ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

Ομαδοποίηση ‘‘εν πτήσει’’ δοσοληψιών σε ομάδες με παρόμοια χαρακτηριστικά φόρτου εργασίας

Χρήστος Κωνσταντίνου Κλουκίνας

Μεταπτυχιακή Εργασία

Ηράκλειο, Απρίλιος 1997

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

Ομαδοποίηση ‘‘εν πτήσει’’ δοσοληψιών σε ομάδες με παρόμοια χαρακτηριστικά φόρτου εργασίας

Εργασία που υποβλήθηκε από τον
Χρήστο Κωνσταντίνου Κλουκίνα
ως μερική εκπλήρωση των απαιτήσεων
για την απόκτηση
ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΔΙΠΛΩΜΑΤΟΣ ΕΙΔΙΚΕΥΣΗΣ

Συγγραφέας:

Χρήστος Κωνσταντίνου Κλουκίνας
Τμήμα Επιστήμης Υπολογιστών

Εισηγητική Επιτροπή:

Χρήστος Νικολάου, Αναπληρωτής Καθηγητής, Επόπτης

Γιώργος Τζιρίτας, Αναπληρωτής Καθηγητής, Μέλος

Γιώργος Γεωργακόπουλος, Επίκουρος Καθηγητής, Μέλος

Γιώργος Ποταμιάς, Ερευνητής, Μέλος

Δεκτή:

Πάνος Κωσταντόπουλος, Αναπληρωτής Καθηγητής
Πρόεδρος Επιτροπής Μεταπτυχιακών Σπουδών

Ηράκλειο, Απρίλιος 1997

Στους γονείς μου,
Νέλλη και Κωνσταντίνο

Περίληψη

Η παρούσα εργασία ασχολείται με την **ομαδοποίηση** (*clustering*) **δοσοληψιών** (*transactions*), που εμφανίζονται σε ένα **κατανεμημένο** (*distributed*) **σύστημα επεξεργασίας δοσοληψιών** (*OLTP system*), συμφώνως προς τα χαρακτηριστικά του **φόρτου εργασίας** (*workload*) αυτών.

Αποτελεί συνέχεια της μεταπτυχιακής εργασίας [Lab95], διαφοροποιείται, δε, από αυτήν στο ότι η ομαδοποίηση γίνεται πάνω σε μία δυναμικώς μεταβαλλόμενη ακολουθία δεδομένων και όχι πλέον σε ένα στατικό σύνολο αυτών.

Για το σκοπό αυτό, έχουν υλοποιηθεί δύο δυναμικοί αλγόριθμοι ομαδοποίησης, εκ των οποίων ο πρώτος βασίζεται στη χρήση ενός νευρωνικού δικτύου τύπου K-MEANS, ενώ ο δεύτερος κάνει χρήση γραφοθεωρητικών μεθόδων ομαδοποίησης.

Abstract

The following work deals with clustering of transactions in a distributed OLTP system, according to their workload characteristics.

It continues the work done in [Lab95], differentiating from it by the fact that it no longer deals with static sets of data and thus with batch clustering algorithms. Instead, it deals with a dynamic sequence of data which calls upon the use of an ‘‘on the fly’’ clustering algorithm.

For this purpose, two dynamic clustering algorithms have been implemented. The first one is based on an artificial neural network of the adaptive K-MEANS family of networks, while the second one is based on a graph - theoretical clustering method.

Ευχαριστίες

Θα ήθελα να εκφράσω τις ευχαριστίες μου σε ορισμένους ανθρώπους, η βοήθεια των οποίων υπήρξε καθοριστική για την περάτωση της παρούσας εργασίας.

Πρώτα από όλους, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου, κ. Χρήστο Νικολάου, ο οποίος μου έδωσε τη δυνατότητα να ασχοληθώ με το θέμα αυτό και με καθοδήγησε καθόλη τη διάρκεια των μεταπτυχιακών σπουδών μου.

Επίσης, θα ήθελα να ευχαριστήσω τα μέλη της εξεταστικής επιτροπής της μεταπτυχιακής μου εργασίας, καθηγητές κ. Γιώργο Τζιρίτα, κ. Γιώργο Γεωργακόπουλο και κ. Γιώργο Ποταμιά, για τις συμβουλές τους και τις πολύ σημαντικές παρατηρήσεις τους πάνω σε διάφορα θέματα της εργασίας αυτής, καθώς και τον Αναπληρωτή Καθηγητή του Φυσικού Τμήματος κ. Νίκο Κυλάφη για τις συμβουλές του και τη βοήθειά του.

