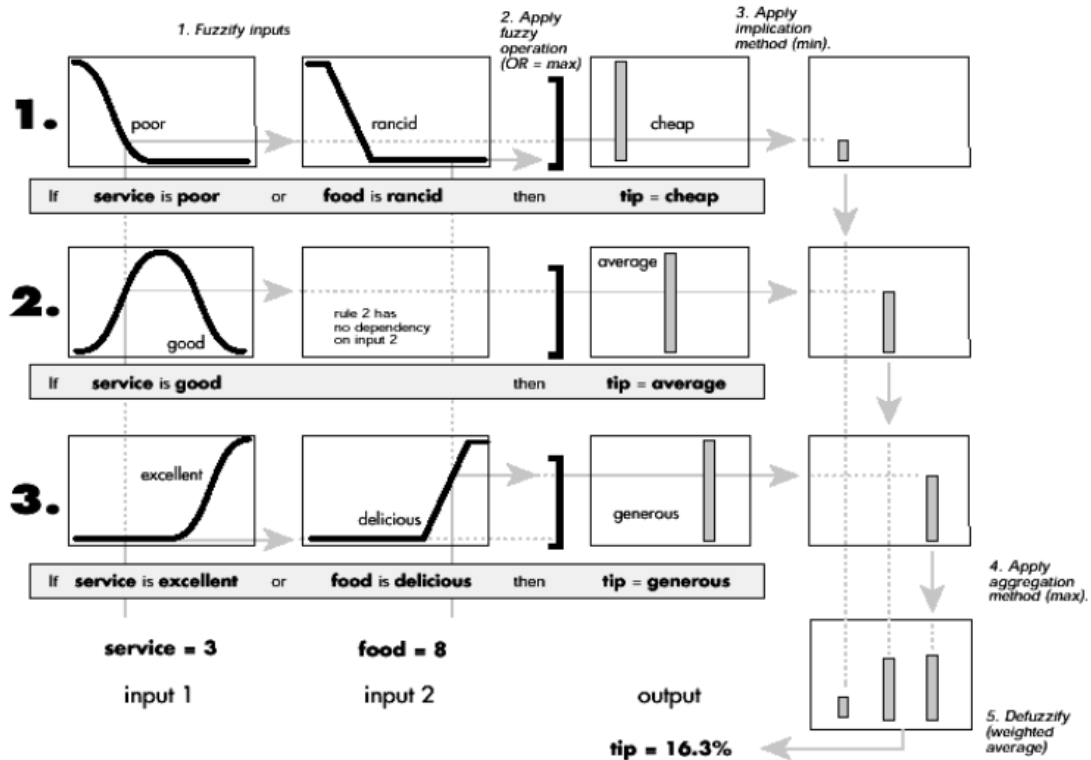
Στη συνέχεια το αποτέλεσμα της λογικής πράξης πολλαπλασιάζεται με το συμπέρασμα του κανόνα και δίνει το αποτέλεσμα του κανόνα. ($w_i y^i$)

Η ίδια διαδικασία επαναλαμβάνεται για κάθε κανόνα και στο τέλος εφαρμόζεται η μέθοδος των σταθμισμένων μέσων όρων για το αποασαφοποιημένο αποτέλεσμα.

$$y = \frac{w_1 z^1 + w_2 z^2 + \dots + w_k z^k}{w_1 + w_2 + \dots + w_k}$$

Όπου k ο αριθμός των κανόνων .

Στη συνέχεια ακολουθεί ένα απλό παράδειγμα τροποποιημένο για χρήση ενός συστήματος Sugeno μηδενικού βαθμού. Οι δύο είσοδοι του μοντέλου είναι η εξυπηρέτηση (service) σε ένα εστιατόριο και η ποιότητα του φαγητού (food) ενώ η σταθερά της εξόδου αντιπροσωπεύει το φιλοδώρημα (tip). Βέβαια εδώ οι κανόνες περιέχουν **min** και όχι **product** όπως δήξαμε προηγουμένως οπότε για τον υπολογισμό του βαθμού πυροδότησης αντί για τον τελεστή min χρησιμοποιούμε τον max.



Σχήμα2.2: Ασαφής συλλογισμός τύπου Sugeno (0ης τάξης).

Εδώ παρατηρούμε πως το σύστημα είναι παρόμοιο με το ασαφή σύστημα τύπου Mamdani εάν το αποτέλεσμα του τελευταίου θεωρηθεί ασαφές μονοσύνολο (singleton). Οι μόνες διαφορές είναι πως στο σύστημα Mamdani για την διεξαγωγή του βαθμού πυροδότησης δεν είμαστε υποχρεωμένοι να χρησιμοποιήσουμε τους τελεστές min-max και πως η έξοδος του συμπεράσματος του Mamdani μπορεί να είναι και συνάρτηση συμμετοχής

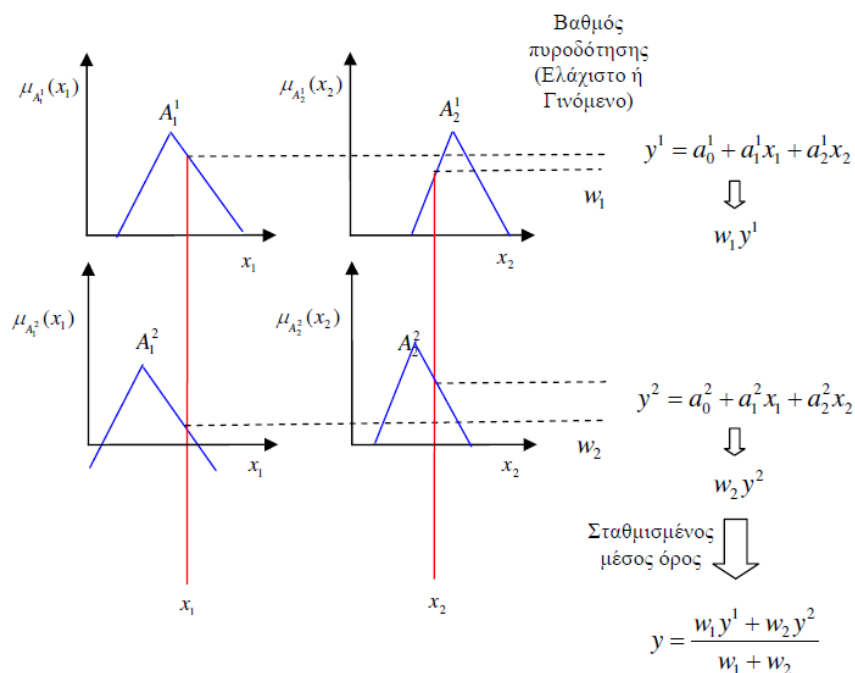
2.4.β Συστήματα T.S.K. n βαθμού.

Η μορφή των κανόνων ενός συστήματος Sugeno είναι :

Εάν x_1 είναι $\mu(\chi_1)_i^1$ και x_2 είναι $\mu(\chi_2)_i^2$ και..... και x_n είναι $\mu(\chi_n)_i^n$ **Τότε**
 $z^i = c^i + a_1^i x_1 + a_2^i x_2 + \dots + a_n^i x_n$

Στον υπολογισμό του βαθμού πυροδότησης χρησιμοποιείτε πάλι ο τελεστής min-max καθώς και στον υπολογισμό του αποασαφοποιημένου αποτελέσματος χρησιμοποιείτε πάλι ο σταθμισμένος μέσος όρος . Στη πράξη μερικές φορές για απλούστευση των υπολογισμών, το αποτέλεσμα υπολογίζεται με τη χρήση του weighted sum ($y = w_1 y^1 + w_2 y^2 + \dots + w_k y^k$) έναντι του σταθμισμένου μέσου όρου . Παρόλα αυτά , αυτή η απλούστευση μπορεί να οδηγήσει στην απόλυτα των λεκτικών σημασιών των συναρτήσεων συμμετοχής , εκτός και αν το άθροισμα των βαθμών πυροδότησης ($\sum w_i$) πλησιάζει το 1 .

Σ το παρακάτω σχήμα παρουσιάζεται ένας απλός συλλογισμός Sugeno 1^{ου} βαθμού με 2 εισόδους και 2 κανόνες.



Σχήμα 2.2: Ασαφής συλλογισμός τύπου Sugeno (1ης τάξης).

2.5 Συμπεράσματα και παρατηρήσεις

Η μέθοδος ασαφούς συλλογισμού Sugeno ονομάζεται **συναρτησιακός συλλογισμός** ή **συναρτησιακό ασαφές σύστημα** είναι πολύ χρήσιμη μεθοδολογία για την αντιμετώπιση πολύπλοκων προβλημάτων.

Τα πλεονεκτήματα του συλλογισμού Sugeno είναι:

- 1) Χρησιμοποιείται ευρέως για ασαφή προτυποποίηση ενός συστήματος από δεδομένα εισόδου-εξόδου. Είναι σημαντική η προσφορά της μεθόδου Sugeno στη δραστική μείωση των κανόνων που απαιτούνται για τη μοντελοποίηση του συστήματος.
- 2) Η αποφυγή της αποασαφοποίησης αφού αυτή η μέθοδος για τον υπολογισμό της τελικής εξόδου χρησιμοποιεί τη μέθοδο του σταθμισμένου μέσου όρου.
- 3) Δίνει σχεδόν τα ίδια αποτελέσματα και με τις κλασικές μηχανές ασαφούς συμπεράσματος .
- 4) Στην πράξη εφαρμόζεται εύκολα, είναι πολύ γρήγορη και χρειάζεται μικρό υπολογιστικό φορτίο και λίγο χρόνο στη φάση της εκπαίδευσης του ασαφούς συστήματος.
- 5) Επειδή η έξοδος του κάθε κανόνα αποτελεί γραμμική συνάρτηση των συναρτήσεων συμμετοχής των εισόδων οπότε μπορεί να λειτουργήσει πολύ καλά ως επιτηρητής για έναν γραμμικό ελεγκτή (π.χ. PID)
- 6) Επειδή είναι πιο πρακτικό και αποδοτικό από το σύστημα του Mamdani χρησιμοποιείτε από πολλές μεθόδους βελτιστοποίησης για τη δημιουργία ασαφών μοντέλων που να ικανοποιούν καλύτερα το σετ των δεδομένων.

Τα μειονεκτήματα του ασαφούς συλλογισμού Sugeno είναι:

- 1) Αντίθετα από τον συλλογισμό Mamdani, στο συλλογισμό Sugeno δεν εφαρμόζεται ο συνθετικός κανόνας συμπεράσματος . Όταν η είσοδος είναι ασαφής τότε μπορούμε να χρησιμοποιήσουμε τη μέθοδο του ταιριάσματος των ασαφών συνόλων και του υπολογισμού του βαθμού συμβατότητας των δύο γεγονότων.
- 2) Στον ασαφή συλλογισμό τύπου Sugeno η ανθρώπινη γνώση δεν μπορεί να ενσωματωθεί όπως στον ασαφή συλλογισμό του τύπου Mamdani.

Κεφάλαιο 3 : Εφαρμογές σε δεδομένα ρυζιού και PMV

Εισαγωγή .

Στην παρούσα ενότητα με τη μέθοδο subtractive clustering δημιουργείται ένα σύστημα ασαφούς λογικής το οποίο εκπαιδεύεται στην πρώτη περίπτωση σε δεδομένα από δημοσκόπηση ρυζιού και στη δεύτερη από δειγματοληψία των αποτελεσμάτων του δείχτη άνεσης PMV.

3.1.α Δεδομένα ρυζιού.

Έχουμε 107 μέτρησης από μία δημοσκόπηση ρυζιού.

Κάθε μέτρηση αποτελείται από 6 στοιχεία από τα οποία τα 5 αφορούν κάποια χαρακτηριστικά του ρυζιού ενώ το τελευταίο αποτελεί μια γενική άποψη για το πόσο αρεστό ήταν το ρύζι .(Ο πίνακας των δεδομένων βρίσκεται στο παράρτημα Α.)

Η εργασία που ακολουθεί είναι η μελέτη των παραμέτρων του subtractive clustering αλλά και της κατανομής των δηγμάτων σε δεδομένα εκπαίδευσης και ελέγχου με σκοπό την δημιουργία ενός FIS το οποίο με 5 εισόδους θα υπολογίζει όσο το δυνατόν πιο προσεγγιστικά την έξοδο σε σύγκριση την πραγματική τιμή του δείγματος.

3.1.β Δειγματοληψία PMV

$$PMV=[0.35\exp(-0.042H_M/A_N)+0.032] H_M/A_N(1-\eta)-0.32\{4.066(35.7-0.0275H_M/A_N(1-\eta)-84.2-P_a)\}-0.42\{H_M/A_N(1-\eta)-58\}-1.410^{-3}H_M/A_N(34-\theta_a)-0.0017H_M/A_N(44-P_a)-0.71\sigma F_{cl}\{(\theta_{cl} + 234)^4 - (\theta_{mrt} + 273)^4\}-F_{cl}h_c(\theta_{cl} - \theta_a)$$

A_N =Εμβαδό σώματος

F_{cl} =Λόγος του εμβαδού της επιφάνειας ντυμένου σώματος προς το εμβαδό της επιφάνειας γυμνού σώματος.

h_c =Συντελεστής μεταβίβασης θερμότητας με μεταφορά.

H_M =Ρυθμός παραγωγής θερμότητας από μεταβολισμό.

P_a =Μερική πίεση ατμών του αέρα.

θ_α =Θερμοκρασία περιβάλλοντος.

θ_{cl} =Θερμοκρασία της εξωτερικής επιφάνειας ντυμένου σώματος.

θ_{mrt} =Μέση θερμοκρασία ακτινοβολίας

σ = Σταθερά Stefan Boltzmann (5.78×10^{-8})

3.2 Λειτουργία Genfis2.m

Η συνάρτηση Genfis2.m αποτελεί ένα πολύ ισχυρό εργαλείο για την παραγωγή συστημάτων ασαφούς λογικής από ένα σετ εισόδων-εξόδων με την χρήση της μεθόδου ομαδοποίησης Subtractive clustering.

Αρχικά εφαρμόζοντας την μέθοδο Subtractive clustering εντοπίζονται τα κέντρα των cluster και συντάσσονται οι κανόνες. Διάφοροι παράμετροι όπως το εύρος (σ) των γκαουσιανών συναρτήσεων συμμετοχής και οι έξοδοι των κανόνων υπολογίζονται από τη γραμμική μέθοδο ελαχίστων τετραγώνων.

Η συνάρτηση συντάσσετε ως: FIS = GENFIS2(XIN,XOUT,RADII)

Το fis που παράγει είναι ένα Sugeno τύπου σύστημα ασαφούς λογικής , XIN είναι τα δεδομένα εισόδου και XOUT τα δεδομένα εξόδου στα οποία θέλουμε να προσαρμοστεί το σύστημα. Η μεταβλητή RADII δέχεται τιμές από 0 έως 1 και συμβολίζει την ακτίνα επιρροής για το subtractive clustering. Μπορεί να έχει διαστάσεις [1xN] δηλαδή κάθε διάσταση-τύπος δεδομένων να έχει διαφορετική ακτίνα επιρροής.

Η σύνταξη FIS = GENFIS2(...,XBOUNDS) μας δίνει την δυνατότητα να αλλάξουμε την κλίμακα στην οποία θα κανονικοποιητέ η κάθε είσοδο ή έξοδο. Η προκαθορισμένη κλίμακα όπως και στο subtractive clustering είναι [0,1] αυτό αλλάζει μεταβάλλοντας τις τιμές της τιμής XBOUNDS η οποία έχει δομή [2xN] όπου N ο αριθμός των τύπων των δεδομένων. Στην ουσία ορίζουμε για κάθε τύπο δεδομένων τα όρια στα οποία θέλουμε να κανονικοποιηθεί.

Στη σύνταξη `FIS = GENFIS2(...,XBOUNDS,OPTIONS)` το πεδίο `OPTIONS` επιτρέπει αλλαγές σε κάποιες προκαθορισμένες σταθερές του αλγορίθμου. Το πεδίο `OPTIONS` είναι το ίδιο για τον αλγόριθμο `subtractive clustering` και περισσότερες πληροφορίες φαίνονται με την εντολή `HELP SUBCLUST`.

`FIS = GENFIS2(...,XBOUNDS,OPTIONS,USER_CENTERS)` Στο πεδίο `USER_CENTERS` ο χρήστης έχει τη δυνατότητα να ορίσει τα αρχικά `cluster centers`. Έχει διαστάσεις $[C \times N]$ όπου C ο αριθμός των `cluster` και N το πλήθος των τύπων δεδομένων.

3.3 Εφαρμογή

Παρακάτω εξετάζουμε τα αποτελέσματα για 2 διαφορετικές τιμές του `cluster radius` (0.5, 0.3) και 3 διαφορετικές κατανομές των δειγμάτων.

Γενικά το `cluster radius` πρέπει να έχει όσο το δυνατόν μεγαλύτερη τιμή καθώς για μεγαλύτερη ακτίνα έχουμε λιγότερους κανόνες άρα και λιγότερες ανάγκες για υπολογιστική ισχύ.

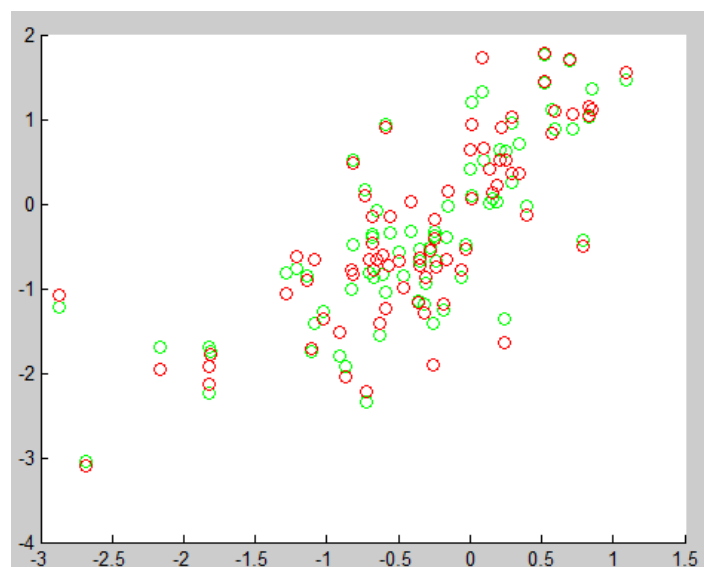
Το 70% των δειγμάτων χρησιμοποιείτε σαν εκπαίδευση και το υπόλοιπο 30% για έλεγχο. Το πρόβλημα που προκύπτει είναι πως κατά την εκπαίδευση τα δεδομένα πρέπει να είναι αντιπροσωπεύουν κάθε κατάσταση του χώρου ώστε το σύστημα ασαφούς λογικής να εκπαιδευτεί σε κάθε πιθανή είσοδο.

Με τη εντολή: `fismat=genfis2(train_input,train_output,0.5);`
Πραγματοποιούμε ομαδοποίηση στα δεδομένα `[train_input,train_output]` με την μέθοδο `subtractive clustering` με `cluster radius` ίσο με 0.5 και στη συνέχεια δημιουργείται και εκπαιδεύεται ένα σύστημα ασαφούς λογικής. Στην περίπτωση του ρυζιού ο πίνακας `train_input` έχει 5 στήλες και γραμμές όσα τα δεδομένα εκπαίδευσης ενώ ο πίνακας `train_output` έχει μία στήλη και γραμμές όσες και ο προηγούμενος. Στην περίπτωση του `PMV` ο πρώτος έχει διαστάσεις $[2 \times n]$ και ο δεύτερος πάλι $[1 \times n]$.

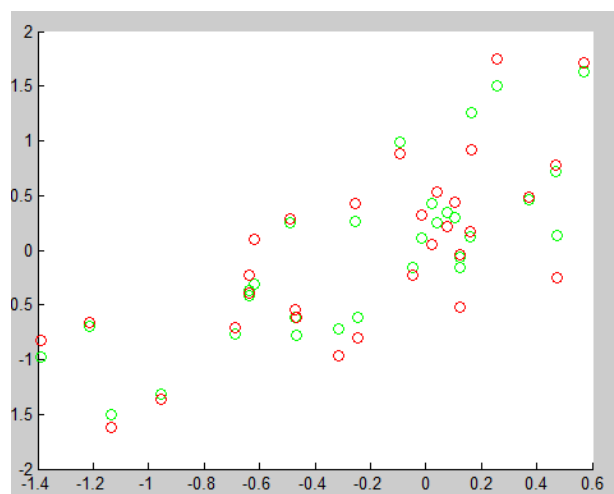
Με την εντολή `fuzout=evalfis(input,fismat);`
υπολογίζεται η έξοδος του `fis` για κάθε είσοδο `input`. Στη συνέχεια για τον υπολογισμό του μέσου τετραγωνικού σφάλματος συγκρίνεται το αποτέλεσμα `evalfis` με την πραγματική τιμή. Ο αλγόριθμος για κάθε μία από τις περιπτώσεις βρίσκεται στο παράρτημα Β 3.

3.3.α Για cluster radius 0.5

Δημοσκόπηση ρυζιού



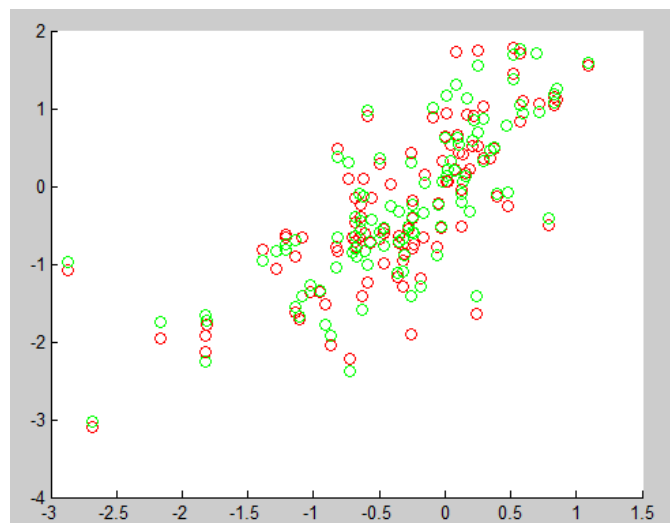
Σχήμα 3.1: με κόκκινο χρώμα απεικονίζεται η πραγματική έξοδος των δεδομένων εκπαίδευσης ενώ με πράσινο η έξοδος του FIS.(70 πρώτα δεδομένα του πίνακα)



Σχήμα3.2: Απεικονίζεται με κόκκινο η πραγματική έξοδος των δεδομένων ελέγχου και με πράσινο αντίστοιχα η έξοδος του FIS.(υπόλοιπα 30 δεδομένα του πίνακα)

Όπως παρατηρούμε και από τα διαγράμματα υπάρχει κάποια αστοχία, η οποία φαίνεται και από το μέσο τετραγωνικό σφάλμα ανά περίπτωση :

$\text{trainMSE} = 0.2070$ και $\text{checkMSE} = 0.2069$ Το σύστημα διαθέτει 4 κανόνες.

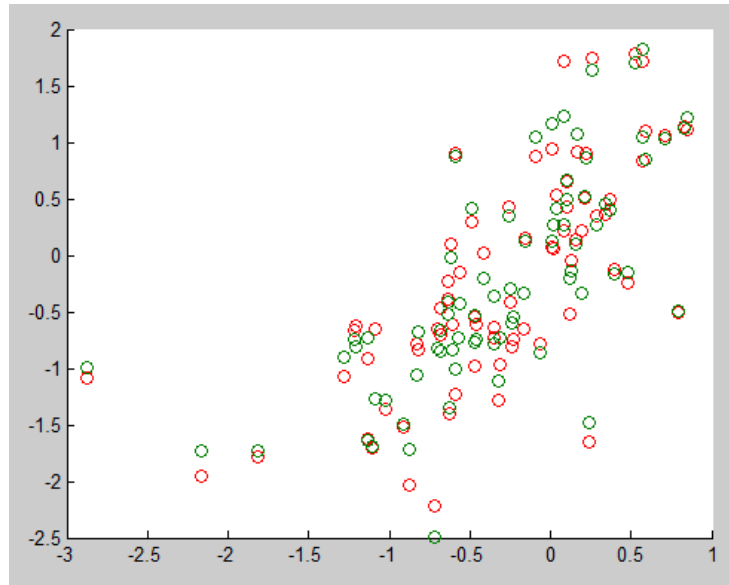


Σχήμα 3.3 : Κόκκινο χρώμα για τα πραγματικά δεδομένα εκπαίδευσης και πράσινο για τα αποτελέσματα που παράγει το FIS. (Όλα τα δεδομένα χρησιμοποιούνται για εκπαίδευση.)

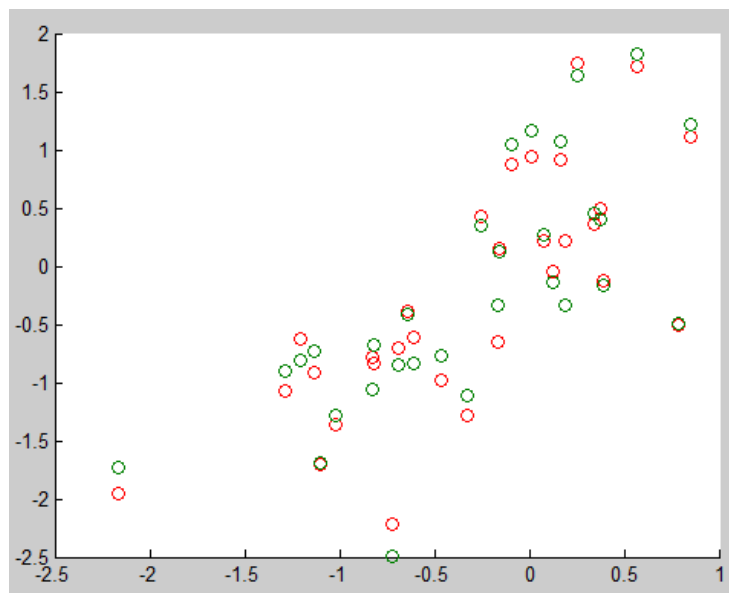
Σε αυτήν την περίπτωση έχουμε εφαρμόσει τον αλγόριθμο σε όλα τα δεδομένα του χώρου. Παρατηρείτε μία ελαφριά βελτίωση με μέσο τετραγωνικό σφάλμα $\text{trainMSE} = 0.1996$.

Παρόλα αυτά δεν ενδείκνυται να χρησιμοποιούνται όλα τα δεδομένα που διαθέτουμε για εκπαίδευση διότι παρόλο που το FIS εκπαιδεύεται καλύτερα, υπάρχει ο κίνδυνος να προσαρμοστεί υπερβολικά σε αυτά, να χάσει την ευελιξία του και να παρουσιάσει σφάλμα σε διαφορετικές καταστάσεις (Over fitting Error).

Το FIS και σε αυτήν την περίπτωση έχει 4 κανόνες.



Σχήμα 3.4: Κόκκινο χρώμα για τα πραγματικά δεδομένα εκπαίδευσης και πρασινο για τα αποτελέσματα που παράγει το FIS .(Χρησιμοποιούνται για εκπαίδευση τα 70 καλύτερα τυχαία δεδομένα.)

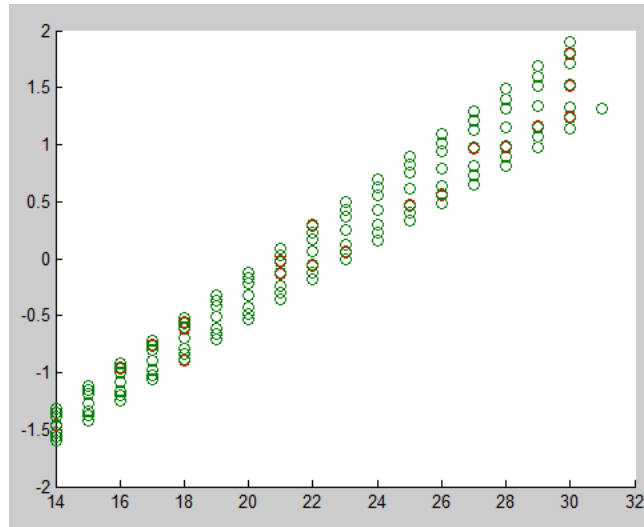


Σχήμα 3.5 : Κόκκινο χρώμα για τα πραγματικά δεδομένα ελέγχου και πρασινο για τα αποτελέσματα που παράγει το FIS .(Υπόλοιπο 30% της τυχαίας κατανομής)

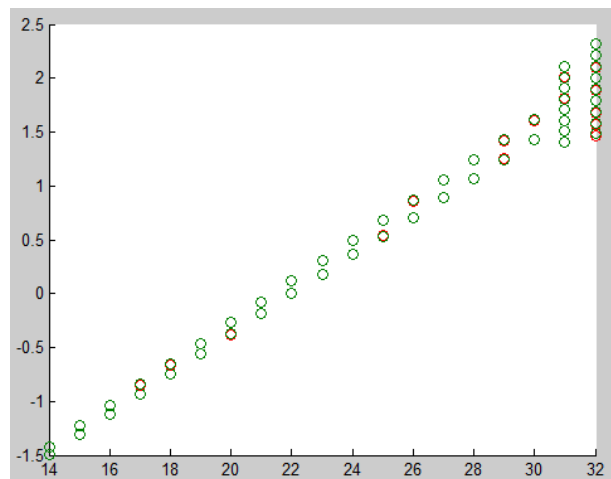
Σε αυτή την περίπτωση χρησιμοποιούμε πάλι το 70% περίπου των δεδομένων για εκπαίδευση και το υπόλοιπο για έλεγχο. Όμως αυτήν τη φορά τα δεδομένα που χρησιμοποιούνται για εκπαίδευση επιλέγονται τυχαία και έχει κρατηθεί το σετ με την καλύτερη απόδοση.

Εδώ το σφάλμα έχει μειωθεί ακόμη περισσότερο, με $\text{trainMSE}=0.1940$ και $\text{checkMSE}=0.1187$ με 4 κανόνες.

Δειγματοληψία PMV



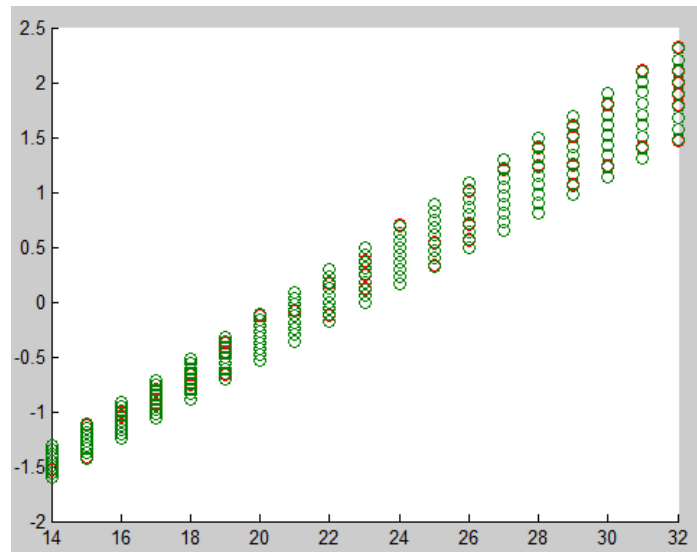
Σχήμα 3.6: Με κόκκινο απεικονίζεται η πραγματική τιμή του PMV ενώ με πράσινο η εξομοιωμένη από το FIS, για τα δεδομένα εκπαίδευσης. (Χρησιμοποιούνται το 70% των πρώτων δεδομένων για εκπαίδευση.)



Σχήμα 3.7 : Με κόκκινο απεικονίζεται η πραγματική τιμή του PMV ενώ με πράσινο εξομοιωμένη από το FIS, για τα δεδομένα ελέγχου. (Υπόλοιπο 30% των δεδομένων.)

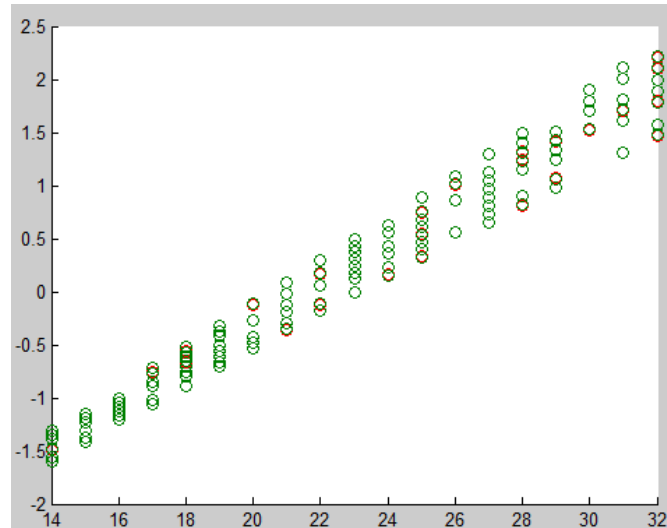
Σε αυτή την περίπτωση έχουμε πάρει όπως και προηγουμένως τις 75 πρώτες τιμές των δεδομένων για εκπαίδευση. Από τις δύο εικόνες μπορούμε να δούμε καθαρά πως στο αποτέλεσμα του ελέγχου (Σχήμα 3.7) το σφάλμα είναι πολύ μεγάλο στις θερμοκρασίες από 30 βαθμούς κελσίου και μετά, πράγμα το οποίο λογικό εφόσον το σύστημα ασαφούς λογικής δεν είναι εκπαιδευμένο σε αυτό το διάστημα.

Αυτό φαίνεται και από το τετραγωνικό σφάλμα : $\text{trainMSE}=0.0024$ και $\text{checkMSE}= 0.0054$ με 9 κανόνες.

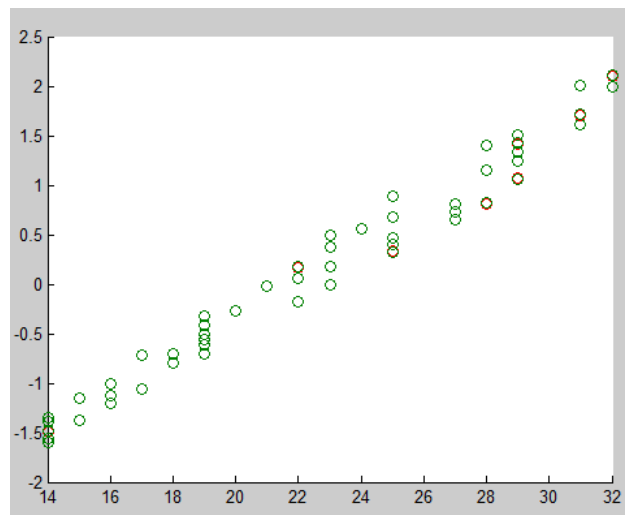


Σχήμα 3.8: Με πράσινο απεικονίζεται η πραγματική τιμή του PMV ενώ με κόκκινο η εξομοιωμένη από το FIS για τα δεδομένα ελέγχου. (Όλα τα δεδομένα χρησιμοποιούνται προς εκπαίδευση.)

Έχοντας εκπαιδέψει το FIS σε όλα τα δεδομένα παίρνουμε το αποτέλεσμα του σχήματος 3.8. Το τετραγωνικό σφάλμα έχει τιμή $\text{trainMSE}=0.0040$ και η υπολογιστική ισχύς του FIS έχει μειωθεί στους 8 κανόνες.



Σχήμα 3.9: Με πράσινο απεικονίζεται η πραγματική τιμή του PMV ενώ με κόκκινο η εξομοιωμένη του FIS για τα δεδομένα εκπαίδευσης. (Έχει επιλεχτεί το καλύτερο τυχαίο 70% των δεδομένων για εκπαίδευση.)



Σχήμα 3.10: Με πράσινο απεικονίζεται η πραγματική τιμή του PMV ενώ με κόκκινο η εξομοιωμένη του FIS στα δεδομένα ελέγχου. (Το υπόλοιπο 30% των δεδομένων.)

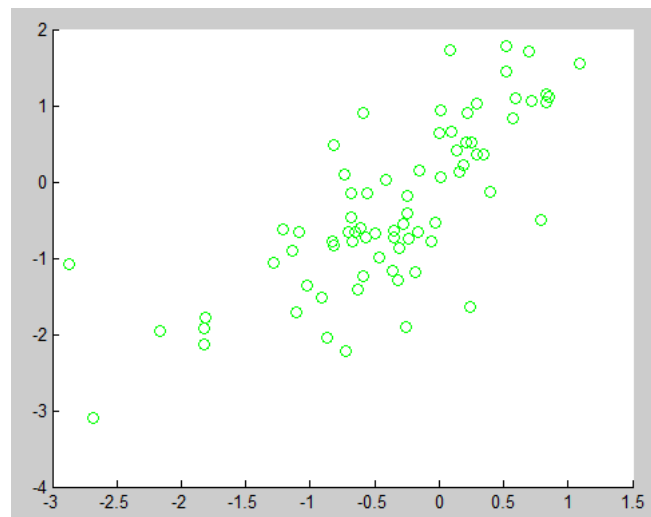
Με 75 τυχαία επιλεγμένα στοιχεία για εκπαίδευση τα σφάλματα είναι : $\text{trainMSE}=0.0039$ και $\text{checkMSE}=0.0027$. Συγκρίνοντας αυτές τις τιμές με εκείνες της πρώτης περίπτωσης παρατηρούμε πως το τετραγωνικό σφάλμα από τα δεδομένα εκπαίδευσης είναι ελαφρώς μεγαλύτερο αλλά το σφάλμα από τα δεδομένα ελέγχου είναι πολύ μικρότερο και με 8 κανόνες αντί για 9 .

Όσον αφορά την περίπτωση που είχαμε ολοκληρωτική εκπαίδευση , παρατηρούμε πως πάλι το αποτέλεσμα είναι αρκετά καλύτερο.

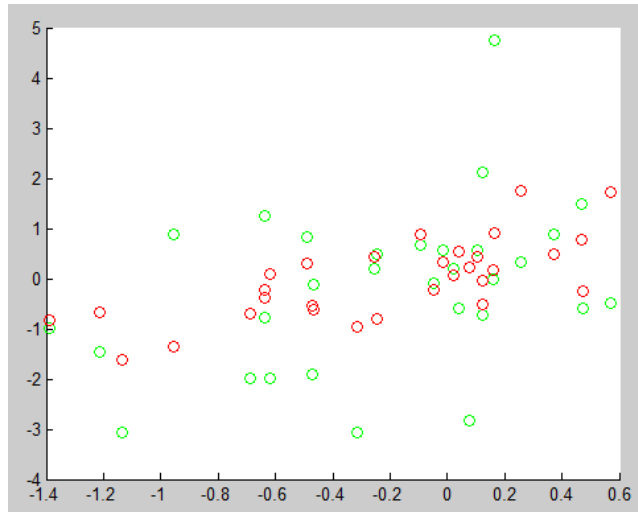
3.3.β Για cluster radius 0.3

Όπως αναφέραμε και σε προηγούμενο κεφάλαιο με το να μειώνεται το cluster radius επιτρέπεται η δημιουργία περισσότερων clusters άρα και περισσότερων κανόνων .

Δημοσκόπηση ρυζιού



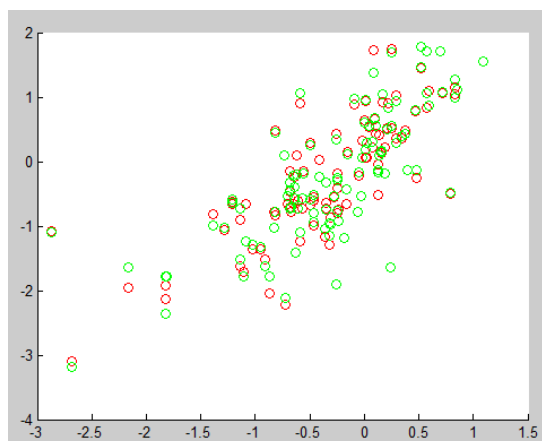
Σχήμα 3.11: Εδώ φαίνονται τα δεδομένα εκπαίδευσης και ελέγχου πλήρως ταυτισμένα.
(Χρησιμοποιούνται το 70% των πρώτων δεδομένων για εκπαίδευση).