Θα ήθελα επίσης να ευχαριστήσω τους φίλους μου Ιωάννη Πάτρα, Απόστολο Ζάρρα, Κωνσταντίνο Χάρη, Γιώργο Καγκάδη, Δημήτρη Παπαδάκη, Μανώλη Μαραζάκη, Γιώργο Γεωργιαννάκη και Ειρήνη Φουντουλάκη για τη βοήθεια και συμπαράστασή τους.

Η ολοκλήρωση της παρούσας εργασίας στάθηκε δυνατή χάρη στην υλικοτεχνική και οικονομική υποστήριξη που μου παρείχαν κατά τη διάρκεια των σπουδών μου, τόσο το Τμήμα Επιστήμης Υπολογιστών του Πανεπιστημίου Κρήτης, όσο και το Ινστιτούτο Πληροφορικής του Ιδρύματος Τεχνολογίας και Έρευνας και ειδικότερα, η ερευνητική ομάδα των Πλειάδων.

Τέλος, θα ήθελα να ευχαριστήσω τους γονείς μου, Νέλλη και Κωνσταντίνο, καθώς και την αδελφή μου, Μαριάννα, για τη συμπαράστασή τους και την αμέριστη υποστήριξη που μου έδωσαν καθόλη τη διάρκεια των σπουδών μου.

Περιεχόμενα

Περίληψη iii

Abstract v

Ευχαριστίες vii

Περιεχόμενα ix

Κατάλογος Πινάκων xi

Κατάλογος Σχημάτων xiii

1 Εισαγωγή 1

2 Στοιχεία Τεχνητών Νευρωνικών Δικτύων 5
2.1 Ο Νευρώνας . . . 6
2.2 Μαθηματικό Μοντέλο του Νευρώνα . . . 7
2.2.1 Συναρτήσεις Ενεργοποίησης . . . 9
2.3 Αρχιτεκτονικές Νευρωνικών Δικτύων . . . 13
2.4 Η Διαδικασία Εκμάθησης . . . 14
2.5 Ομαδοποίηση με τα Νευρωνικά Δίκτυα . . . 18
2.5.1 Ο Αυτο-Οργανωνόμενος Χάρτης Χαρακτηριστικών . . . 18

3 Δυναμικώς Αναπροσαρμοζόμενος K-MEANS 23
3.1 Ο Κλασσικός K-MEANS . . . 23
3.1.1 Παραλλαγές του Κλασσικού K-MEANS . . . 25
3.2 Υλοποίηση του K-MEANS με χρήση Νευρωνικών Δικτύων . . . 26
3.3 Προβλήματα του Αναπροσαρμοζόμενου K-MEANS . . . 27
3.4 Βέλτιστος Αναπροσαρμοζόμενος K-MEANS . . . 28
3.4.1 Βέλτιστη χρήση των νευρώνων του δικτύου . . . 29
3.4.2 Δυναμική Ρύθμιση του Ρυθμού Εκμάθησης . . . 30
3.5 Ο αναπροσαρμοζόμενος K-MEANS στο CLUE . . . 30

4 Γραφοθεωρητική Μέθοδος Ομαδοποίησης ‘‘Εν Πτήσει’’ 33
4.1 Γενικά Στοιχεία Γραφοθεωρητικών Μεθόδων Ομαδοποίησης . . . 33
4.2 Αρχική Γραφοθεωρητική Μέθοδος Ομαδικής Επεξεργασίας . . . 34
4.3 Μετατροπή της Μεθόδου Ομαδικής Επεξεργασίας σε Μέθοδο Ομαδοποίησης ‘‘Εν Πτήσει’’ . . . 36

5 Αποτελέσματα Αρχικών Πειραμάτων **41**
5.1 Σχεδιασμός Αρχικών Πειραμάτων 41
5.2 Αποτελέσματα του Αναπροσαρμοζόμενου K-MEANS (VWMSE) 43
5.3 Πειράματα με συνάρτηση κόστους τη MSE 49
5.4 Αποτελέσματα με τη Γραφοθεωρητική Μέθοδο Ομαδοποίησης Ομαδικής Επεξεργασίας 54