Σχήμα 3.12: Με κόκκινο απεικονίζονται τα πραγματικά δεδομένα ελέγχου ενώ με πράσινο η εξομοιωμένη του FIS (30% των υπόλοιπων δεδομένων.)

Στο σχήμα 3.11 βλέπουμε καθαρά πως το σύστημα ασαφούς λογικής έχει εξομοιώσει τελείως την εκπαίδευση με $\text{trainMSE} = 6.0841e-015$ αλλά από το σχήμα 3.12 βλέπουμε πως υπάρχει πολύ μεγαλύτερο σφάλμα από πριν με $\text{checkMSE} = 1.4812$.

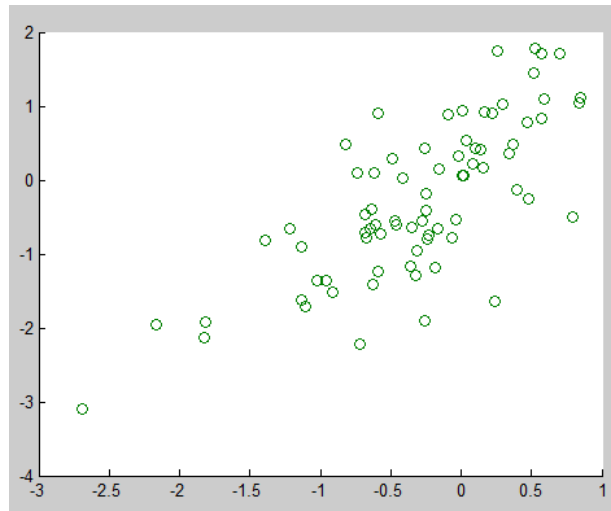
Παρόλο που το FIS σε αυτή τη περίπτωση διαθέτει υπερβολικά πολύ υπολογιστική ισχύ 22 κανόνων βλέπουμε πως έχει αποτύχει να πλησιάσει τα δεδομένα ελέγχου. Πράγμα το οποίο αποτελεί καθαρό παράδειγμα του over fitting error .



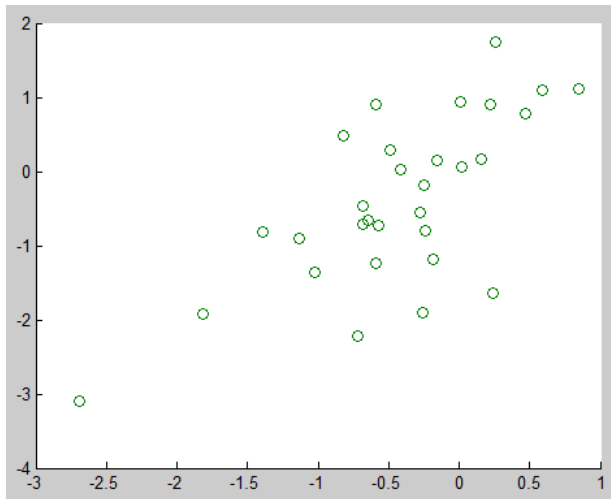
Σχήμα 3.13: Κόκκινο χρώμα για τα πραγματικά δεδομένα εκπαίδευσης και πράσινο για τα αποτελέσματα που παράγει το FIS .(Όλα τα δεδομένα χρησιμοποιούνται για εκπαίδευση.)

Στο σχήμα 3.13 φαίνονται τα αποτελέσματα για το FIS που είναι εκπαιδευμένο σε όλα τα δεδομένα του χώρου .

Εδώ το σφάλμα είναι : $\text{trainMSE} = 0.1551$. Μπορούμε να πούμε πως είναι σχετικά μικρότερο από πριν, και με πολύ λιγότερες υπολογιστικές απαιτήσεις καθώς έχει μόλις 10 κανόνες.



Σχήμα 3.14: Εδώ φαίνονται τα πραγματικά δεδομένα εκπαίδευσης καθώς το FIS έχει ταυτιστεί πλήρως. (Και πάλι για εκπαίδευση χρησιμοποιούνται το καλύτερο, τυχαίο 70%.)

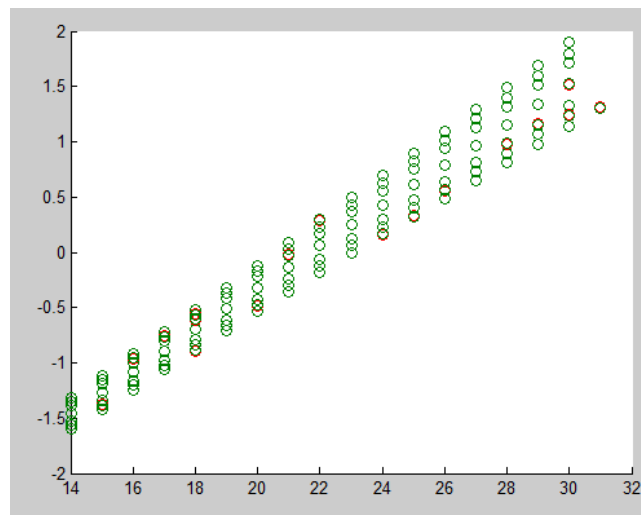


Σχήμα 3.15: Και εδώ φαίνονται μόνο τα δεδομένα ελέγχου καθώς ταυτίζονται πλήρως από την έξοδο του FIS. (Για έλεγχο γίνεται χρήση του υπόλοιπου 30%.)

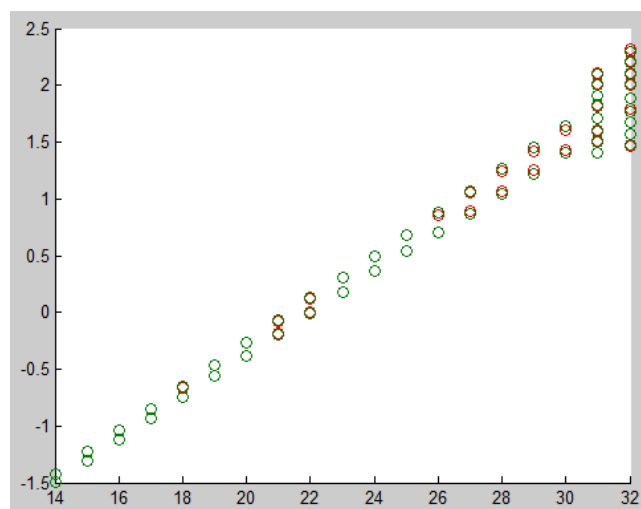
Σε αντίθεση με την πρώτη περίπτωση που είχαν επιλεχτεί για εκπαίδευση οι 75 πρώτες τιμές, τώρα επιλεχτήκαν τυχαία τιμές οι οποίες αντιπροσωπεύουν ολόκληρο τον χώρο με αποτέλεσμα τα σφάλματα εκπαίδευσης και ελέγχου να είναι μηδενικά .

Το μόνο πρόβλημα είναι πως και εδώ το FIS καταναλώνει πολύ υπολογιστική ισχύ, έχοντας 19 κανόνες.

Δειγματοληψία pmv

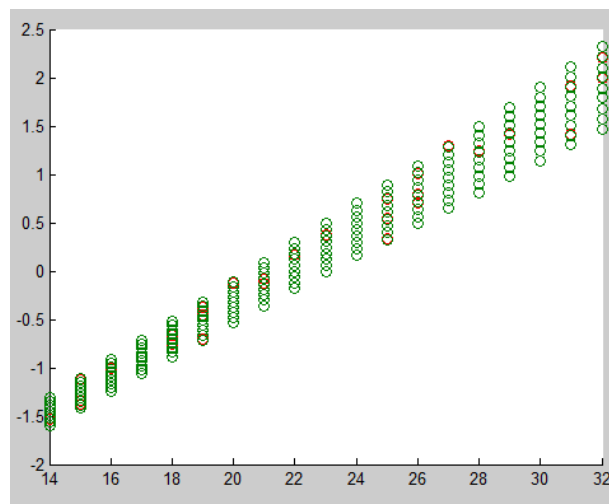


Σχήμα 3.16: Με κόκκινο απεικονίζεται η πραγματική τιμή του PMV ενώ με πράσινο εξομοιωμένη από το FIS, για τα δεδομένα εκπαίδευσης.(Χρησιμοποιούνται το 70% των πρώτων δεδομένων για εκπαίδευση.)



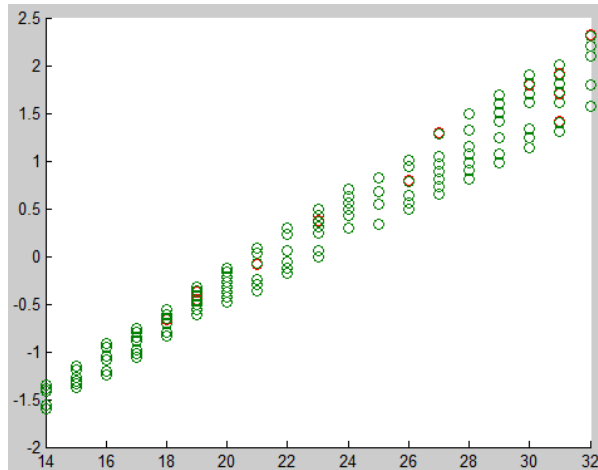
Σχήμα 3.17: Με κόκκινο απεικονίζεται η πραγματική τιμή του PMV ενώ με πράσινο εξομοιωμένη από το FIS, για τα δεδομένα ελέγχου. (Υπόλοιπο 30% των δεδομένων.)

Παρατηρούμε πως ακόμα και αυξάνοντας την υπολογιστική δύναμη του FIS στους 15 κανόνες δεν καταφέραμε να καλύψουμε το μειονέκτημα που οφείλεται στην άστοχη εκπαίδευση. Για την ακρίβεια, το $\text{trainMSE}=0.0021$ έπεσε ελάχιστα αλλά η δυσκαμψία που προκάλεσε στο σύστημα ασαφούς λογικής προκάλεσε τεράστια αύξηση στο σφάλμα ελέγχου, όπου $\text{checkMSE}=0.0126$.

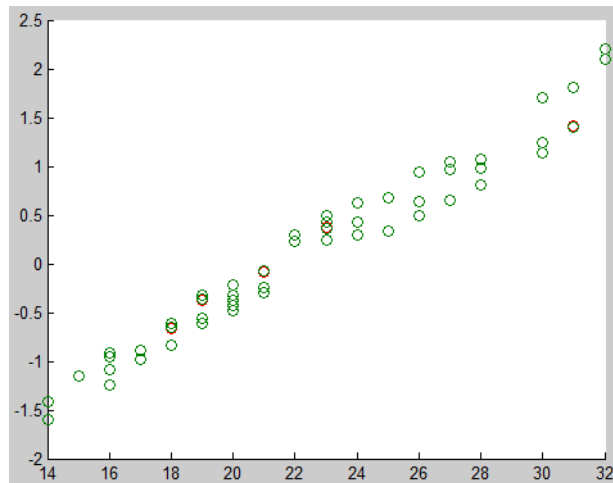


Σχήμα 3.18: Με πράσινο απεικονίζεται η πραγματική τιμή του PMV ενώ με κόκκινο η εξομοιωμένη από το FIS για τα δεδομένα ελέγχου. (Όλα τα δεδομένα χρησιμοποιούνται προς εκπαίδευση.)

Εδώ όπως ήταν αναμενόμενο το σφάλμα μειώθηκε στο μισό με $\text{trainMSE}=0.0024$ όμως αυξήθηκε πολύ ο αριθμός των κανόνων όπου έγιναν 21.



Σχήμα 3.19: Με κόκκινο απεικονίζεται η πραγματική τιμή του PMV ενώ με πράσινο η εξομοιωμένη του FIS για τα δεδομένα εκπαίδευσης. (Έχει επιλεχτεί το καλύτερο τυχαίο 70% των δεδομένων για εκπαίδευση.)



Σχήμα 3.20: Με κόκκινο απεικονίζεται η πραγματική τιμή του PMV ενώ με πράσινο η εξομοιωμένη του FIS στα δεδομένα ελέγχου. (Το υπόλοιπο 30% των δεδομένων.)

Οι τιμές των σφαλμάτων σε αυτή τη περίπτωση είναι $\text{trainMSE}=0.0023$ και $\text{checkMSE}=0.0016$ με 19 κανόνες .

Το σφάλμα της εκπαίδευσης είναι σχεδόν το ίδιο με την προηγούμενη περίπτωση παρατηρούμε όμως πως στην περιοχή που γίνεται ο έλεγχος η ανταπόκριση του συστήματος ασφούς λογικής είναι καλύτερη.

3.4 Συμπεράσματα και παρατηρήσεις

Εν τέλει συμπεράνουμε πως το FIS για την σωστή εκπαίδευση πρέπει να έχει αντιπροσωπευτικά δεδομένα καθώς και τις κατάλληλες σταθερές ώστε να έχουμε όσο το δυνατόν καλύτερη ανταπόκριση σε πιθανές εισόδους με την μικρότερο δυνατό αριθμό κανόνων.

Στον πίνακα που ακολουθεί παρουσιάζονται όλα τα αποτελέσματα συγκεντρωμένα.

Σετ δεδομένων	Μέγεθος ακτίνας	Κατανομή δεδομένων εκπαίδευσης	Mse εκπαίδευσης	Mse ελέγχου	Αριθμός κανόνων
ΡΥΖΙ	0.5	70% πρώτα	0.2070	0.2069	4
		Όλα	0.1996	-	4
		70% Τυχαία	0.1940	0.1187	4
	0.3	70% πρώτα	6.0841e-015	1.4812	22
		Όλα	0.1551	-	10
		70% Τυχαία	0	0	19
PMV	0.5	70% πρώτα	0.0024	0.0054	9
		Όλα	0.0040	-	8
		70% Τυχαία	0.0039	0.0027	8
	0.3	70% πρώτα	0.0021	0.0126	15
		Όλα	0.0024	-	21
		70% Τυχαία	0.0023	0.0016	19

4. Η ΜΕΘΟΔΟΣ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ BIG BANG BIG CRUNCH

4.1 Εισαγωγή

Όλοι οι εξελικτικοί αλγόριθμοι είναι ευριστικές, βασισμένες σε πληθυσμό διαδικασίες αναζήτησης οι οποίες ενσωματώνουν τυχαίες μεταβολές και επιλογή. Ο αλγόριθμος Big-Bang Big-Crunch είναι μια μέθοδος βελτιστοποίησης η οποία βασίζεται σε μία από τις θεωρίες εξέλιξης του σύμπαντος, από την οποία πείρε και το όνομα της Big Bang και Big Crunch Theory. Στη φάση του Big Bang η διάχυση της ενέργειας προκαλεί αταξία και τυχαιότητα, έπειτα στην φάση Big Crunch τα τυχαία διασκορπισμένα σωματίδια μπαίνουν σε μια τάξη.

Γενικά ένα πρόβλημα βελτιστοποίησης θεωρείται η εύρεση ενός σετ παραμέτρων $\{x_n\}$ οι οποίες ελαχιστοποιούν την συνάρτηση κόστους $f(x_0, \dots, x_n)$. Η βελτιστοποίηση απαιτεί να ληφθεί η καλύτερη απόφαση για ένα σετ περιστάσεων.

4.2 Big-Bang Big-Crunch (BB-BC)

Όπως είπαμε και προηγούμενος η τυχαιότητα παρομοιάζεται με την διάχυση της ενέργειας στο σύμπαν ενώ τη σύγκλιση σε ένα τοπικό ή ολικό βέλτιστο σημείο μπορούμε να τη δούμε ως βαρυτική έλξη. Από τη στιγμή που η διάχυση της ενέργειας προκαλεί αταξία από τα προηγούμενως ταξινομημένα σωματίδια, θα χρησιμοποιήσουμε τη τυχαιότητα από την μεταμόρφωση μίας προηγούμενης λύσης που προήλθε από σύγκλιση για τη δημιουργία νέων υποψηφίων λύσεων.

Η δημιουργία του αρχικού πληθυσμού ονομάζεται φάση Big Bang. Σε αυτή τη φάση όλες οι υποψήφια λύσεις διασκορπίζονται ομοιόμορφα στον χώρο αναζήτησης. Επειδή όμως οι γεννήτρια τυχαίων κανονικών αριθμών μπορεί να παράγει αριθμούς εκτός του χώρου αναζήτησης είναι απαραίτητο να περιοριστούν οι τιμές τους.

Επομένως δημιουργείτε η μήτρα των τυχαίων σωματιδίων : $X = [x_1, x_2, \dots, x_n]$

Στη συνέχεια ακολουθεί η φάση Big-Crunch. Αυτή είναι μια διαδικασία σύγκλισης των τυχαίων δεδομένων σε ένα σημείο. Μπορεί να έχει πολλές εισόδους αλλά μόνο μία έξοδος η οποία είναι το σημείο για το οποίο η συνάρτηση κόστους έχει την ελάχιστη τιμή.

Οπότε το σημείο x_c που είναι το κέντρο βάρους υπολογίζεται ως :

$$x_c = \frac{\sum_{i=1}^N \frac{1}{f(x_i)} x_i}{\sum_{i=1}^N \frac{1}{f(x_i)}}$$

Όπου x_i είναι το τρέχον σωματίδιο,

$f(x_i)$ είναι η τιμή της συνάρτησης κόστους για το τρέχων σωματίδιο. Όσο πιο κοντά είναι η τιμή της συνάρτησης κόστους στο 0 τόσο μεγαλύτερη επιρροή έχει το σωματίδιο x_i στην τιμή του κέντρου βάρους.

Ύστερα από τη φάση Big-Crunch ο αλγόριθμος πρέπει να δημιουργήσει νέα μέλη, τα οποία θα χρησιμοποιηθούν για το Big-Bang της επόμενης επανάληψης. Αυτό γίνεται με πολλούς τρόπους, ένας από αυτούς είναι η επιστροφή στο πρώτο βήμα με τη δημιουργία ενός τυχαίου πληθυσμού από την αρχή. Σε αυτή την περίπτωση ο αλγόριθμος δεν θα είχε καμία διαφορά από έναν αλγόριθμο τυχαίας αναζήτησης καθώς οι επόμενες επαναλήψεις δεν χρησιμοποιούν την γνώση από τις προηγούμενες. Ως επακόλουθο ένας τέτοιος αλγόριθμος συγκλίνει πολύ αργά.

Ένας αλγόριθμος βελτιστοποίησης πρέπει να συγκλίνει σε ένα ιδανικό σημείο, αλλά παράλληλα για να είναι πραγματικά λειτουργικός πρέπει να δίνει σε σημεία στον χώρο αναζήτησης φθίνουσα πιθανότητα. Συγκεκριμένα, ένας μεγάλος αριθμός λύσεων πρέπει να βρίσκονται κοντά στο 'ιδανικό σημείο', αλλά τα υπόλοιπα σημεία ύστερα από επαναλήψεις πρέπει διασκορπίζονται στον χώρο. Αυτή η αναλογία των σημείων γύρο από την ιδανική τιμή και των διασκορπισμένων πρέπει να μειώνεται με το πέρασμα των επαναλήψεων.

Τα παραπάνω επιτυγχάνονται με το να σκορπίζονται τα νέα σημεία γύρο από το κέντρο βάρους. Επομένως ακολουθεί η ενημέρωση της μίτρας των σωματιδίων ώστε να ξεκινήσει εκ-νέου η φάση Big-Bang της επόμενης επανάληψης. Οι νέοι υποψήφιοι υπολογίζονται γύρο από το κέντρο βάρους από την εξίσωση.

$$x^{new} = x^c + lr/k$$

Όπου x^c είναι το κέντρο βάρους.

l είναι το πάνω όριο των σωματιδίων

r είναι ένας τυχαίος κανονικός αριθμός

και k είναι ο αριθμός της τρέχων επανάληψης.