6 Αποτελέσματα Τελικών Πειραμάτων **57**
6.1 Σενάριο πειραμάτων 57
6.2 Συνθετικά δεδομένα 58
6.3 Πραγματικά δεδομένα 61

7 Συμπεράσματα - Μελλοντικές Κατευθύνσεις **65**
7.1 Χρήση των Αλγόριθμων σε Πραγματικά Συστήματα 65
7.1.1 Παραλληλοποίηση των Αλγόριθμων 65
7.2 Εύρεση Βέλτιστου Αλγόριθμου 66
7.3 Μελλοντικές Κατευθύνσεις 67

Α Ισοδυναμία των Συναρτήσεων VWMSE και MSE **69**

Β Αλγόριθμοι Κατασκευής του Ελάχιστου Ζευγνύοντος Δένδρου **71**

Γ Interacting with CLUE **73**
Γ.1 CLUE command line options 73
Γ.2 CLUE Reference Input File 74
Γ.3 SCRIPT Reference Input File 78

Δ Αποτελέσματα Αρχικών Πειραμάτων με τον Αναπροσαρμοζόμενο K-MEANS **83**
Δ.1 Πειράματα με συνάρτηση κόστους τη VWMSE 83
Δ.2 Πειράματα με συνάρτηση κόστους τη MSE 89

Ε Αποτελέσματα Αρχικών Πειραμάτων με τη Γραφοθεωρητική Μέθοδο **93**

ΣΤ Τελικά Πειράματα **97**
ΣΤ.1 Διαχωρισμός των συνόλων δεδομένων 97
ΣΤ.2 Αρχεία εντολών του CLUE που χρησιμοποιήθησαν 99
ΣΤ.2.1 script-1000-nnmse 100
ΣΤ.2.2 script-1000-nnvwmse 100
ΣΤ.2.3 script-1000-graphos 101
ΣΤ.2.4 script-1000-kmeans 101
ΣΤ.2.5 script-1000-halc 102
ΣΤ.3 Πλήθη ομάδων που κατασκευάσθησαν 103

Βιβλιογραφία **106**

Κατάλογος Πινάκων

2.1 Ο αλγόριθμος SOFM 21

3.1 Γενική περιγραφή του κλασσικού αλγόριθμου K-MEANS. 24
3.2 Υλοποίηση του K-MEANS με ένα τεχνητό νευρωνικό δίκτυο. 27

4.1 Γραφοθεωρητικός - ομαδικής επεξεργασίας - αλγόριθμος διάσπασης ενός ελαχίστου ζευγνύοντος δένδρου σε υποδένδρα. 35
4.2 Γραφοθεωρητικός αλγόριθμος ομαδοποίησης δεδομένων ‘‘εν πτήσει’’ (Μέρος πρώτο) 38
4.3 Γραφοθεωρητικός αλγόριθμος ομαδοποίησης δεδομένων ‘‘εν πτήσει’’ (Μέρος δεύτερο) 39

Δ.1 NN-VWMSE: Μη επικαλυπτόμενα τετράγωνα δίχως θόρυβο (Αρχικοποίηση τυχαία) 85
Δ.2 NN-VWMSE: Μη επικαλυπτόμενα τετράγωνα δίχως θόρυβο 85
Δ.3 NN-VWMSE: Μη επικαλυπτόμενα τετράγωνα με θόρυβο (Αρχικοποίηση τυχαία) 86
Δ.4 NN-VWMSE: Μη επικαλυπτόμενα τετράγωνα με θόρυβο 86
Δ.5 NN-VWMSE: Μερικώς επικαλυπτόμενα τετράγωνα δίχως θόρυβο (Αρχικοποίηση τυχαία) 87
Δ.6 NN-VWMSE: Μερικώς επικαλυπτόμενα τετράγωνα δίχως θόρυβο 87
Δ.7 NN-VWMSE: Μερικώς επικαλυπτόμενα τετράγωνα με θόρυβο (Αρχικοποίηση τυχαία) 88
Δ.8 NN-VWMSE: Μερικώς επικαλυπτόμενα τετράγωνα με θόρυβο 88
Δ.9 NN-MSE: Μή επικαλυπτόμενα τετράγωνα δίχως θόρυβο (Αρχικοποίηση τυχαία) 89
Δ.10 NN-MSE: Μή επικαλυπτόμενα τετράγωνα δίχως θόρυβο 89
Δ.11 NN-MSE: Μή επικαλυπτόμενα τετράγωνα με θόρυβο (Αρχικοποίηση τυχαία) . 90
Δ.12 NN-MSE: Μή επικαλυπτόμενα τετράγωνα με θόρυβο 90
Δ.13 NN-MSE: Μερικώς επικαλυπτόμενα τετράγωνα δίχως θόρυβο (Αρχικοποίηση τυχαία) 91
Δ.14 NN-MSE: Μερικώς επικαλυπτόμενα τετράγωνα δίχως θόρυβο 91
Δ.15 NN-MSE: Μερικώς επικαλυπτόμενα τετράγωνα με θόρυβο (Αρχικοποίηση τυχαία) 92
Δ.16 NN-MSE: Μερικώς επικαλυπτόμενα τετράγωνα με θόρυβο 92