Συνοψίζοντας τα βήματα του αλγορίθμου (BB-BC) έχουμε .

Βήμα1. Δημιουργία αρχικού πληθυσμού N τυχαίων υποψηφίων, δίνοντας σημασία στα όρια του χώρου αναζήτησης .

Βήμα 2. Υπολογισμός της τιμής της συνάρτησης κόστους για όλους τους υποψηφίους.

Βήμα3. Βάση τη συνάρτηση (1) υπολογίζουμε το κέντρο βάρους . Αντί αυτού μπορούμε να χρησιμοποιήσουμε την πιο ταιριαστή λύση ως κέντρο βάρους .

Βήμα4. Υπολογισμός νέων υποψηφίων γύρω από το κέντρο βάρους προσθέτοντας ή αφαιρώντας έναν κανονικό τυχαίο αριθμό , ο οποίος καθώς περνούν οι επαναλήψεις η τιμή του μειώνεται .

Βήμα 5. Επιστροφή στο βήμα 2 μέχρι να ικανοποιηθούν τα κριτήρια.

4.3 Παραδείγματα

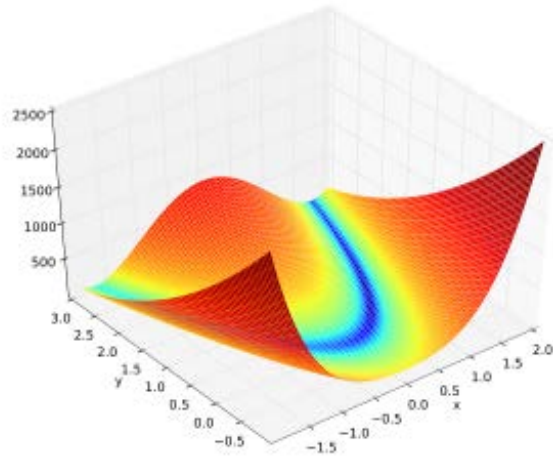
Στην παράγραφο που ακολουθεί αναλύσουμε την λειτουργία του αλγορίθμου και μέσα από τα παραδείγματα που ακολουθούν θα τονίσουμε κάποια συμπεράσματα.

4.3.α Η συνάρτηση Rosenbrock

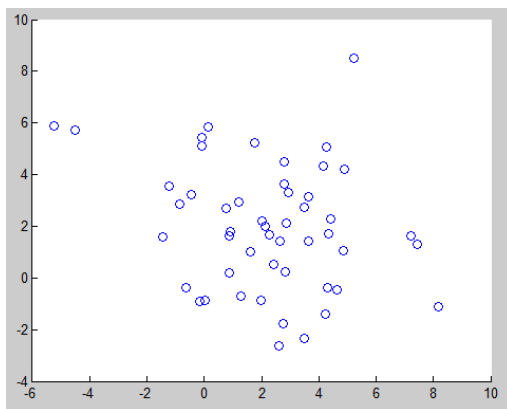
Η συνάρτηση Rosenbrock προτάθηκε το 1960 από τον Howard H. Rosenbrock ως πρόβλημα δοκιμής για τις μεθόδους βελτιστοποίησης και ορίζεται από τον τύπο :

$$f(x) = (1 - x)^2 + 100(y - x^2)^2$$

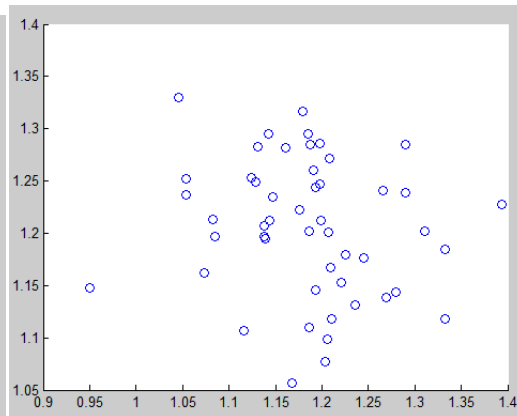
Όπως φαίνεται και από το σχήμα 3.3.1 το ολικό ελάχιστο της συνάρτησης βρίσκεται μέσα σε μια στενή σαν παραβολική επίπεδη κοιλάδα. Το να εντοπιστεί η κοιλάδα δεν είναι κάτι το σπουδαίο όμως η σύγκλιση στο ολικό ελάχιστο είναι δύσκολη.



σχήμα 4.3.1 Η συνάρτηση Rosenbrock. Έχει ολικό ελάχιστο στο σημείο $(x, y) = (1, 1)$ όπου $f(x, y) = 0$



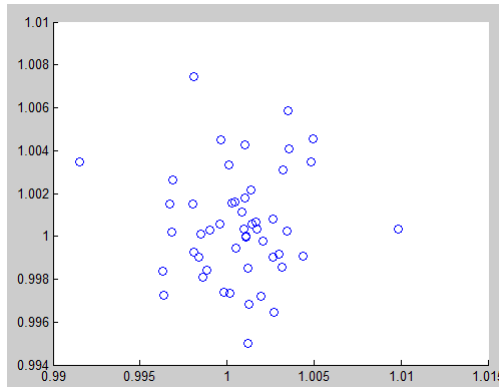
Σχήμα 4.3.2



Σχήμα 4.3.3

Στο σχήμα 4.3.2 βλέπουμε το πρώτο πλήθος των 50 υποψηφίων από την πρώτη φάση Big-Bang όπου είναι απολύτως τυχαία κατανεμημένοι στον χώρο .

Στο σχήμα 4.3.3 παρατηρούμε πως ύστερα από 5 επαναλήψεις οι υποψήφιοι έχουν περιοριστεί στις τιμές από 0.9 έως 1.4 δηλαδή γύρο από το βέλτιστο σημείο.

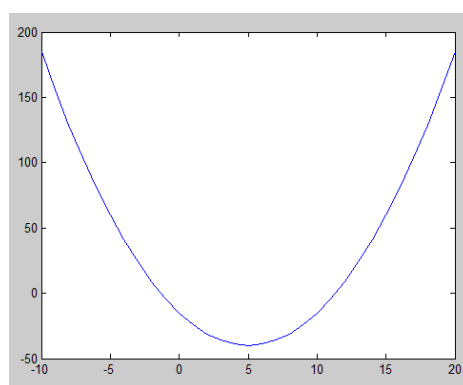


Σχήμα 4.3.4

Στην εικόνα φαίνεται η θέση των σωματιδίων μετά από 200 επαναλήψεις. Παρατηρούμε πως όλα τα σωματίδια είναι κοντά στο σημείο [1,1], πράγμα το οποίο φαίνεται και από την τιμή του κέντρου μάζας που είναι [1.0006,1.003].

4.4 Best fit solution

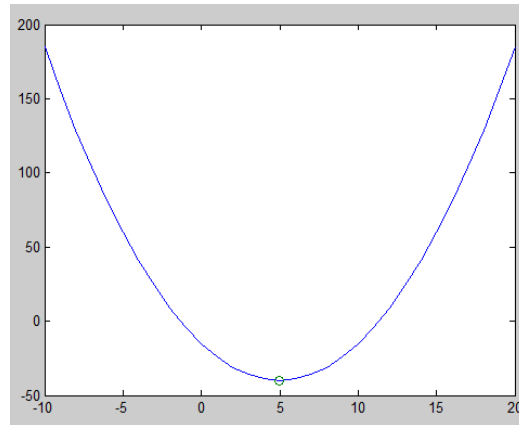
Στο παράδειγμα με την συνάρτηση Rosenbrock ο αλγόριθμος κατάφερε και σύγκλινε πολύ γρήγορα στο ολικό ελάχιστο της συνάρτησης και με μεγάλη ακρίβεια. Όμως αν στην περίπτωση που η συνάρτηση κόστους παίρνει και αρνητικές τιμές τότε παρατηρούμε πως ο αλγόριθμος αδυνατεί να συγκλίνει, εφόσον δεν είναι σε θέση να υπολογίσει το κέντρο βάρους.



Σχήμα 4.4.1

Τρέχοντας τον αλγόριθμο με κριτήριο κόστους την συνάρτηση που απεικονίζεται στην εικόνα 4.4.1 $f(x) = (x - 5)^2 - 40$ το αποτέλεσμα που λαμβάνουμε είναι το 1.7550, πράγμα τελείως λάθος εφόσον ολικό ελάχιστο έχει στην τιμή -40 για $x=5$.

Χρησιμοποιώντας τώρα την πιο ταιριαστή λύση αντί το κέντρο μάζας το βέλτιστο σημείο που μας δίνει ο αλγόριθμος είναι το αναμενόμενο 5 όπως φαίνεται και στην εικόνα 4.4.2.



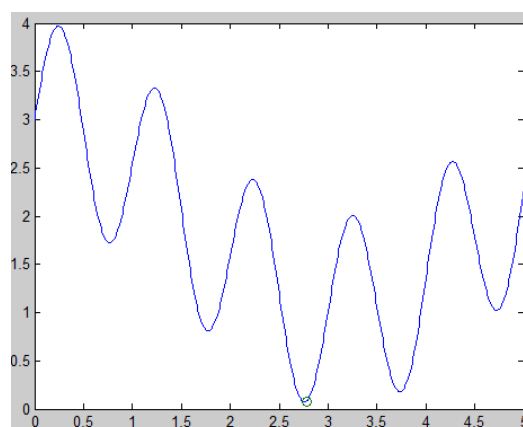
Σχήμα 4.4.2

Στη συνέχεια εξετάζουμε τη λειτουργικότητα του αλγορίθμου σε εξισώσεις όπου υπάρχουν πολλά τοπικά ακρότατα.

Η εξίσωση που χρησιμοποιείται αυτή τη φορά είναι $f(x) = \cos(x) + \sin(2\pi x) + 2$

Βάση τον αλγόριθμο που βρίσκεται στο παράρτημα (...) έχουμε τα παρακάτω αποτελέσματα.

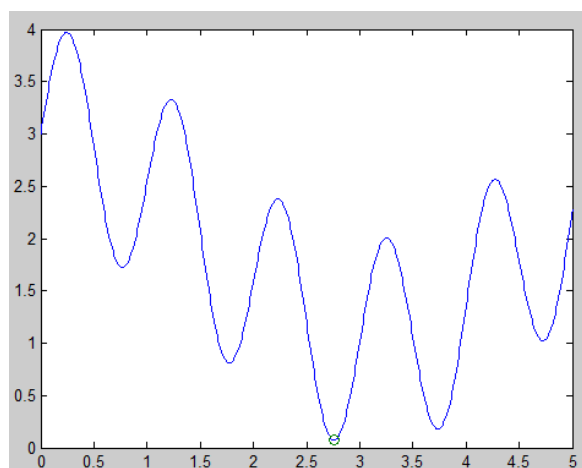
Χρησιμοποιώντας την κλασική μέθοδο με το κέντρο βάρους με 40 σωματίδια και 40 επαναλήψεις :



Σχήμα 4.4.3

Εδώ βλέπουμε πως έχει γίνει σύγκλιση στο ολικό ελάχιστο αν και υπάρχει μια μικρή απόκλιση .

Χρησιμοποιώντας τη βέλτιστη δυνατή λύση έχουμε:



Σχήμα 4.4.4

Παρόλο που στην εικόνα 4.4.4 ο αλγόριθμος φαίνεται να λειτουργεί άψογα, υπάρχουν περιπτώσεις όπου το αποτέλεσμα απέχει τελείως από το βέλτιστο σημείο.

Αυτό συμβαίνει γιατί η γνώση που επιλέγει ο αλγόριθμος για την ενημέρωση των κεντρών στη νέα φάση Big-Bang βασίζεται αυστηρά στο σωματίδιο που δίνει στην μικρότερη τιμή στην συνάρτηση κόστους. Επομένως είναι πιθανών , από την πρώτη διάχυση των σωματιδίων να γίνει σύγκλιση σε ένα τοπικό ελάχιστο αλλά όχι στο ολικό με αποτέλεσμα ο αλγόριθμος να αποτύχει. Αυτό μπορεί να αντιμετωπιστεί με το να αυξηθεί ο αριθμός των σωματιδίων ώστε να καλυφτεί όσο το δυνατόν μεγαλύτερη επιφάνια στη πρώτη φάση Big-Bang.

4.5 Συμπεράσματα και παρατηρήσεις

Συνοψίζοντας, χρησιμοποιώντας το κέντρο βάρους πρέπει να αντιμετωπίσουμε το πρόβλημα των αρνητικών τιμών αλλά το γεγονός ότι παίζουν ρόλο όλα τα σωματίδια εξασφαλίζει πιο ολοκληρωμένη γνώση για την επόμενη επανάληψη.

Από την άλλη η χρήση της βέλτιστης λύσης δεν έχει πρόβλημα με αρνητικά αποτελέσματα, όμως πρέπει να ληφθούν μέτρα για την αξιοπιστία των σωματιδίων.

Κεφάλαιο 5

Βελτιστοποίηση συστήματος ασαφούς λογικής με τη μέθοδο Big Bang – Big Crunch.

5.1 Εισαγωγή

Όπως είδαμε στα προηγούμενα κεφάλαια η μέθοδος bbbc είναι στην ουσία ένας αλγόριθμος αναζήτησης του ολικού ελαχίστου μιας συνάρτησης κόστους.

Στην παρούσα ενότητα, πραγματοποιούμε εκπαίδευση στο σύστημα ασαφούς λογικής με την παραπάνω μέθοδο και ως συνάρτηση κόστους το μέσο τετραγωνικό σφάλμα. Επειδή όπως τονίσαμε και προηγουμένως ο αλγόριθμος big bang big crunch χρειάζεται να περιοριστεί ο χώρος αναζήτησης του, πράγμα το οποίο επιτυγχάνετε με τη χρήση της γνώσης από την εκπαίδευση του genfis2.

5.2 Λειτουργία

Αρχικά εφαρμόζουμε τη συνάρτηση genfis2.m για να δημιουργήσουμε ένα σύστημα ασαφούς λογικής από ένα σετ δεδομένων. Από αυτό το σύστημα ασαφούς λογικής που δημιουργήσαμε στην παρούσα άσκηση επιλέγουμε να εκπαιδεύσουμε το εύρος των συναρτήσεων συμμετοχής των κανόνων με σκοπό τη μείωση του μέσου τετραγωνικού σφάλματος.

Στη συνέχεια ορίζουμε τις παραμέτρους του αλγορίθμου οι οποίες είναι ο αριθμός σωματιδίων, ο αριθμός επαναλήψεων, και η το ποσοστό στο οποίο θα απέχουν οι τιμές τις οποίες ψάχνει ο αλγόριθμος από τις αρχικές.

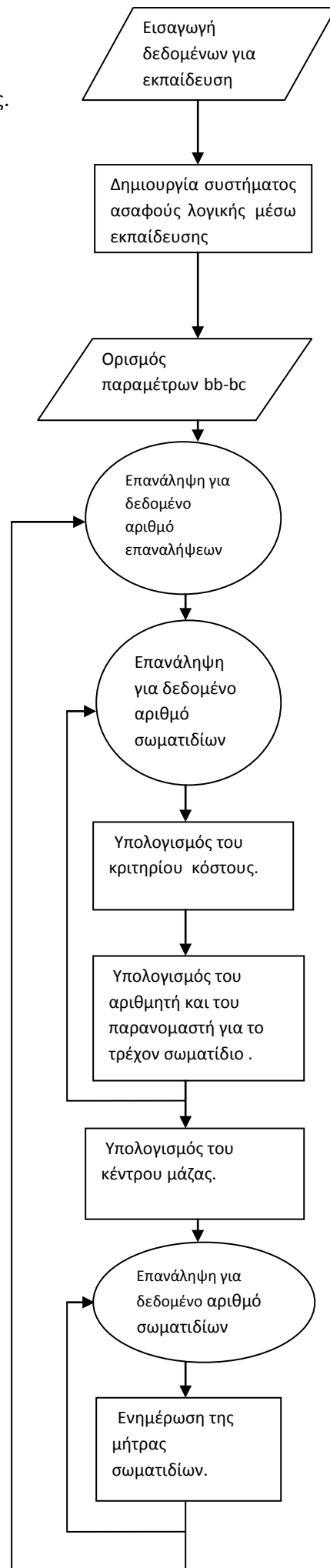
Έπειτα φορτώνονται ένα-ένα όλα τα σωματίδια στις παραμέτρους του FIS, υπολογίζετε η τιμή του κριτηρίου.

Αμέσως μετά από τις τιμές του κριτηρίου για κάθε σωματίδιο υπολογίζεται το κέντρο μάζας.

Τέλος γίνεται η ενημέρωση της μήτρας σωματιδίων και ακολουθεί η επόμενη επανάληψη.

Ακολουθεί το διάγραμμα ροής που παρουσιάζει και οπτικά τα παραπάνω βήματα.

Διάγραμμα ροής.



5.3 Μαθηματική ανάλυση

Η τιμή των σωματιδίων ορίζεται ως $rw(i, k) = rand[dratio * w(k), uratio * w(k)]$

Όπου $0 < i < \text{αριθμός σωματιδίων}$ και $0 < k < \text{αριθμός κανόνων}$

$dratio$ και $uratio$ δηλώνουν πόσο πάνω και κάτω θα διακυμαίνονται οι τιμές των σωματιδίων από τις αρχικές. Στην εφαρμογή μας χρησιμοποιούμε και για τα δύο τη τιμή 1.1 δηλαδή οι τιμές των σωματιδίων να κυμαίνονται από 90% έως 110% της αρχικής.

Η τιμή rw συμβολίζει τα σωματίδια και $w(k)$ το εύρος της συνάρτησης συμμετοχής.

Για τον υπολογισμό του κέντρου μάζας χρησιμοποιείτε η φόρμουλα που δήξαμε στο προηγούμενο κεφάλαιο :

$$x_c = \frac{\sum_{i=1}^N \frac{1}{f(rw_i)} x_i}{\sum_{i=1}^N \frac{1}{rw(x_i)}}$$

Όπου $f(rw_i)$ είναι η τιμή του κριτηρίου με είσοδο το τρέχων σωματίδιο.

Στην εφαρμογή σαν κριτήριο χρησιμοποιούμε το μέσο τετραγωνικό σφάλμα, το οποίο το ορίζουμε ως:

$$trnmse = \frac{norm(fuzout - y)}{\sqrt{\text{πλήθος}}}$$

Όπου $fuzout$ είναι η έξοδος του συστήματος ασαφούς λογικής και όπως φαίνεται από την εξίσωση την αφαιρούμε από την αντίστοιχη πραγματική έξοδο για να βρούμε την απόκλιση και τέλος διαιρούμε με το πλήθος των δεδομένων.

Έπειτα ακολουθεί η ενημέρωση των σωματιδίων που όπως δήξαμε και σε προηγούμενο κεφάλαιο πραγματοποιείτε με τη φόρμουλα :

$$x^{new} = x^c + lr/k$$

Όπου x^c είναι το κέντρο βάρους.

l είναι το πάνω όριο των σωματιδίων

r είναι ένας τυχαίος κανονικός αριθμός

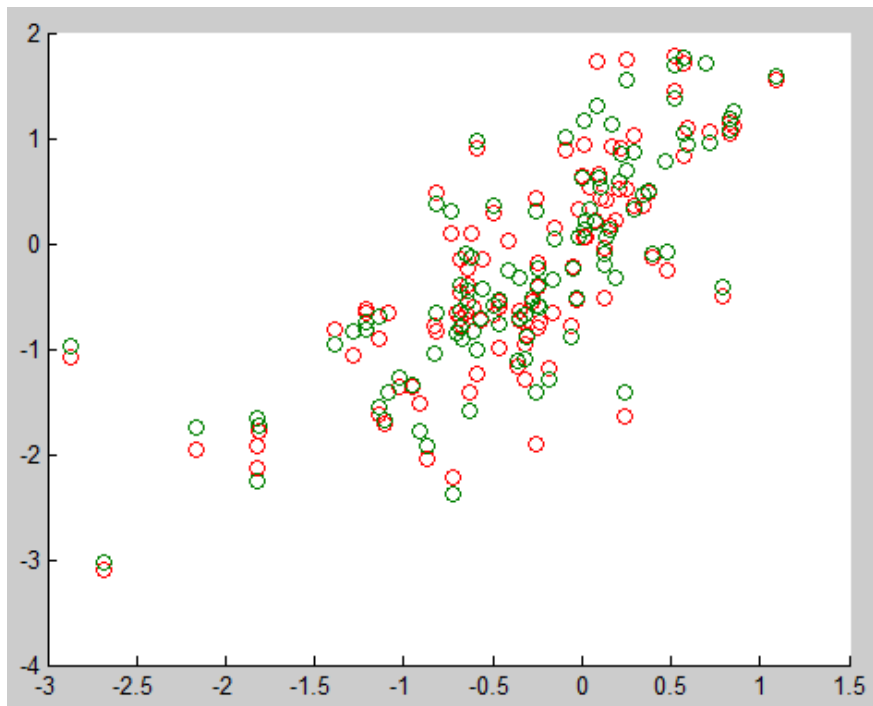
και k είναι ο αριθμός της τρέχων επανάληψης.

5.4 Εφαρμογή στο Matlab

Στην παρούσα ενότητα χρησιμοποιούμε όλα τα παραπάνω για να εκπαιδύσουμε ένα σύστημα ασαφούς λογικής από το κεφάλαιο 5 με σκοπό να το κάνουμε να πλησιάσει την πραγματική έξοδο.

Επομένως δοκιμάζουμε την μέθοδο για την περίπτωση που εκπαιδεύεται ένα FIS σε όλα τα δεδομένα και `cluster radius 0.5`.

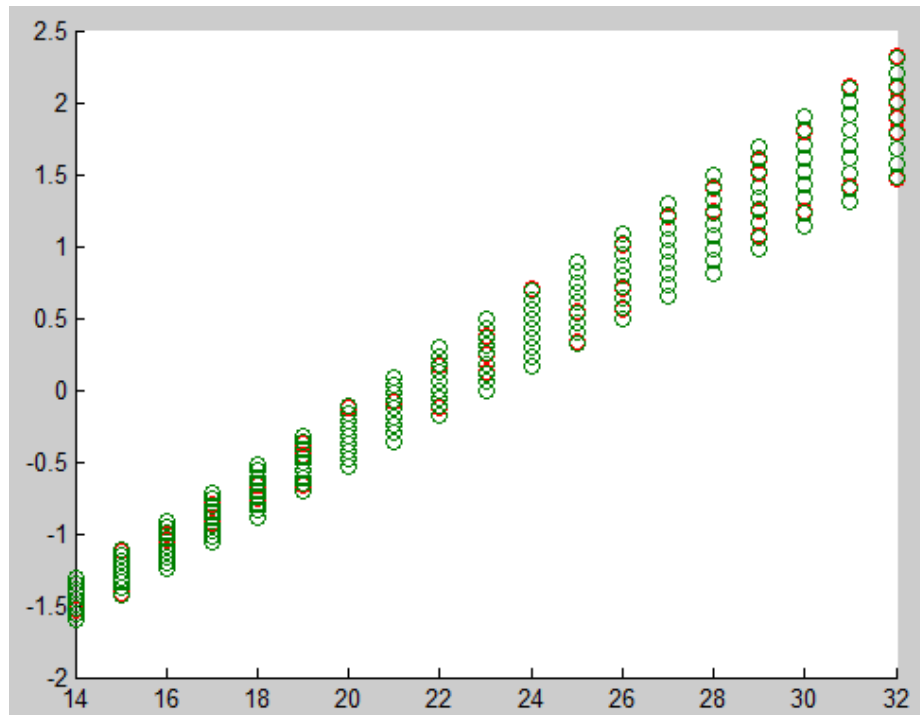
5.4.α Δεδομένα ρυζιού



Σχήμα 5.4.1: Με κόκκινο παρουσιάζονται τα πραγματικά δεδομένα ενώ με πράσινο τα προϊόντα εκπαίδευσης.

Από το προηγούμενο κεφάλαιο η εκπαίδευση σε όλα τα δεδομένα έδωσε μέσο τετραγωνικό σφάλμα `trnrmse= 0.1996`. Μετά την βελτιστοποίηση με τη μέθοδο Big bang-Big crunch το μέσο τετραγωνικό σφάλμα έπεσε στο `0.1985`.

5.4.β Δεδομένα PMV



Σχήμα 5.4.2: Με κόκκινο παρουσιάζονται τα πραγματικά δεδομένα ενώ με πράσινα τα προϊόντα εκπαίδευσης.

Η απλή εκπαίδευση του `genfis2.m` σε αυτή την περίπτωση άφηνε μέσο τετραγωνικό σφάλμα με τιμή 0.0040. Ύστερα από την εκπαίδευση με BigBang-BigCrunch το σφάλμα μειώνεται στο $\text{trnrmse}=0.0039$.

5.5 Παρατηρήσεις

Η διαφορά που παρουσιάζεται στο σφάλμα μπορεί να είναι φαινομενικά μικρή αλλά πρέπει να αναλογιστούμε πως έχουμε να κάνουμε με ένα σύστημα το οποίο είναι ήδη εκπαιδευμένο στα δεδομένα τα οποία πραγματοποιείτε η εκπαίδευση. Επίσης το γεγονός ότι ασκούμε βελτιστοποίηση μόνο σε έναν παράγοντα δεν μπορεί να αποφέρει πάντα μεγάλη διαφορά. Τα παραπάνω μπορούμε να τα διακρίνουμε αν συγκρίνουμε τα αποτελέσματα και με το εργαλείο βελτιστοποίησης anfis τα οποία είναι τα ίδια και σε κάποιες περιπτώσεις χειρότερα.

Τύπος δεδομένων	Μόνο genfis2	BB-BC	Anfis
Ρύζι	0.1996	0.1985	0.1985
PMV	0.0040	0.0039	0.0040

ΚΕΦΑΛΑΙΟ 6 Συνδυασμός και εφαρμογή διάφορων μεθοδολογιών

Εισαγωγή

Στο παρόν κεφάλαιο γίνεται δημιουργία ενός συστήματος ασαφούς λογικής με τη χρήση τριών διαφορετικών μεθοδολογιών, για τρία διαφορετικά συστήματα (δημοσκόπηση ρυζιού ,αποτελέσματα pmv και η συνάρτηση $f(x) = \sin(2x) / e^{\frac{x}{5}}$

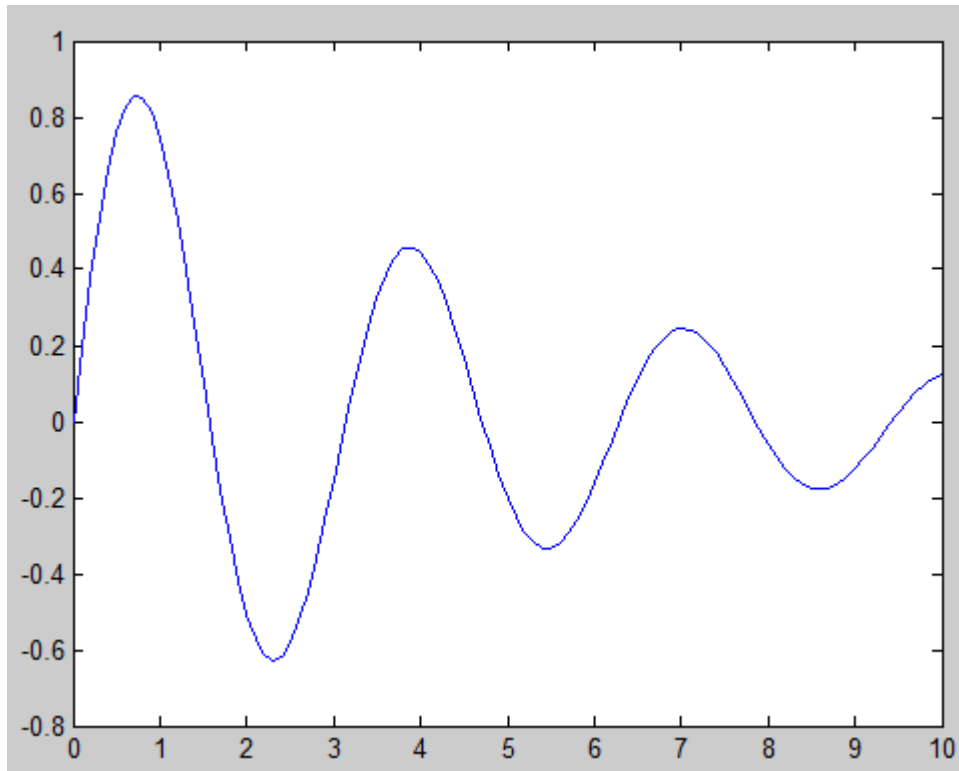
Οι μεθοδολογίες που θα χρησιμοποιηθούν είναι : Αρχικά η μέθοδος anfis όπου παράγει ένα εκπαιδευμένο και βελτιστοποιημένο σύστημα , η συνάρτηση genfis2 όπως την είδαμε σε προηγούμενο κεφάλαιο και τέλος μία υβριδική μέθοδος που κάνει χρήση της ομαδοποίησης Gustafson Kessel και Big Bang Big Crunch .

6.1 Υβριδική μέθοδος GK-BBBC

Όπως τονίσαμε και προηγουμένως στη συγκεκριμένη μέθοδο χρησιμοποιούμε αρχικά τη μέθοδο ομαδοποίησης Gustafson Kessel για τη δημιουργία του αρχικού συστήματος ασαφούς λογικής και στη συνέχεια χρησιμοποιούμε την μέθοδο Big Bang Big Crunch όπως στο κεφάλαιο 6 για βελτιστοποίηση του εύρους των συναρτήσεων συμμετοχής των εισόδων.

6.2 Μοντελοποίηση της συνάρτησης $f(x) = \sin(2x) / e^{\frac{x}{5}}$

Η συγκεκριμένη συνάρτηση μοντελοποιείτε στο διάστημα [0 10] με βήμα 0.1 . Η συνάρτηση είναι ημιτονοειδής και μέσα σε αυτό το διάστημα περιέχει πολλά τοπικά ακρότατα πράγμα το οποίο κάνει την μοντελοποίηση της δύσκολη . Στο παρακάτω διάγραμμα φέεται η συμπεριφορά της συνάρτησης.



Σχήμα 6.1

6.2 α Μοντελοποίηση με τη μέθοδο anfis

Η μέθοδος anfis δημιουργεί από ένα σετ δεδομένων ένα ασαφές σύστημα τύπου Sugeno το οποίο στη συνέχεια εκπαιδεύει με τη μέθοδο ελαχίστων τετραγώνων και οπισθόδρομης διάδοσης (Back Propagation).

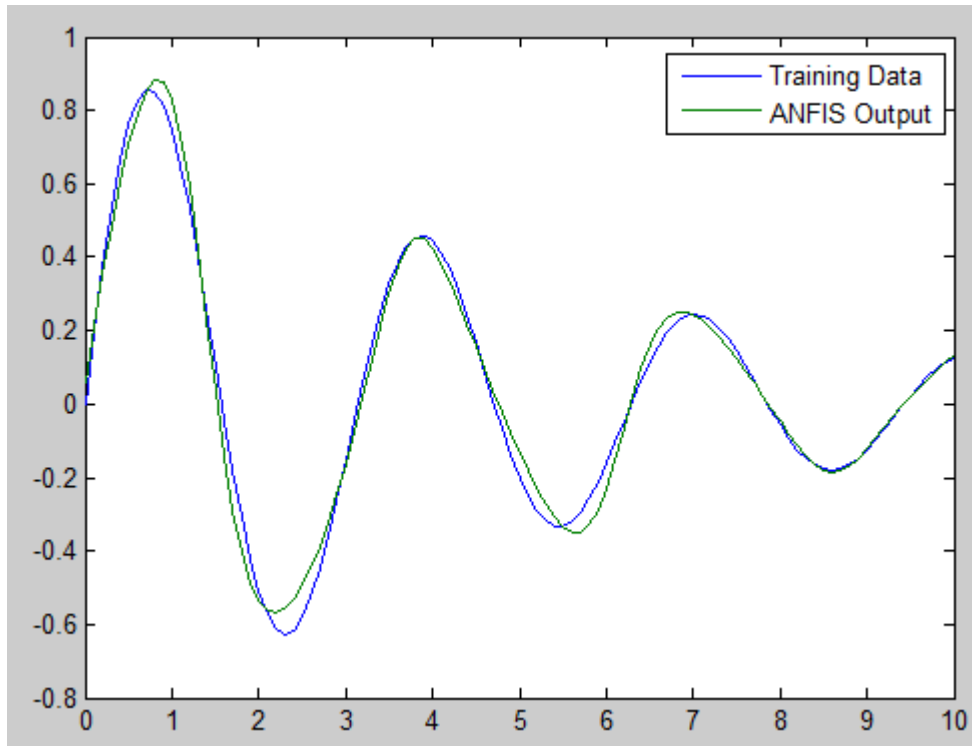
Η σύνταξη της συντάσσεται ως εξής : `[fis,error,stepsize] = anfis(trnData)`

Όπου trnData είναι τα δεδομένα εκπαίδευσης υπό μορφή πίνακα $[N \times Y]$ όπου N το πλήθος των μετρήσεων και Y το πλήθος των μεταβλητών ανά μέτρηση .

fis: το βελτιστοποιημένο σύστημα ασαφούς λογικής

error : το τελικό μέσο τετραγωνικό σφάλμα

και stepsize : η μεταβολή του σφάλματος μεταξύ δύο εποχών εκπαίδευσης

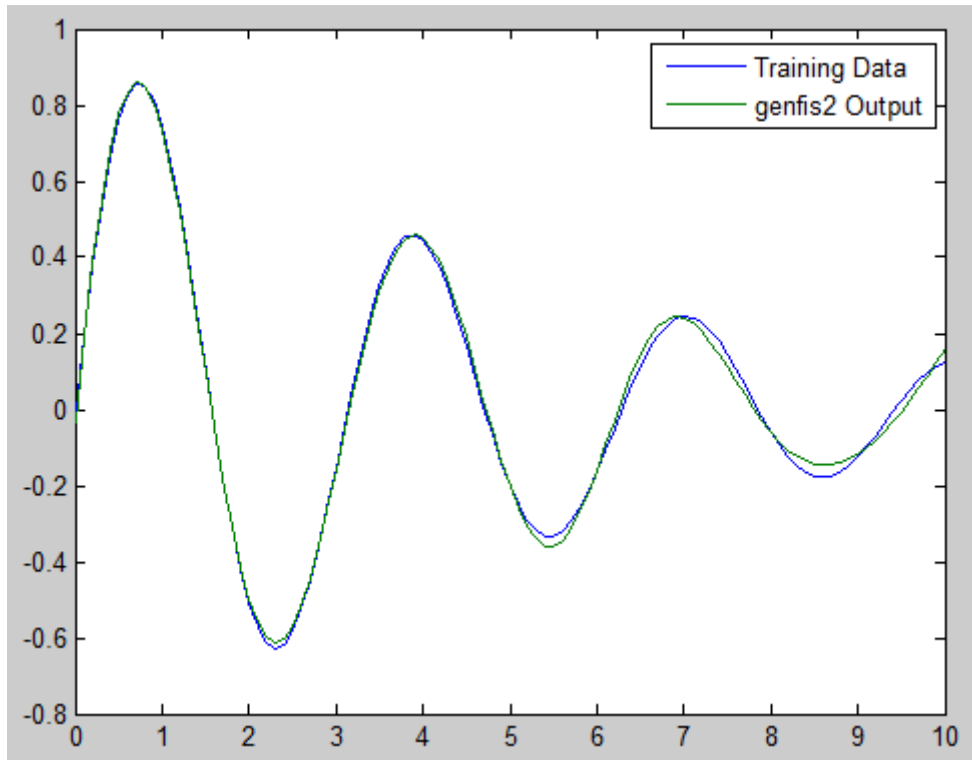


Σχήμα 6.2

Στο παραπάνω σχήμα με πράσινο φαίνεται η έξοδος του συστήματος ασαφούς λογικής. Όπως παρατηρούμε πλησιάζει πάρα πολύ την πραγματική τιμή πράγμα το οποίο φαίνεται και από το τετραγωνικό σφάλμα το οποίο έχει τιμή 0.042 . Σε αυτή τη περίπτωση το σύστημα ασαφούς λογικής έχει υπολογιστική ισχύ 5 κανόνων .

6.2.β Μοντελοποίηση με τη συνάρτηση genfis2

Όπως έχουμε αναφέρει και σε προηγούμενα κεφάλαια η συνάρτηση genfis2 δέχεται ως δεδομένο την ακτίνα επιρροής των κανόνων καθώς πραγματοποιεί Subtractive clustering . Στο συγκεκριμένο παράδειγμα χρησιμοποιούμε ακτίνα 0.4 για να έχουμε τον ίδιο αριθμό κανόνων με την μέθοδο anfis .



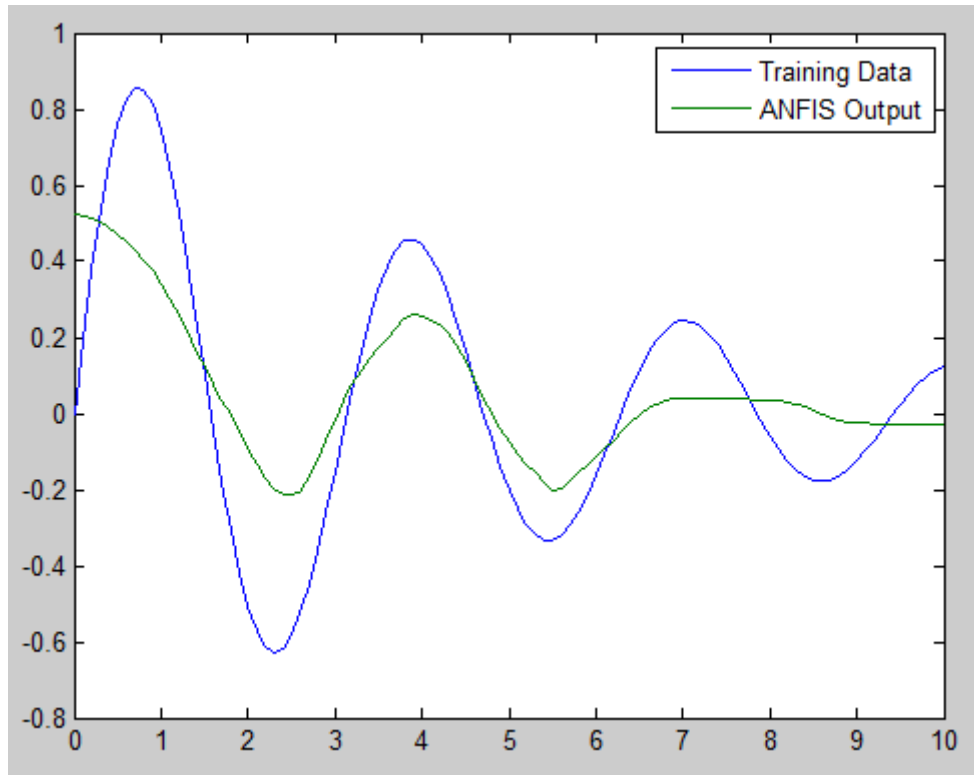
Σχήμα 6.3: Με πράσινο απεικονίζεται η έξοδος του fis

Όπως παρατηρούμε η έξοδος του fis έχει πολύ καλύτερη ανταπόκριση από εκείνη του συστήματος anfis με την ίδια υπολογιστική ισχύ 5 κανόνων . Το τετραγωνικό σφάλμα αυτή την περίπτωση είναι 0.0183.