Ε.1 Γραφοθεωρητική Μέθοδος: Μη επικαλυπτόμενα τετράγωνα, δίχως θόρυβο . . 94
Ε.2 Γραφοθεωρητική Μέθοδος: Μη επικαλυπτόμενα τετράγωνα, με θόρυβο 94
Ε.3 Γραφοθεωρητική Μέθοδος: Μερικώς επικαλυπτόμενα τετράγωνα, δίχως θόρυβο 95

Ε.4 Γραφοθεωρητική Μέθοδος: Μερικώς επικαλυπτόμενα τετράγωνα, με θόρυβο . 95

ΣΤ.1 Πλήθος ομάδων για το σύνολο συνθετικών δεδομένων 1000x1000=100:diagsR 104
ΣΤ.2 Πλήθος ομάδων για το σύνολο συνθετικών δεδομένων 10000x1000=100:diagsR 104
ΣΤ.3 Πλήθος ομάδων για το σύνολο συνθετικών δεδομένων 30000x1000=100:diagsR 105
ΣΤ.4 Πλήθος ομάδων για το σύνολο πραγματικών δεδομένων PULS 105
ΣΤ.5 Πλήθος ομάδων για το σύνολο πραγματικών δεδομένων DOA 105

Κατάλογος Σχημάτων

2.1 Χονδρική αναπαράσταση ενός νευρώνα 6
2.2 Μαθηματικό μοντέλο ενός νευρώνα 8
2.3 Δύο παραλλαγές του μοντέλου του νευρώνα k 10
2.4 Συνάρτηση κατωφλίου 11
2.5 Τμηματικώς-Γραμμική συνάρτηση 11
2.6 Τμηματικώς-Γραμμική συνάρτηση 12
2.7 Δίκτυα εμπρόσθιας μετάδοσης 13
2.8 Αναδραστικό δίκτυο 14
2.9 Δίκτυα με δομή πίνακα 15
2.10 Ο αυτο-οργανωνόμενος χάρτης χαρακτηριστικών 20
2.11 Τα συναπτικά βάρη του j-οστού νευρώνα 20
2.12 Γειτονιές στον SOFM κατά την πάροδο του χρόνου 20

4.1 Λειτουργία της γραφοθεωρητικής μεθόδου: κατασκευή πλήρους γράφου και του ελαχίστου ζευγνύοντος δένδρου αυτού από ένα σύνολο δεδομένων 34
4.2 Προσθήκη ενός νέου δεδομένου ν και μετατροπή του ελαχίστου ζευγνύοντος δένδρου 36