6.2.γ Μοντελοποίηση με την υβριδική μέθοδος GK-BBCC

Σε αυτήν την περίπτωση το σύστημα ασαφούς λογικής δεν είναι αρκετά εκπαιδευμένο ώστε να παράγει ανταγωνιστικά αποτελέσματα σε σύγκριση με τις δύο προηγούμενες μεθόδους.

Παρόλα αυτά αποτελεί ένα πρώτο βήμα για εκπαίδευση και βελτιστοποίηση καθώς πέραν από τον παράγοντα που βελτιστοποιεί το BB-BC υπάρχουν πάρα πολλοί άλλοι που μπορούν να βελτιστοποιηθούν.



Σχήμα 7.4

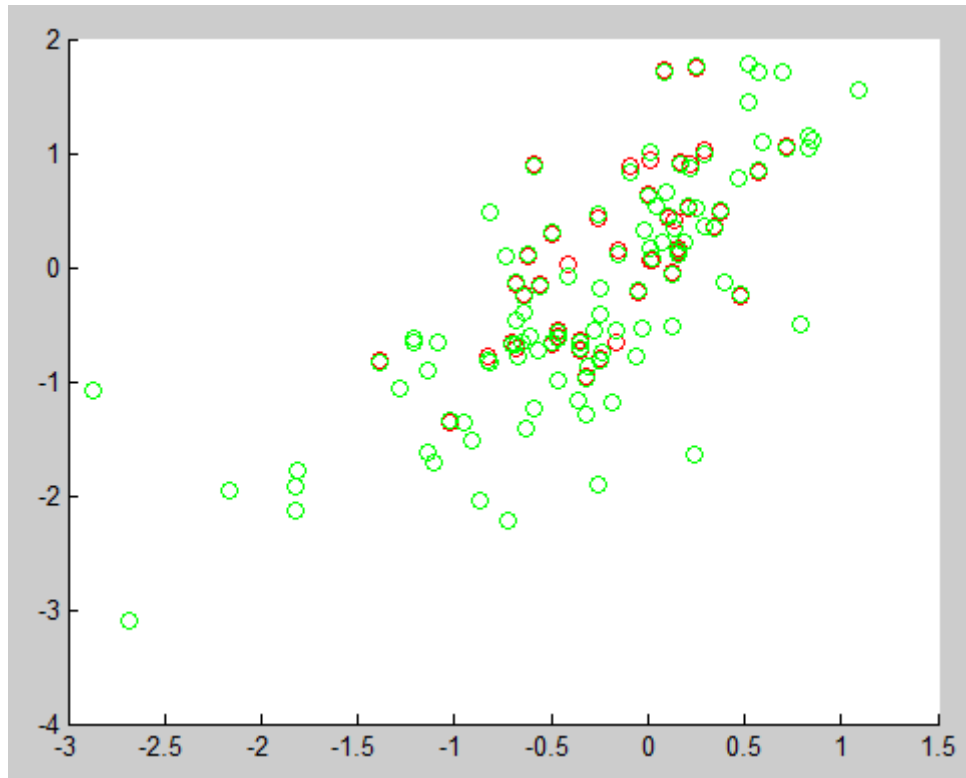
Εδώ φαίνεται καθαρά η έλλειψη ποιότητας σε σύγκριση με τις άλλες μεθόδους πράγμα το οποίο λογικό καθώς το σύστημα είναι τύπου Mamdani και μόνο με έναν παράγοντα εκπαιδευμένο. Αυτό που αποδεικνύει την επιτυχή βελτιστοποίηση είναι η μεταβολή του τετραγωνικού σφάλματος καθώς πέφτει από 0.2714 σε 0.2093.

6.3 Μοντελοποίηση των μετρήσεων ρυζιού

Ακολουθεί η ίδια διαδικασία για τα δεδομένα ρυζιού τα οποία είχαν παρουσιαστεί στο 4^ο κεφάλαιο.

6.3.α Μοντελοποίηση με τη μέθοδο anfis

Χρησιμοποιώντας με τον ίδιο τρόπο όπως προηγουμένως την συνάρτηση anfis έχουμε.



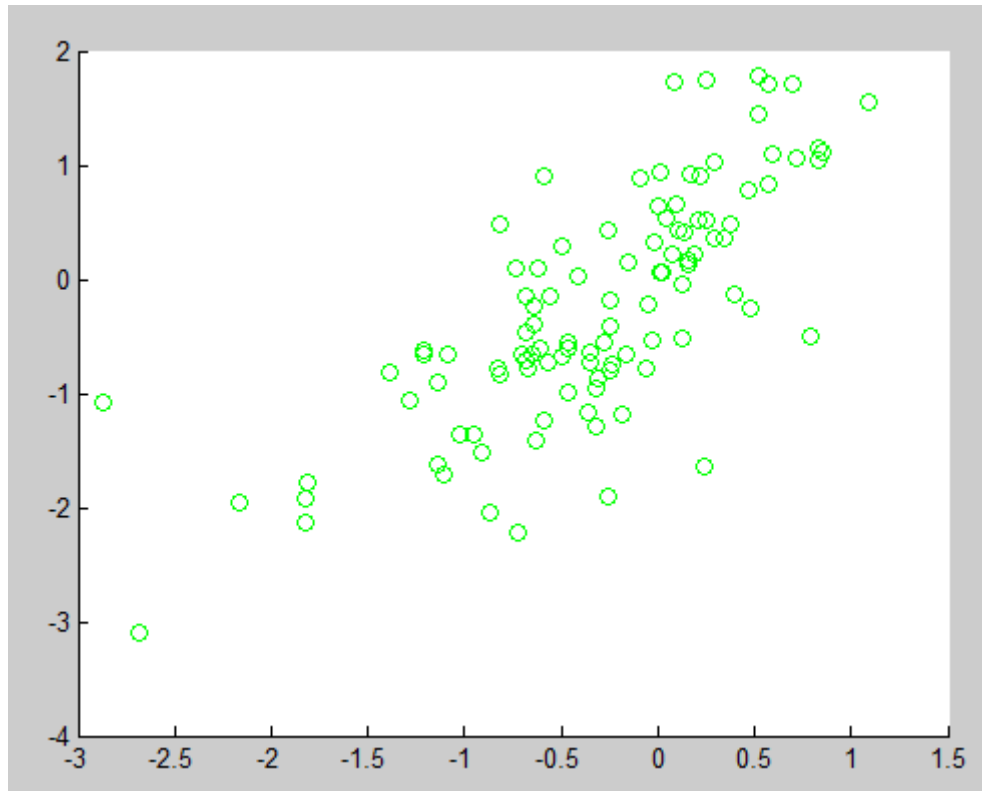
Σχήμα 6.5 : Με κόκκινο απεικονίζονται τα πραγματικά δεδομένα και με πράσινο τα αποτελέσματα του fis.

Εδώ πρέπει πάλι να τονίσουμε πως τα δεδομένα του ρυζιού είναι 5 διαστάσεων για αυτό το λόγο από τις 4 εισόδους έχει επιλεγεί μία και η έξοδος ώστε να γίνει εύκολη απεικόνιση

Όπως φαίνεται και από το διάγραμμα τα αποτελέσματα είναι αγγίζουν πολύ τα πραγματικά καθώς το τετραγωνικό σφάλμα είναι 0.023. Το μειονέκτημα σε αυτή την περίπτωση είναι η μεγάλη υπολογιστική ισχύς του συστήματος ασαφούς λογικής καθώς περιέχει 32 κανόνες.

6.3.β Μοντελοποίηση με τη συνάρτηση genfis2

Ακολουθεί η ίδια διαδικασία με τη συνάρτηση genfis2 όπως ακριβώς και στο κεφάλαιο 4 .

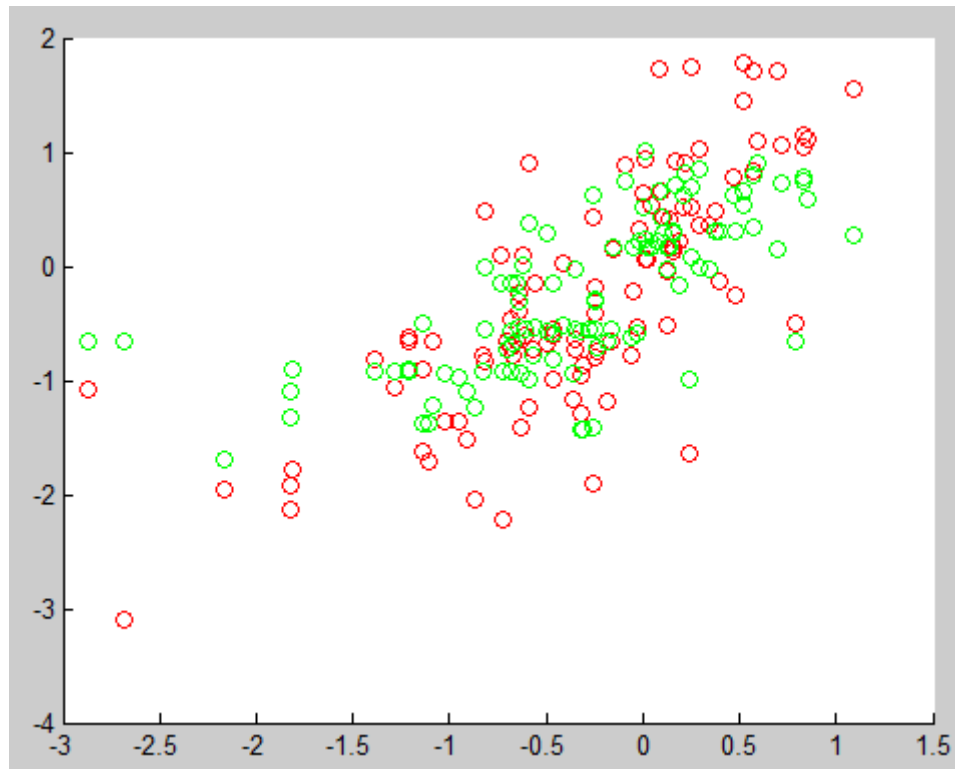


Σχήμα 6.6: Τα δεδομένα του *fis* δεν φαίνονται καθώς έχουν ταυτιστεί πλήρως με τα πραγματικά.

Όπως φαίνεται και από το διάγραμμα έχουμε πλήρης ταυτοποίηση της εξόδου του *fis* με τα πραγματικά δεδομένα . Το τετραγωνικό σφάλμα αυτή την περίπτωση είναι 0 .

6.3.γ Μοντελοποίηση με την υβριδική μέθοδο GK-BBVC

Σε αυτή την περίπτωση τα αποτελέσματα είναι :



Σχήμα 6.7 : Εδώ με κόκκινο παρουσιάζονται τα πραγματικά δεδομένα και με πράσινο τα αποτελέσματα της μοντελοποίησης.

Το σφάλμα όπως ήταν αναμενόμενο είναι πολύ μεγαλύτερο από τις προηγούμενες μεθόδους παρόλα αυτά κάνοντας τη σύγκριση πριν και μετά τη βελτιστοποίηση BB-BC το μέσο τετραγωνικό σφάλμα μειώνεται από 0.6020 σε 0.5320 , πράγμα το οποίο χαρακτηρίζει την βελτιστοποίηση επιτυχής.

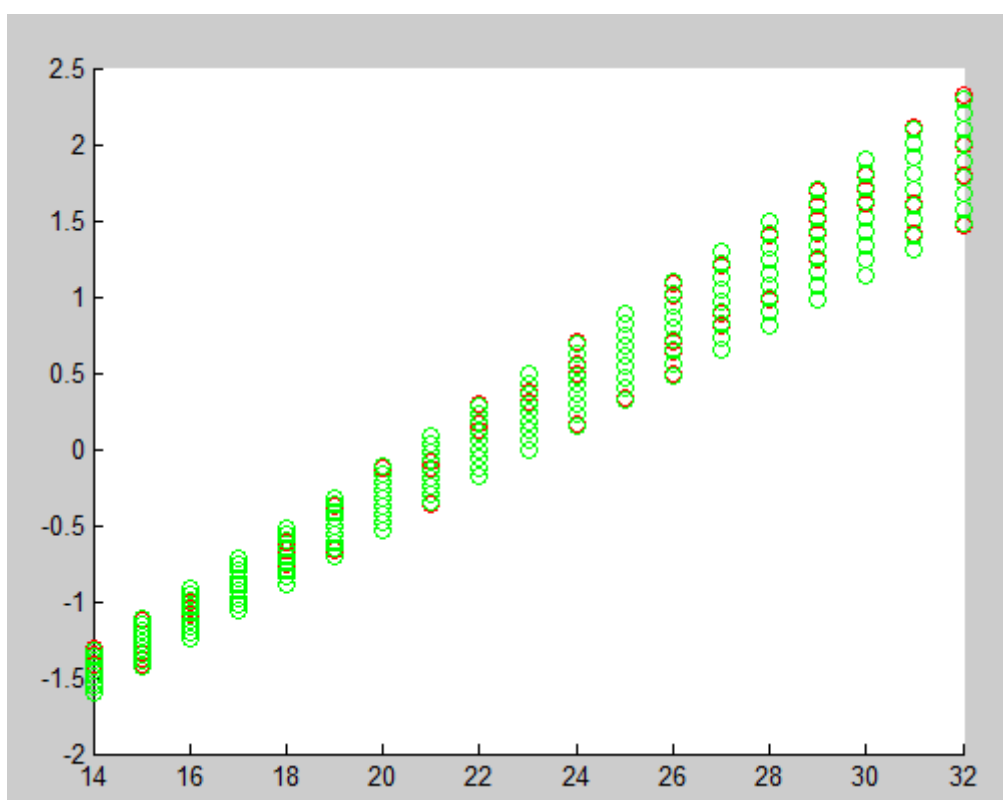
Το σύστημα σε αυτή την περίπτωση επιλέξαμε να έχει 20 κανόνες.

6.4 Μοντελοποίηση PMV

Ακολουθεί η ίδια διαδικασία για τα δεδομένα ρυζιού τα οποία είχαν παρουσιαστεί στο 4^ο κεφάλαιο. Πρέπει και εδώ να σημειωθεί πως για την απεικόνιση των δεδομένων, από τις 3 διαστάσεις έχουν επιλεγεί 2 : η πρώτη είσοδος και η έξοδος.

6.4.α Μοντελοποίηση με τη μέθοδο anfis

Τα αποτελέσματα είναι:

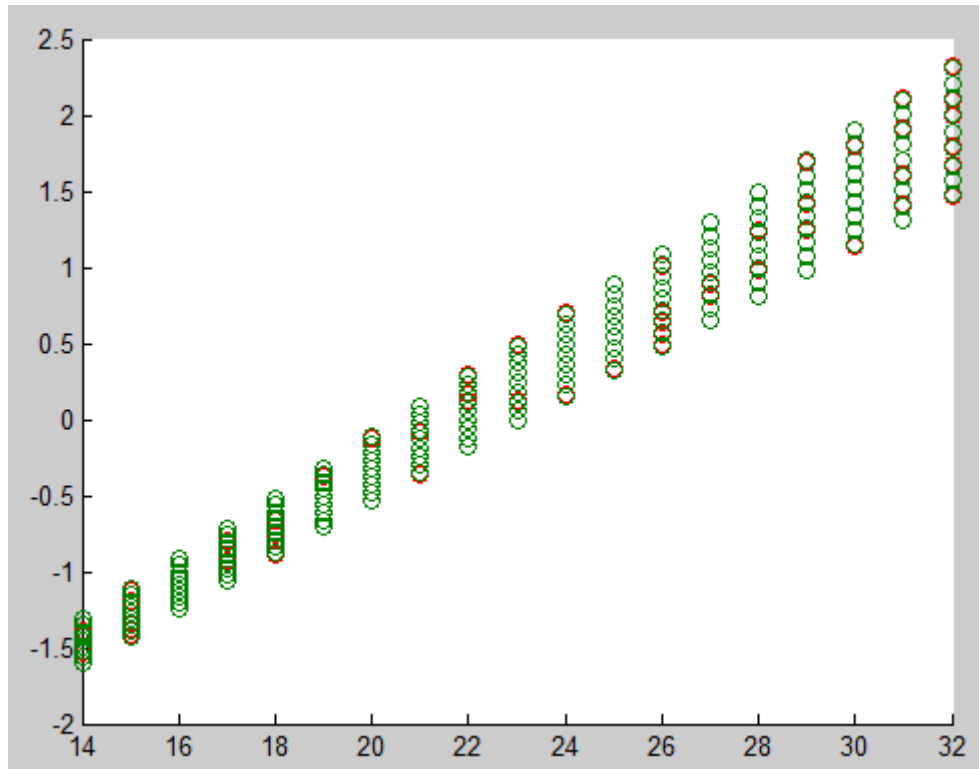


Σχήμα 6.8: Με κόκκινο παρουσιάζονται τα πραγματικά δεδομένα του pmv ενώ με πράσινο η έξοδος του fis .

Το τετραγωνικό σφάλμα σε αυτή τη περίπτωση είναι 0.0052 με 4 κανόνες.

6.4.β Μοντελοποίηση με τη μέθοδο genfis2

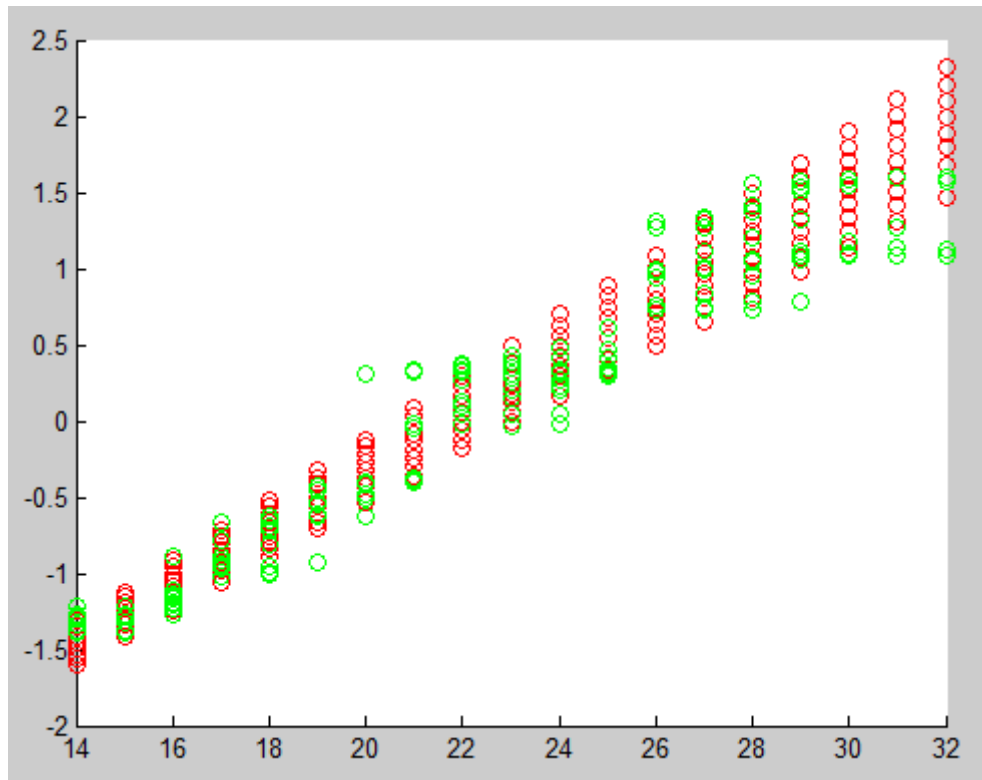
Τα αποτελέσματα σε αυτή την περίπτωση είναι :



Σχήμα 6.9: Κόκκινο για τα δεδομένα εκπαίδευσης και πράσινο για την έξοδο του fis.

Τώρα δημιουργήσαμε ένα fis παρόμοιας δύναμης με το προηγούμενο. Το τετραγωνικό σφάλμα που προκύπτει είναι 0.0050 και το fis έχει υπολογιστική ισχύ 5 κανόνων.