5.1 Γραφική παράσταση των διαφορετικών τύπων δεδομένων 42
5.2 NN-VWMSE: Μη επικαλυπτόμενα τετράγωνα δίχως θόρυβο (Αρχικοποίηση τυχαία) 45
5.3 NN-VWMSE: Μη επικαλυπτόμενα τετράγωνα δίχως θόρυβο 45
5.4 NN-VWMSE: Μη επικαλυπτόμενα τετράγωνα με θόρυβο (Αρχικοποίηση τυχαία) 46
5.5 NN-VWMSE: Μη επικαλυπτόμενα τετράγωνα με θόρυβο 46
5.6 NN-VWMSE: Μερικώς επικαλυπτόμενα τετράγωνα δίχως θόρυβο (Αρχικοποίηση τυχαία) 47
5.7 NN-VWMSE: Μερικώς επικαλυπτόμενα τετράγωνα δίχως θόρυβο 47
5.8 NN-VWMSE: Μερικώς επικαλυπτόμενα τετράγωνα με θόρυβο (Αρχικοποίηση τυχαία) 48
5.9 NN-VWMSE: Μερικώς επικαλυπτόμενα τετράγωνα με θόρυβο 48
5.10 NN-MSE: Μη επικαλυπτόμενα τετράγωνα δίχως θόρυβο (Αρχικοποίηση τυχαία) 50
5.11 NN-MSE: Μη επικαλυπτόμενα τετράγωνα δίχως θόρυβο 50
5.12 NN-MSE: Μη επικαλυπτόμενα τετράγωνα με θόρυβο (Αρχικοποίηση τυχαία) . 51
5.13 NN-MSE: Μη επικαλυπτόμενα τετράγωνα με θόρυβο 51
5.14 NN-MSE: Μερικώς επικαλυπτόμενα τετράγωνα δίχως θόρυβο (Αρχικοποίηση τυχαία) 52
5.15 NN-MSE: Μερικώς επικαλυπτόμενα τετράγωνα δίχως θόρυβο 52
5.16 NN-MSE: Μερικώς επικαλυπτόμενα τετράγωνα με θόρυβο (Αρχικοποίηση τυχαία) 53

5.17 NN-MSE: Μερικώς επικαλυπτόμενα τετράγωνα με θόρυβο 53
5.18 Γραφοθεωρητική μέθοδος: Μη επικαλυπτόμενα τετράγωνα δίχως θόρυβο. . . . 55
5.19 Γραφοθεωρητική μέθοδος: Μη επικαλυπτόμενα τετράγωνα με θόρυβο. 55
5.20 Γραφοθεωρητική μέθοδος: Μερικώς επικαλυπτόμενα τετράγωνα δίχως θόρυβο. 56
5.21 Γραφοθεωρητική μέθοδος: Μερικώς επικαλυπτόμενα τετράγωνα με θόρυβο. . . 56

6.1 Διακύμανση του Τετραγωνικού Σφάλματος για το σύνολο συνθετικών δεδομένων 1000x1000=100:diagsR. 59
6.2 Διακύμανση του Τετραγωνικού Σφάλματος για το σύνολο συνθετικών δεδομένων 10000x1000=100:diagsR. 60
6.3 Διακύμανση του Τετραγωνικού Σφάλματος για το σύνολο συνθετικών δεδομένων 30000x1000=100:diagsR. 61
6.4 Διακύμανση του Τετραγωνικού Σφάλματος για το σύνολο πραγματικών δεδομένων PULS. 63
6.5 Διακύμανση του Τετραγωνικού Σφάλματος για το σύνολο πραγματικών δεδομένων DOA. 64

Κεφάλαιο 1

Εισαγωγή

Στις σύγχρονες εφαρμογές των βάσεων δεδομένων οι απαιτήσεις στην υπολογιστική ισχύ, το μέγεθος της κυρίως μνήμης, καθώς και το μέγεθος των εξωτερικών αποθηκευτικών μέσων (δίσκων, ταινιών) οδήγησαν στην κατασκευή κατανεμημένων συστημάτων, που μπορούν να ανταπεξέλθουν στις απαιτήσεις αυτές με πολύ μικρότερο κόστος λειτουργίας και μεγαλύτερες δυνατότητες άμεσης αναβάθμισης από οποιοδήποτε σύστημα που θα βασιζόταν στη χρήση ενός υπερυπολογιστή.

Ένας εξίσου σημαντικός παράγων που οδήγησε στην υιοθέτηση των κατανεμημένων συστημάτων είναι το γεγονός ότι πλέον χρησιμοποιούνται αυτά για πλείστες δραστηριότητες, π.χ. κράτηση θέσεων από ταξιδιωτικά γραφεία ή διαχείριση λογαριασμών στα διάφορα υποκαταστήματα τραπεζών, οι οποίες εκ της φύσεώς τους οδηγούν στην κατανεμημένη σχεδίαση και λειτουργία του τελικού συστήματος που θα κληθεί να τις υποστηρίξει. Είναι, παραδείγματος χάριν, φανερό ότι η κατάτμηση ενός συστήματος επεξεργασίας τραπεζικών δοσοληψιών οδηγεί στην κατά πολύ ταχύτερη επεξεργασία του μεγαλύτερου αριθμού των δοσοληψιών, που ούτως ή άλλως κάνουν χρήση πόρων που βρίσκονται τοπικώς, όπως η ανάληψη ή η κατάθεση ενός ποσού χρημάτων από/σε ένα λογαριασμό που έχει ανοιχθεί στο συγκεκριμένο υποκατάστημα, περιορίζοντας έτσι τον αριθμό των δοσοληψιών που πρέπει να μεταφερθούν στους κεντρικούς υπολογιστές της τράπεζας.

Για την καλύτερη επίδοση των κατανεμημένων συστημάτων αυτών, είναι απαραίτητη η εφαρμογή αλγόριθμων δρομολόγησης των δοσοληψιών και εξισορρόπησης του φόρτου αυτών στους κατά τόπους υπολογιστικούς κόμβους του συστήματος, προκειμένου να γίνει ταχύτερη η επεξεργασία των δοσοληψιών, κάνοντας χρήση πόρων τους συστήματος που κατά καιρούς υποχρησιμοποιούνται.

Οι αλγόριθμοι αυτοί, για την αποδοτική λειτουργία τους, χρειάζονται να έχει γίνει ένας χαρακτηρισμός του φόρτου εργασίας των δοσοληψιών ώστε να γνωρίζουν το είδος και το μέγεθος των απαιτήσεων αυτών σε πόρους του συστήματος. Όπως έχει παρατηρηθεί, οι απαιτήσεις αυτές, π.χ. η χρήση συγκεκριμένων αρχείων, της κεντρικής μονάδας επεξεργασίας, η παρουσία σημείων συγχρονισμού, κ.λ.π. είναι εν γένει ανεξάρτητες από το **ρυθμό άφιξης** (*arrival rate*) των δοσοληψιών.

Στην εργασία [Lab95] προτάθηκε ότι η γεωγραφική και οργανωτική δομή ενός οργανισμού επηρεάζει άμεσα τον τρόπο κατά τον οποίο αποθηκεύονται αλλά και προσπελαύνονται τα δεδομένα της υποκείμενης βάσης δεδομένων. Ως εκ τούτου, χρησιμοποιήθηκε η τριάδα (αύξων αριθμός προγράμματος, αύξων αριθμός χρήστη, αύξων αριθμός τερματικού) προκειμένου να αναγνωρίζονται οι δοσοληψίες, θεωρώντας ότι δύο δοσοληψίες που έχουν ίδια τα τρία αυτά χαρακτηριστικά θα παρουσιάζουν σε πολύ μεγάλο βαθμό ομοιότητα ως προς τις απαιτήσεις τους.

Σε ένα σύγχρονο οργανισμό όμως, ο αριθμός των διαφορετικών τριάδων που μπορούν

να έχουν οι δοσοληψίες είναι τεράστιος, ενώ μόνο ένας πολύ μικρός αριθμός αυτών μπορεί να χρησιμοποιηθεί από τους διάφορους αλγόριθμους δρομολόγησης και εξισορρόπησης φόρτου. Επομένως, πρέπει να ομαδοποιηθούν οι δοσοληψίες σε μεγαλύτερες ομάδες που παρουσιάζουν κοινά χαρακτηριστικά ως προς τις απαιτήσεις τους.

Στα πλαίσια του ερευνητικού προγράμματος LYDIA [ESP], αναπτύχθηκε από την ομάδα Παράλληλων και Κατανεμημένων Συστημάτων του Ινστιτούτου Πληροφορικής, του Ιδρύματος Τεχνολογίας και Έρευνας, το CLUE, ένα περιβάλλον για την ομαδοποίηση δοσοληψιών σε ομάδες με κοινά χαρακτηριστικά φόρτου εργασίας, που περιγράφεται στην μεταπτυχιακή εργασία του Αλέξανδρου Λαμπρινίδη [Lab95]. Στην εργασία αυτή, μελετήθηκε το πρόβλημα της ομαδοποίησης στατικών συνόλων δοσοληψιών, που καταγράφονται κατά τη λειτουργία ενός κατανεμημένου συστήματος επεξεργασίας δοσοληψιών. Υλοποιήθησαν, δε, τρεις αλγόριθμοι ομαδοποίησης, ο HALC, ο BOND ENERGY ALGORITHM [MSW72] και ο K-MEANS και μελετήθηκε η επίδοσή τους στο εν λόγω πρόβλημα.