6.4.γ Μοντελοποίηση με την υβριδική μέθοδο GK-BBVC



Σχήμα 6.10 : Κόκκινο χρώμα για τα δεδομένα του ρ_n και πράσινο για την έξοδο του fis .

Όπως ήταν αναμενόμενο και σε αυτήν την περίπτωση το σφάλμα είναι πολύ μεγαλύτερο από τις προηγούμενες μεθόδους. Αλλά όμως και πάλι φαίνεται η βελτιστοποίηση καθώς το σφάλμα μειώθηκε από 0.3875 σε 0.2049.

6.5 Γενικές παρατηρήσεις

Στον πίνακα που ακολουθεί παρουσιάζονται οι επιδόσεις των μεθοδολογιών για κάθε μία από τις 3 περιπτώσεις και στη συνέχεια σχολιάζονται τα αποτελέσματα.

Περίπτωση/Μέθοδος	Anfis	Genfis2	GK-BBBC
Συνάρτηση	0.042/ 5 Κανόνες	0.018 /5 Κανόνες	0.20 /10 Κανόνες
Ρύζι	0.023/ 32 Κανόνες	0/23 Κανόνες	0.53/20 Κανόνες
PMV	0.0052/ 4 Κανόνες	0.0050/ 5 Κανόνες	0.20 /20 Κανόνες

Παρατηρούμε πως στις δύο περιπτώσεις η μέθοδος genfis2 είναι αισθητά καλύτερη από τη μέθοδο anfis αλλά στην τρίτη περίπτωση συμβαίνει το ανάποδο. Βλέπουμε πως η μέθοδος anfis με έναν λιγότερο κανόνα παράγει ένα σύστημα ασαφούς λογικής το ίδιο αξιόπιστο.

Όσον αφορά την τρίτη μέθοδο ο αριθμός των κανόνων είναι καθαρά πειραματικός καθώς η μέθοδος ομαδοποίηση Gustafson Kessel δεν υπολογίζει αριθμό κεντρώων αλλά τον δέχεται από τον χρήστη επομένως είναι ακόμα ένας παράγοντας που δύναται βελτιστοποίησης.

Παράρτημα Α

Α.1 Συνάρτηση Lagrange πολλών μεταβλητών.

Για το ακόλουθο μη γραμμικό πρόβλημα βελτιστοποίησης με περιορισμούς ισότητας .

$$\min_{x_1, \dots, x_n} f(x_1, \dots, x_n)$$

Όπου ισχύει ο περιορισμός

$$g_j(x) = b_j \quad \text{όπου } j = 1, 2, \dots, m$$

Η συνάρτηση Lagrange γράφεται :

$$L(X, \lambda) = f(X) + \sum_{j=1}^m \lambda [g_j(x) - b_j]$$

όπου :

$$X = (x_1, \dots, x_n)$$

$$\lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)$$

Αναγκαίες συνθήκες για ευσταθή σημεία στην L

$$\frac{\partial L}{\partial x_i} = 0 \quad \text{ή} \quad \frac{\partial f(x)}{\partial x_i} + \sum_{j=1}^m \lambda \frac{\partial g_j(x)}{\partial x_i} = 0$$

$$\frac{\partial L}{\partial \lambda_k} = 0 \quad \text{ή} \quad \frac{\partial f(x)}{\partial x_i} + \sum_{j=1}^m [g_j(x) - b_j] \frac{\partial \lambda_i}{\partial \lambda_k}$$

Παρατηρείτε ότι $\frac{\partial \lambda_i}{\partial \lambda_k} = \delta_{kj}$

όπου $\delta_{kj} = \begin{cases} 1 & \text{για } j = k \\ 0 & \text{για } j \neq k \end{cases}$

A2 Λειτουργία Genfis2.m

Η συνάρτηση Genfis2.m αποτελεί ένα πολύ ισχυρό εργαλείο για την παραγωγή συστημάτων ασαφούς λογικής από ένα σετ εισόδων-εξόδων με την χρήση της μεθόδου ομαδοποίησης Subtractive clustering.

Αρχικά εφαρμόζοντας την μέθοδο Subtractive clustering εντοπίζονται τα κέντρα των cluster και συντάσσονται οι κανόνες. Διάφοροι παράμετροι όπως το εύρος (σ) των γκαουσιανών συναρτήσεων συμμετοχής και οι έξοδοι των κανόνων υπολογίζονται από τη γραμμική μέθοδο ελαχίστων τετραγώνων.

Η συνάρτηση συντάσσετε ως: FIS = GENFIS2(XIN,XOUT,RADII)

Το fis που παράγει είναι ένα Sugeno τύπου σύστημα ασαφούς λογικής, XIN είναι τα δεδομένα εισόδου και XOUT τα δεδομένα εξόδου στα οποία θέλουμε να προσαρμοστεί το σύστημα. Η μεταβλητή RADII δέχεται τιμές από 0 έως 1 και συμβολίζει την ακτίνα επιρροής για το subtractive clustering. Μπορεί να έχει διαστάσεις [1xN] δηλαδή κάθε διάσταση-τύπος δεδομένων να έχει διαφορετική ακτίνα επιρροής.

Η σύνταξη FIS = GENFIS2(...,XBOUNDS) μας δίνει την δυνατότητα να αλλάξουμε την κλίμακα στην οποία θα κανονικοποιηθεί η κάθε είσοδο ή έξοδο. Η προκαθορισμένη κλίμακα όπως και στο subtractive clustering είναι [0,1] αυτό αλλάζει μεταβάλλοντας τις τιμές της τιμής XBOUNDS η οποία έχει δομή [2xN] όπου N ο αριθμός των τύπων των δεδομένων. Στην ουσία ορίζουμε για κάθε τύπο δεδομένων τα όρια στα οποία θέλουμε να κανονικοποιηθεί.

Στη σύνταξη FIS = GENFIS2(...,XBOUNDS,OPTIONS) το πεδίο OPTIONS επιτρέπει αλλαγές σε κάποιες προκαθορισμένες σταθερές του αλγορίθμου. Το πεδίο OPTIONS είναι το ίδιο για τον αλγόριθμο subtractive clustering και περισσότερες πληροφορίες φαίνονται με την εντολή HELP SUBCLUST.

FIS = GENFIS2(...,XBOUNDS,OPTIONS,USER_CENTERS) Στο πεδίο USER_CENTERS ο χρήστης έχει τη δυνατότητα να ορίσει τα αρχικά cluster centers. Έχει διαστάσεις [CxN] όπου C ο αριθμός των cluster και N το πλήθος των τύπων δεδομένων.

A3 Δεδομένα ρυζιού

Γεύση (x1)	Εμφάνιση (x2)	Γεύση (x3)	Κολλώδες (x4)	Σκληράδα (x5)	Γενική αξιολόγηση (y)
0,523	0,913	1,571	1,6	0	1,784
0,699	1,543	1,76	1,944	-0,875	1,706
-0,720	-2,0220	-1,733	-1,864	1,200	-2,217
-0,309	-0,377	-0,766	-0,96	0,326	-0,875
0,192	0,885	-0,166	-0,106	0,234	0,218
-0,163	-0,094	-0,315	0,083	0,034	-0,655
-0,274	0,044	-0,194	-0,401	1,57	-0,545
0,291	0,984	0,935	0,68	-0,652	1,033
-2,687	-2,853	-3,067	-1,75	2,392	-3,101
-0,704	-0,424	-0,518	-0,779	0,052	-0,652
1,094	1,987	1,613	1,514	-1,156	1,552
-0,247	0,334	0,122	-0,705	0,694	-0,416
-0,673	0,032	-0,838	-0,837	0,354	-0,784
-1,103	-1,046	-1,678	-1,335	-0,46	-1,71
-0,033	0,162	-0,541	-0,159	0,616	-0,529
-0,351	0,119	-0,786	-0,435	0,129	-0,635
0,594	0,266	0,885	1	-0,548	1,105
-0,156	0,173	0,272	0,092	0,163	0,145
-0,256	-0,593	-1,012	-1,403	1,165	-1,901
-0,183	-0,496	-0,77	-1,538	1,78	-1,188
-0,871	-1,27	-1,655	-1,307	1,281	-2,037
-0,060	-0,11	-0,95	-0,719	1,159	-0,788
-1,135	0,163	-0,47	0,285	-0,023	-0,911
-0,609	-0,215	-0,681	-0,534	0,963	-0,612
0,393	0,388	-0,232	0,154	-0,42	-0,127
-2,163	-0,911	-1,61	-1,328	0,145	-1,952
0,831	0,6	0,953	0,953	-0,163	1,141
0,081	1,712	0,946	0,678	-0,257	1,723
0,836	0,799	0,782	1,529	-0,769	1,038
0,214	0,544	0,679	0,314	-0,347	0,51
-0,827	-0,759	-1,012	-0,517	0,197	-0,788
0,221	0,708	0,737	0,923	-0,495	0,899
-0,558	-0,343	-0,331	0,266	-0,756	-0,157
-0,684	-0,33	-0,185	0,349	-0,545	-0,155
-1,816	-0,952	-1,409	-1,882	0,639	-1,783
0,013	1,01	0,927	1,135	-0,387	0,937
0,344	0,856	0,37	0,327	0,227	0,36

-0,323	-0,772	-0,895	-1,002	0,517	-1,286
-0,82	0,608	-0,288	-0,311	0,386	-0,837
0,24	-0,906	-0,715	-1,467	1,626	-1,646
-0,571	-0,151	-0,635	-0,093	0,956	-0,73
0,789	0,821	-0,769	0,428	-1,815	-0,508
-0,353	0,115	-0,394	0,722	-0,089	-0,731
-0,467	-0,622	-0,661	-0,302	0,698	-0,986
-0,491	0,021	-0,221	-0,563	0,385	-0,672
0,848	0,837	1,54	1,112	-0,475	1,113
0,099	0,12	0,387	1,252	-0,638	0,655
-0,738	0,989	0,415	0,574	-0,647	0,093
-0,82	-0,702	1,203	1,579	-1,212	0,483
-0,629	-1,049	-1,271	-1,116	1,637	-1,407
-0,593	-0,898	-0,883	-0,647	0,323	-1,235
0,14	0,181	0,06	0,136	-0,234	0,416
-1,022	-0,827	-1,294	-0,583	0,437	-1,357
-0,649	-0,288	0,257	0,327	-0,412	-0,653
-0,413	0,026	-0,108	0,393	-0,291	0,019
-0,358	-0,106	-1,275	-1,059	0,505	-1,168
-1,825	-2,018	-1,867	-1,517	0,114	-2,136
-0,002	-0,144	0,545	0,674	-0,111	0,641
0,158	0,163	0,03	0,359	-0,128	0,135
0,516	0,73	1,218	1,662	-0,366	1,44
-0,235	-0,566	-0,289	-0,763	0,048	-0,749
-1,085	-0,614	-1,24	-1,12	0,958	-0,65
-1,283	0,09	-0,651	-0,19	-0,131	-1,067
0,25	0,232	0,378	1,983	-1,231	0,519
0,288	0,205	0,147	0,399	0,365	0,352
0,567	1,502	0,824	0,53	-0,387	0,842
0,011	0,26	0,102	0,487	-0,402	0,067
-0,912	-0,911	-1,448	-1,666	1,375	-1,524
-0,588	0,792	1,023	0,995	-0,489	0,901
-0,686	-0,682	-0,514	-0,088	-0,43	-0,466
-2,871	0,59	-0,117	0,195	0,051	-1,084
-1,208	-0,257	-0,426	-0,237	0,24	-0,628
-1,819	-0,856	-1,758	-1,014	1,094	-1,925
0,713	1,172	0,648	1,065	-0,886	1,053
-0,248	0,185	0,204	-0,41	0,588	-0,186
-0,638	0,462	-0,149	-0,214	0,389	-0,23
0,255	0,98	0,837	1,64	0,028	1,747
-1,212	-0,074	-0,317	-0,144	0,18	-0,658
-0,093	0,712	0,594	1,27	-0,236	0,88
-0,315	-0,352	-0,741	-0,288	0,201	-0,964
0,021	0,805	0,367	0,19	0,33	0,056
0,158	0,265	0,053	0,373	-0,101	0,164
0,372	0,388	0,448	0,291	-0,402	0,489
-0,47	-0,301	-0,481	0,173	0,263	-0,546
-0,244	-0,044	-0,466	-0,464	0,469	-0,804
-0,049	0,401	-0,097	0,062	0,282	-0,223
-0,639	0,63	-0,089	0,166	-0,293	-0,394

0,039	0,524	0,207	1,082	-1,3030	0,5370
0,079	1,231	0,373	0,281	-1,273	0,218
0,104	-0,194	0,261	0,838	0,085	0,434
0,475	0,707	-0,172	0,097	-0,247	-0,246
0,468	0,333	0,511	1,188	-0,646	0,78
-0,465	-0,195	-0,491	-0,671	0,837	-0,614
0,125	0,127	-0,264	0,461	0,197	-0,046
-0,491	0,525	0,419	0,78	-0,512	0,291
-1,388	-0,114	-0,894	-0,168	0,028	-0,821
-0,955	-1,146	-1,107	-0,922	0,106	-1,367
-0,255	0,232	0,517	0,398	-0,247	0,424
-0,686	-0,252	-0,749	-0,428	0,048	-0,704
-0,016	0,674	-0,046	0,662	-0,694	0,326
-0,618	-0,425	0,098	0,471	0,158	0,104
0,57	1,542	1,302	1,842	-0,563	1,712
-1,134	-1,281	-1,031	-1,477	1,345	-1,624
0,124	0,588	0,053	-0,405	-0,322	-0,517
0,163	1,088	1,074	0,683	-0,135	0,913

Α 4 Δεδομένα pmv

Θερμοκρασία	Ποσοστό Υγρασίας (%)	Τιμή pmv
14	20	-1,6
14	30	-1,56
14	40	-1,53
14	60	-1,46
14	80	-1,39
14	90	-1,35
14	100	-1,31
15	20	-1,42
15	30	-1,38
15	40	-1,34
15	60	-1,27
15	80	-1,19
15	90	-1,15
15	100	-1,12
16	20	-1,24
16	30	-1,2
16	40	-1,16
16	60	-1,08
16	80	-1
16	90	-0,96
16	100	-0,92
17	20	-1,06
17	30	-1,02
17	40	-0,98
17	60	-0,89
17	80	-0,8
17	90	-0,76
17	100	-0,72
18	20	-0,89
18	30	-0,84
18	40	-0,79
18	60	-0,7

18	80	-0,61
18	90	-0,56
18	100	-0,52
19	20	-0,71
19	30	-0,66
19	40	-0,61
19	60	-0,51
19	80	-0,41
19	90	-0,37
19	100	-0,32
20	20	-0,53
20	30	-0,48
20	40	-0,43
20	60	-0,32
20	80	-0,22
20	90	-0,17
20	100	-0,12
21	20	-0,36
21	30	-0,3
21	40	-0,24
21	60	-0,13
21	80	-0,02
21	90	0,03
21	100	0,09
22	20	-0,18
22	30	-0,12
22	40	-0,06
22	60	0,06
22	80	0,17
22	90	0,23
22	100	0,29
23	20	0
23	30	0,06
23	40	0,12
23	60	0,25
23	80	0,37
23	90	0,43
23	100	0,5
24	20	0,16
24	30	0,23
24	40	0,3
24	60	0,43
24	80	0,56
24	90	0,63
24	100	0,7
25	20	0,33
25	30	0,4
25	40	0,47
25	60	0,61
25	80	0,75

25	90	0,82
25	100	0,89
26	20	0,49
26	30	0,56
26	40	0,64
26	60	0,79
26	80	0,94
26	90	1,01
26	100	1,09
27	20	0,65
27	30	0,73
27	40	0,81
27	60	0,97
27	80	1,13
27	90	1,21
27	100	1,29
28	20	0,81
28	30	0,9
28	40	0,98
28	60	1,15
28	80	1,32
28	90	1,4
28	100	1,49
29	20	0,98
29	30	1,07
29	40	1,16
29	60	1,34
29	80	1,51
29	90	1,6
29	100	1,69
30	20	1,14
30	30	1,24
30	40	1,33
30	60	1,52
30	80	1,71
30	90	1,8
30	100	1,9
31	20	1,31
31	30	1,41
31	40	1,51
31	60	1,71
31	80	1,91
31	90	2,01
31	100	2,11
32	20	1,47
32	30	1,57
32	40	1,68
32	60	1,89
32	80	2,1
32	90	2,21

32	100	2,32
14	50	-1,49
14	70	-1,42
15	50	-1,31
15	70	-1,23
16	50	-1,12
16	70	-1,04
17	50	-0,93
17	70	-0,85
18	50	-0,75
18	70	-0,66
19	50	-0,56
19	70	-0,46
20	50	-0,38
20	70	-0,27
21	50	-0,19
21	70	-0,08
22	50	0
22	70	0,12
23	50	0,18
23	70	0,31
24	50	0,36
24	70	0,5
25	50	0,54
25	70	0,68
26	50	0,71
26	70	0,86
27	50	0,89
27	70	1,05
28	50	1,07
28	70	1,24
29	50	1,25
29	70	1,42
30	50	1,43
30	70	1,61
31	50	1,61
31	70	1,81
32	50	1,79
32	70	2

Παράρτημα Β

Β.1 Αλγόριθμος Gustafson Kessel

Είσοδοι:

```
load fcmdata.dat
Z=fcmdata;
tol=0.001;
m=2;
c=3;
```

Προετοιμασία μητρώων:

```
[N,n]=size(Z);
N1=ones(N,1); n1=ones(n,1); c1=ones(1,c);
U=zeros(N,c);
d=U;
F=zeros(n,n,c);
```

Αρχικοποίηση του πίνακα U:

```
minZ=c1'*min(Z); maxZ=c1'*max(Z);
V=minZ+(maxZ-minZ).*rand(c,n);
for j=1:c,
    ZV=Z-N1*V(j,:);
    d(:,j)=sum((ZV.^2)')';
end;
d=(d+1e-10).^(-1/(m-1));
U0=(d./(sum(d)')*c1);
```

Επανάληψη:

```
while max(max(U0-U))>tol
    U=U0; Um=U.^m; sumU=sum(Um);
    V=(Um'*Z)./(n1*sumU)';
    for j=1:c,
        ZV=Z-N1*V(j,:);
        f=n1*Um(:,j)'.*ZV'*ZV/sumU(j);
        d(:,j)=sum(ZV*(det(f)^(1/n)*inv(f)).*ZV,2);
    end;
    jk=d;
    d=(d).^(-1/(m-1));
    U0=(d./(sum(d)')*c1);
end
```

Δημιουργία τελικού F και U:

```
U=U0; Um=U.^m; sumU=n1*sum(Um);
for j=1:c,
    ZV=Z-N1*V(j,:);
    F(:,j)=n1*Um(:,j)'.*ZV'*ZV/sumU(1,j);
```

Εύρεση των cluster και προβολή τους σε διάγραμμα :

```
U=U';
maxU = max(U);
index1 = find(U(1, :) == maxU);
index2 = find(U(2, :) == maxU);
index3 = find(U(3, :) == maxU);
```

figure

```
line(fcndata(index1, 1), fcndata(index1, 2), 'linestyle', ...  
'none', 'marker', 'o', 'color', 'g');  
line(fcndata(index2, 1), fcndata(index2, 2), 'linestyle', ...  
'none', 'marker', 'x', 'color', 'r');  
line(fcndata(index3, 1), fcndata(index3, 2), 'linestyle', ...  
'none', 'marker', '+', 'color', 'b');  
hold on  
plot(V(1,1),V(1,2), 'ko', 'markersize', 15, 'LineWidth', 2)  
plot(V(2,1),V(2,2), 'kx', 'markersize', 15, 'LineWidth', 2)  
plot(V(3,1),V(3,2), 'k+', 'markersize', 15, 'LineWidth', 2)
```

Ανάλυση του αλγορίθμου

Z: Πίνακας $N \times n$ όπου N είναι ο αριθμός των δεδομένων και n ο αριθμός των σετ δεδομένων.