Από τη χρήση των αλγόριθμων αυτών, παρατηρήθηκε ότι παρόλο που είναι δυνατόν να εξαχθούν σημαντικά συμπεράσματα για το πού θα πρέπει να επεξεργάζονται οι δοσοληψίες και για το πώς θα πρέπει να κατανεμηθεί η υποκείμενη βάση δεδομένων, εν τούτοις θα ήταν επιθυμητό να υπάρχει και ένα σύστημα το οποίο να μπορεί να κάνει την ομαδοποίηση αυτή ‘‘εν πτήσει’’, δηλαδή κατά τη διάρκεια της λειτουργίας του συστήματος επεξεργασίας δοσοληψιών, ώστε να μπορεί να αναγνωρίζει μεταβολές στον τρόπο χρήσης του όλου συστήματος και αναλόγως να βοηθά αυτό να λαμβάνει αποφάσεις για το ποιός θα ήταν ο βέλτιστος τρόπος λειτουργίας του ανά πάσα στιγμή.

Στα πλαίσια της παρούσας μεταπτυχιακής εργασίας μελετήθηκε το πρόβλημα αυτό της ομαδοποίησης ‘‘εν πτήσει’’ και ανεπτύχθησαν δύο βασικές μέθοδοι για την επίλυσή του. Η πρώτη εκ των δύο χρησιμοποιεί ένα νευρωνικό δίκτυο τύπου K-MEANS, που έχει τη δυνατότητα να αναπροσαρμόζει τις ομάδες στις οποίες έχει κατατάξει τις δοσοληψίες, δίνοντας ανά πάσα στιγμή μία κατά το δυνατόν πιστότερη αναπαράσταση του τρόπου χρήσης του συστήματος και να μεταβάλλει το ρυθμό της αναπροσαρμογής αυτής, αναλόγως προς το πόσο μεγάλες είναι οι μεταβολές στη χρήση του συστήματος.

Η δεύτερη βασίζεται σε μία γραφοθεωρητική μέθοδο ομαδοποίησης τύπου **ομαδικής επεξεργασίας** (*batch processing*), η οποία μεταβλήθηκε καταλλήλως στην εργασία αυτή, προκειμένου να μπορεί να επιτελέσει ομαδοποίηση ‘‘εν πτήσει’’. Στις μεταβολές και προσθήκες που έγιναν στην αρχική μέθοδο, έγινε προσπάθεια να βρεθούν τρόποι αναπροσαρμογής των ομάδων των δοσοληψιών με το μικρότερο δυνατόν κόστος, κάνοντας *τοπικές αλλαγές μόνο* στην εκάστοτε αναπαράσταση της εικόνας του συστήματος. Έτσι, η τελική μέθοδος διαφέρει από την αρχική, την οποία και χρησιμοποιεί πλέον μόνον κατά την αρχικοποίησή της.

Για τη διαπίστωση των δυνατοτήτων των προαναφερθέντων μεθόδων, καθώς και των περιορισμών τους, έγιναν μία σειρά από πειράματα, τόσο με τεχνητώς κατασκευασμένα δεδομένα, όσο και με δύο πραγματικά **ίχνη** (*traces*) των δοσοληψιών που εμφανίστηκαν κατά τη χρήση δύο συστημάτων επεξεργασίας δοσοληψιών, τα οποία παρασχέθησαν από την Siemens Nixdorf Informationssysteme AG [SNI].