C: ο αριθμός των clusters (στην περίπτωση μας χρησιμοποιούνται 3)

m: εκθέτης ασάφειας (η τιμή του συνήθως είναι 2 και καθορίζει πόσο 'soft' θα είναι η ταξινόμηση).

Tol: σφάλμα που χρησιμοποιείται σαν κριτήριο για τον τερματισμό του αλγορίθμου.

$N1 = \text{ones}(N, 1)$; Πίνακας με άσους με μία στήλη και N γραμμές $n1 = \text{ones}(n, 1)$;

Παρομοίως με μία στήλη και n γραμμές

$c1 = \text{ones}(1, c)$; Πίνακα με άσους που έχει μία γραμμή και c στήλες

$U = \text{zeros}(N, c)$; Πίνακας βαθμών συμμετοχής έχει γραμμές όσες τα δεδομένα και στήλες όσες τα clusters.

$d = U$; Πίνακας αποστάσεων.

$F = \text{zeros}(n, n, c)$; Πίνακας συνδιακίμανσης (covariance matrix) είναι τρισδιάστατος $n \times n$ για κάθε cluster .

$\min Z = c1' * \min(Z)$; $\max Z = c1' * \max(Z)$; Εδώ υπολογίζονται οι οριακές τιμές των δεδομένων εισόδου.

$V = \min Z + (\max Z - \min Z) * \text{rand}(c, n)$; Δημιουργείται ο πίνακας των συντεταγμένων των cluster όπου δίνεται μια τυχαία τιμή και στη συνέχεια κανονικοποιήτε στα όρια που υπολογίστηκαν παραπάνω. Είναι πίνακας $c \times n$ και στη περίπτωση μας έχει 3

γραμμές(αριθμός των cluster) και 2 στήλες (αριθμός σετ δεδομένων). Άρα κάθε γραμμή περιέχει τις συντεταγμένες των clusters.

for j=1:c, Για κάθε cluster.

ZV=Z-N1*V(j,:); Για κάθε j ο πίνακας V(j,:) είναι πίνακας 1xη και ο N1 είναι πίνακας Nx1 επομένως το γινόμενο τους είναι πίνακας Nxη ο οποίος στη συνέχεια αφαιρείται από τον Z και δίνει τη διαφορά ZV

d(:,j)=sum((ZV.^2)'); Τετραγωνίζονται όλα τα στοιχεία του πίνακα ZV και στη συνέχεια προστίθενται τα στοιχεία της κάθε στήλης ανά γραμμή.

Τα στοιχεία καταχωρούνται σε μία στήλη και αντιπροσωπεύουν την ευκλείδεια απόσταση του κάθε σημείου από τον cluster της αντίστοιχης στήλης .

d=(d+1e-100).^(-1/(m-1)); Προσθέτει μια μικρή τιμή(κοντά στο 0) στις αποστάσεις και στη συνέχεια τις αντιστρέφει.

U0=(d./(sum(d)'+c1)); Δημιουργία του πίνακα συμμετοχής. Αρχικά προσθέτοντας τα στοιχεία κάθε στήλης ανά γραμμή δημιουργούμε έναν Nx1 πίνακα ,στη συνέχεια πολλαπλασιάζοντας με τον c1 όπου είναι 1xη το γινόμενο τους είναι ένας πίνακας Nxη.Τέλος διαιρούμε κάθε στοιχείο του d με τον παραπάνω πίνακα.

while max(max(U0-U))>tol Το εσωτερικό max εκφράζει το μεγαλύτερο αποτέλεσμα της αφαίρεσης ένα για κάθε στήλη , έπιτα το εξωτερικό επιλέγει τη μεγαλύτερη τιμή από τον πίνακα 1xη

U=U0;Ενημέρωση του πίνακα συμμετοχής

Um=U.^m; Υψώνεται κάθε στοιχείο του U με τον εκθέτη ασαφοποίησης
sumU=sum(Um) Προστίθενται όλα τα στοιχεία του Um ανά στήλη με αποτέλεσμα έναν 1xη πίνακα.

V=(Um'*Z)./(n1*sumU)';Ενημέρωση των κεντρώων των cluster .Το γινόμενο (Um'*Z) δίνει αποτέλεσμα έναν cηη πίνακα όπου κάθε στοιχείο του διαιρείτε με το γινόμενο (n1*sumU)' που είναι επίσης cηη πίνακας.

for j=1:c, για κάθε cluster.

ZV=Z-N1*V(j,:)Ενημέρωση της διαφοράς ZV όπως στο προηγούμενο βήμα.

f=n1*Um(:,j)'.*ZV'*ZV/sumU(j); Δημιουργία του πίνακα συνδιακίμανσης (covariance matrix). Το γινόμενο (n1*Um(:,j)') είναι πίνακας ηηη στη συνέχεια κάθε στοιχείο του πολλαπλασιάζεται με το (ZV'*ZV/sumU(j) όπου το sumU(j) είναι ένα νούμερο . Το τελικό αποτέλεσμα του πολλαπλασιασμού είναι ηηη πίνακας.

d(:,j)=sum(ZV*(det(f)^(1/n)*inv(f)).*ZV,2); Υπολογισμός αποστάσεων με τη χρήση του f .

det(f) : η ορίζουσα του f για το τρέχον j (cluster)

$1/n$: όπου n το πλήθος των σετ δεδομένων

$\text{Inv}(f)$: ο αντίστροφος του f

Επομένως το $(\det(f)^{(1/n)} * \text{inv}(f))$ δίνει πίνακα $n \times n$ και επειδή ο ZV είναι $N \times n$ το γινόμενο τους δίνει πίνακα $N \times n$. Στη συνέχεια κάθε στοιχείο του πολλαπλασιάζεται με το ZV . Τέλος προσθέτονται τα στοιχεία κάθε στήλης ανά γραμμή και καταχωρούνται στον πίνακα αποστάσεων, στη στήλη που αντιπροσωπεύει το τρέχων j .

$d = (d).^(-1/(m-1))$; Αντιστροφή του πίνακα αποστάσεων

$U_0 = (d./(\text{sum}(d)' * c_1))$; Ενημέρωση του U_0

$U = U_0$; $U_m = U.^m$; $\text{sum}U = n_1 * \text{sum}(U_m)$; Τελικό U

for $j=1:c$,

$ZV = Z - N_1 * V(j,:)$;

$F(:,j) = n_1 * U_m(:,j)' * ZV' * ZV / \text{sum}U(1,j)$;

end; Τελικό F το οποίο είναι τριδιάστατος πίνακας ($n \times n \times c$)

B 2 Αλγόριθμος BIG BANG- BIG CRUNCH

```
%Αρχικός αριθμός σωματιδίων και επαναλήψεων
n_part=100;
n_it=100;
%Πρώτη φάση big bang. Δημιουργία ενός πίνακα 40 στοιχείων με τυχαίους
%αριθμούς από -10 έως 10
part_mat=-10+(10+10).*rand(n_part,1);
%Αρχικοποίηση των μητρώων των αθροισμάτων για τους υπολογισμούς του
κέντρου βάρους και των νέων σωματιδίων.
sum_arrey_num=zeros(30);
sum_arrey_denum=zeros(30);
%αρχικοποίηση του πίνακα κεντρών βαρών ανά επανάληψη. Το σημείο αυτό
είναι προαιρετικό καθώς σε κάθε επανάληψη αλλάζει, αλλά το κρατάμε
έτσι και αλλιώς για απεικόνιση.
cent_mas=zeros(1);
%Εκκίνηση του αλγορίθμου .Εξωτερικό loop για κάθε αριθμό επαναλήψεων.
for iteration_num=1:n_it
    %Τρέξιμο του αλγορίθμου για κάθε ένα σωματίδιο ξεχωριστά για
    υπολογισμό του κέντρου βάρους.
    for cur_part=1:n_part
        %Υπολογισμός του αριθμητή και του παρανομαστή του κλάσματος.
        sum_arrey_num(cur_part)=part_mat(cur_part)*1/((part_mat(cur_part)-
        4)^2);
        sum_arrey_denum(cur_part)=1/((part_mat(cur_part)-4)^2);
    end
    cent_mas=sum(sum_arrey_num)/sum(sum_arrey_denum);
    % Εκκίνηση του big crunch δημιουργώντας νέα σωματίδια κοντά στο
    κέντρο βάρους.
    for cur_part=1:n_part
        part_mat(cur_part)=cent_mas+(max(part_mat)*random('normal',0,0.5,1,1)
        /cur_part);
    end
end
```

```
end
end
```

B 3 Δημιουργία συστήματος ασαφούς λογικής από ομαδοποιημένα δεδομένα

Όπως αναφέρει και ο τίτλος ο αλγόριθμος δέχεται ως είσοδο τον πίνακα που περιέχει τα κέντρα των clusters και από αυτά δημιουργεί ένα σύστημα ασαφούς λογικής με τη χρήση του fuzzy toolbox.

Απαραίτητοι είσοδοι για τον αλγόριθμο είναι οι μεταβλητές m και pmv_data , όπου

m είναι ο πίνακας με τα κέντρα των κανόνων και pmv_data το σεντ δεδομένων στο οποίο έχουμε κάνει ομαδοποίηση.

num_var : Ο αριθμός των παραμέτρων ανά μέτρηση και num_mf ο αριθμός των συναρτήσεων συμμετοχής.

```
a=newfis('pmv');
num_var=size(m);
num_var=num_var(1,2)-1;
for i=1:num_var
    a=addvar(a,'input',i,[min(pmv_data(:,i)) max(pmv_data(:,i))]);
end
%-----
%-----
num_mf=size(m);
num_mf=num_mf(1,1);
for j=1:num_var
    for i=1:num_mf
        sigma=(max(pmv_data(:,j))-min(pmv_data(:,j)))/(num_mf);
        a=addmf(a,'input',j,int2str(i),'gaussmf',[sigma m(i,j)]);
    end
end
end
%-----
k=num_var+1;
a=addvar(a,'output','out',[min(pmv_data(:,k)) max(pmv_data(:,k))]);
sigma2=(max(m(:,k))-min(m(:,k)))/(num_mf);
for i=1:num_mf
    a=addmf(a,'output',1,int2str(i),'gaussmf',[sigma2 m(i,k)]);
end
end
%-----
rule=zeros(num_mf,k);
for i=1:num_mf
    for j=1:k
        rule(i,j)=i;
    end
end
end
```

```

r1=ones(num_mf,2);
rulelist=[rule r1];

a=addrule(a,rulelist);

```

B4 Αλγόριθμος τυχαίας κατανομής σε δεδομένα εκπαίδευσης και ελέγχου.

Ο αλγόριθμος λειτουργεί στα δεδομένα του pmv.

```

led=0;
start=1;
rnd_point=zeros(120,1);
for i=1:120
    while led==1 || start==1
        start=0;
        led=0;
        pointer=randi(171,1);
        for j=1:120
            if pointer==rnd_point(j)
                led=1;
            end
        end
        start=1;
        rnd_point(i)=pointer;
    end

start=1;
led=0;
rnd_check=zeros(51,1);
for i=1:30
    while led==1 || start==1
        start=0;
        led=0;
        pointer=randi(171,1);
        for j=1:51;
            if pointer==rnd_check(j);
                led=1;
            end
        end
        for j=1:120;
            if pointer==rnd_point(j);
                led=1;
            end
        end
        start=1;
        rnd_check(i)=pointer;
    end
end
x_train=zeros(120,3);
x_check=zeros(51,3);
for i=1:120
    x_train(i,:)=pmv_data(rnd_point(i,:),:);

```

```

end
for i=1:51
    x_check(i,:)=pmv_data(rnd_point(i,:),:);
end
train_input=x_train(:,1:2);
train_output=x_train(:,3);
check_input=x_check(:,1:2);
check_output=x_check(:,3);

```

B5 Αλγόριθμος δημιουργίας συστήματος ασαφούς λογικής με τη χρήση της μεθόδου Gustafson Kessel και στη συνέχεια βελτιστοποίηση με τη μέθοδο Big Bang Big Crunch για τα δεδομένα του pmv.

```

load pmv_data;
train_input=pmv_data(:,1:2);
train_output=pmv_data(:,3);
Z=pmv_data;
nx=2;
ny=5;
%-----
tol=0.001;
m=2;
c=20;
[N,n]=size(Z);
N1=ones(N,1); n1=ones(n,1); c1=ones(1,c);
U=zeros(N,c);
d=U;
F=zeros(n,n,c);

minZ=c1'*min(Z); maxZ=c1'*max(Z);
V=minZ+(maxZ-minZ).*rand(c,n);
for j=1:c,
    ZV=Z-N1*V(j,:);
    d(:,j)=sum((ZV.^2)')';
end;
d=(d+1e-100).^(-1/(m-1));
U0=(d./(sum(d')'*c1));

while max(max(U0-U))>tol
    U=U0; Um=U.^m; sumU=sum(Um);
    V=(Um'*Z)./(n1*sumU)';
    for j=1:c,
        ZV=Z-N1*V(j,:);
        f=n1*Um(:,j)'.*ZV'*ZV/sumU(j);
        d(:,j)=sum(ZV*(det(f)^(1/n)*inv(f)).*ZV,2);
    end;
    d=(d).^(-1/(m-1));
    U0=(d./(sum(d')'*c1));
end

U=U0; Um=U.^m; sumU=n1*sum(Um);
for j=1:c,
    ZV=Z-N1*V(j,:);
    F(:,j)=n1*Um(:,j)'.*ZV'*ZV/sumU(1,j);
end;

```

```

m=V;
%-----
a=newfis('pmv');
num_var=size(m);
num_var=num_var(1,2)-1;
    for i=1:num_var
        a=addvar(a,'input',i,[min(pmv_data(:,i))
max(pmv_data(:,i))]);
    end
%-----
%-----
num_mf=size(m);
num_mf=num_mf(1,1);
for j=1:num_var
    for i=1:num_mf
        sigma=(max(pmv_data(:,j))-min(pmv_data(:,j)))/(num_mf);
        a=addmf(a,'input',j,int2str(i),'gaussmf',[sigma m(i,j)]);

    end
end
%-----
k=num_var+1;
a=addvar(a,'output','out',[min(pmv_data(:,k)) max(pmv_data(:,k))]);
sigma2=(max(m(:,k))-min(m(:,k)))/(num_mf);
for i=1:num_mf
    a=addmf(a,'output',1,int2str(i),'gaussmf',[sigma2 m(i,k)]);
end
%-----
rule=zeros(num_mf,k);
for i=1:num_mf
    for j=1:k
        rule(i,j)=i;
    end
end
end
r1=ones(num_mf,2);
rulelist=[rule r1];

a=addrule(a,rulelist);

fuzout=evalfis(train_input',a);

trnRMSE1=norm(fuzout-train_output)/sqrt(length(fuzout))
%-----
%-----
% ekpaideysh me bbbc
%orismos metablitwn
num_mf=size(a.rule,2);
num_input=size(a.input,2);
for i=1:num_input
    sigma(i)= a.input(1,i).mf(1,1).params(1,1);
end
n_part=30;
n_it=10;
for i=1:num_input
    part_mat(:,i)=[sigma(i)-sigma(i)*0.1]+([sigma(i)+sigma(i)*0.1]-
[sigma(i)-sigma(i)*0.1]).*rand(n_part,1);
end
sum_arrey_num=zeros(n_part);
sum_arrey_denum=zeros(n_part);
for cur_input=1:num_input

```



```

for iteration_num=1:n_it
    for cur_part=1:n_part
        %setarisma tou sigma
        for cur_mf=1:num_mf

a.input(cur_input).mf(cur_mf).params(1)=part_mat(cur_part,cur_input);
            end
            %-----
            %ypologismos tou kritiriou
            new_val=evalfis(train_input,a);
            criterion=norm(new_val-
train_output)/sqrt(length(new_val));
            %-----

sum_arrey_num(cur_part)=part_mat(cur_part,cur_input)*1/criterion;
            sum_arrey_denum(cur_part)=1/criterion;
            end
            cent_mas=sum(sum_arrey_num)/sum(sum_arrey_denum);
            cent_hist(iteration_num,cur_input)=cent_mas;
            for cur_part=1:n_part

part_mat(cur_part,cur_input)=cent_mas+(max(part_mat(:,cur_input))*ran
dom('normal',0,0.5,1,1)/iteration_num);
                end
            end
end
fuzout2=evalfis(train_input',a);
trnRMSE2=norm(fuzout2-train_output)/sqrt(length(fuzout2))

scatter(pmv_data(:,1),pmv_data(:,3),'red')
hold on
scatter(pmv_data(:,1),fuzout2,'green')

```

B11 Αλγόριθμος για την μοντελοποίηση των δεδομένων ρυζιού με τη συνάρτηση genfis2 κάνοντας τυχαία κατανομή στα δεδομένα εκπαίδευσης και ελέγχου.

```
load rizi_data
led=0;
start=1;
rnd_point=zeros(75,1);
for i=1:75
    while led==1 || start==1
        start=0;
        led=0;
        pointer=randi(105,1);
        for j=1:75
            if pointer==rnd_point(j)
                led=1;
            end
        end
    end
    start=1;
    rnd_point(i)=pointer;
end
start=1;
led=0;
rnd_check=zeros(30,1);
for i=1:30
    while led==1 || start==1
        start=0;
        led=0;
        pointer=randi(105,1);
        for j=1:30;
            if pointer==rnd_check(j);
                led=1;
            end
        end
        for j=1:75;
            if pointer==rnd_point(j);
                led=1;
            end
        end
    end
    start=1;
    rnd_check(i)=pointer;
end
x_train=zeros(75,6);
x_check=zeros(30,6);
for i=1:75
    x_train(i,:)=rizi_data(rnd_point(i,:),:);
end
for i=1:30
    x_check(i,:)=rizi_data(rnd_point(i,:),:);
end

train_input=x_train(:,1:5);
train_output=x_train(:,6);
check_input=x_check(:,1:5);
check_output=x_check(:,6);
```

```
fismat=genfis2(train_input,train_output,0.4);
fuzout=evalfis(train_input,fismat);
trnRMSE=norm(fuzout-train_output)/sqrt(length(fuzout))

scatter(train_input(:,1),train_output,'red')
hold on
scatter(train_input(:,1),fuzout)
figure
fuzout2=evalfis(check_input,fismat);
chkRMSE=norm(fuzout2-check_output)/sqrt(length(fuzout))
scatter(check_input(:,1),check_output,'red')
hold on
scatter(check_input(:,1),fuzout2)
```

BIBΛΙΟΓΡΑΦΙΑ

ΚΕΦΑΛΑΙΟ 1

[1] Robert Babuska, **Fuzzy and Neural Control**, Disc Course Lecture Notes (November 2009)

[2] Agus Priyono, Muhammad Ridwan, Ahmad Jais Alias Riza Atiq O. K. Rahmat, Azmi Hassan, & Mohd. Alauddin Mohd. Ali **Generation of fuzzy rules with subtractive clustering** *Jurnal Teknologi*, 43(D) Dis. 2005: 143-153

[3] <http://www.ai-junkie.com>

ΚΕΦΑΛΑΙΟ 2

[4] Αναστάσιος Ντούνης **ΕΥΦΥΗΣ ΕΛΕΓΧΟΣ**

[5] Σ. Τζαφέστας, **Υπολογιστική Νοημοσύνη, Τόμος Α**, 2002

ΚΕΦΑΛΑΙΟ 3

[6] Chenxu Duan, Xudong Ding, Fenggang Shi, Xianhua Xiao, Peiyong. **PMV-based Fuzzy Algorithms for Controlling Indoor Temperature**, 2011 IEEE

[7] Ken Nozaki, Hisao Ishibuchi, Hideo Tanaka, **A simple but powerful heuristic method for generating fuzzy rules from numerical data**, *Fuzzy Sets and Systems* 86(1997)

ΚΕΦΑΛΑΙΟ 4

[8] Osman K. Erol, Ibrahim Eksin, **A new optimization method: Big Bang–Big Crunch**, *Advances in Engineering Software* (2006)