Η διάρθρωση της εργασίας έχει ως εξής: στο κεφάλαιο 2 παρατίθενται ορισμένα βασικά στοιχεία των τεχνητών νευρωνικών δικτύων, προκειμένου να αποκτήσει ο αναγνώστης μία ιδέα του χώρου αυτού, που θα του επιτρέψει να παρακολουθήσει πιο εύκολα την ανάπτυξη του δυναμικώς αναπροσαρμοζόμενου K-MEANS αλγόριθμου που παρατίθεται στο κεφάλαιο 3. Στο κεφάλαιο 4 παρουσιάζεται η γραφοθεωρητική μέθοδος ομαδοποίησης ‘‘εν πτήσει’’ και στο κεφάλαιο 5 παρουσιάζονται τα αποτελέσματα των αρχικών πειραμάτων που έγιναν κατά την ανάπτυξη των μεθόδων ομαδοποίησης ‘‘εν πτήσει’’. Τέλος, στο κεφάλαιο 6 παρουσιάζονται τα αποτελέσματα των τελικών πειραμάτων που έγιναν με το CLUE.

Οδηγίες για τη χρήση του CLUE παρατίθενται στο παράρτημα Γ, ενώ στο παράρτημα ΣΤ δίνεται μία πλήρης περιγραφή των τελικών πειραμάτων που έγιναν, ώστε να είναι δυνατή η επανάληψή τους.

Κεφάλαιο 2

Στοιχεία Τεχνητών Νευρωνικών Δικτύων

Τα τελευταία χρόνια έχει συγκεντρωθεί μεγάλο ερευνητικό ενδιαφέρον στον τομέα των τεχνητών νευρωνικών δικτύων. Το ενδιαφέρον αυτό πηγάζει μερικώς από την επιθυμία να κατασκευασθούν υπολογιστικά συστήματα με ικανότητες παρόμοιες με αυτές του ανθρώπινου εγκεφάλου.

Ο ανθρώπινος εγκέφαλος λειτουργεί εντελώς διαφορετικά από ένα συμβατικό ηλεκτρονικό υπολογιστή. Δύναται όμως να επιτελέσει λειτουργίες, όπως η αναγνώριση φωνής ή η όραση, πολύ πιο εύκολα και πιο αποδοτικά από οποιονδήποτε σύγχρονο υπολογιστή. Αυτό είναι άμεση συνέπεια του διαφορετικού τρόπου λειτουργίας του, καθώς, αντιθέτως με τους υπολογιστές που στη γενική περίπτωση στηρίζονται στο σειριακό μοντέλο λειτουργίας που είχε ορίσει ο πρωτεργάτης της πληροφορικής John Von Neumann, ο εγκέφαλος στηρίζεται στην παράλληλη λειτουργία και αλληλεπίδραση ενός τεράστιου αριθμού από απλές υπολογιστικές μονάδες, τους νευρώνες. Δηλαδή, έχει τη δυνατότητα να διασπά ένα μεγάλο και δύσκολο πρόβλημα, όπως η αναγνώριση ενός προσώπου σε ένα άγνωστο περιβάλλον, σε πολύ μικρότερα κομμάτια τα οποία τα επεξεργάζονται ταυτοχρόνως μυριάδες νευρώνων.

Η προσπάθεια που έχει γίνει μέχρι στιγμής, έχει οδηγήσει σε ένα αρκετά καλό επίπεδο γνώσης του ακριβούς τρόπου λειτουργίας του εγκεφάλου καθώς και τη δημιουργία μίας πληθώρας μοντέλων, τόσο των νευρώνων όσο και των αλγόριθμων που χρησιμοποιούν αυτοί για την επεξεργασία της πληροφορίας. Έχουν, δε, προταθεί και διάφοροι αλγόριθμοι εκμάθησης και επεξεργασίας της πληροφορίας που παρόλο που δεν βρίσκουν απευθείας φυσικό ανάλογο, εν τούτοις βασίζονται στις ίδιες βασικές αρχές του μεγάλου βαθμού παραλληλισμού και της χρήσης πολύ απλών βασικών υπολογιστικών μονάδων.

Στη συνέχεια του κεφαλαίου αυτού πρόκειται να δοθεί μία περιληπτική περιγραφή του νευρώνα, του κυρίαρχου μαθηματικού μοντέλου αυτού και των βασικών αρχιτεκτονικών και αλγόριθμων εκμάθησης που χρησιμοποιούνται στα τεχνητά νευρωνικά δίκτυα, καθώς και μία σύντομη περιγραφή του αυτο-οργανωνόμενου χάρτη χαρακτηριστικών.

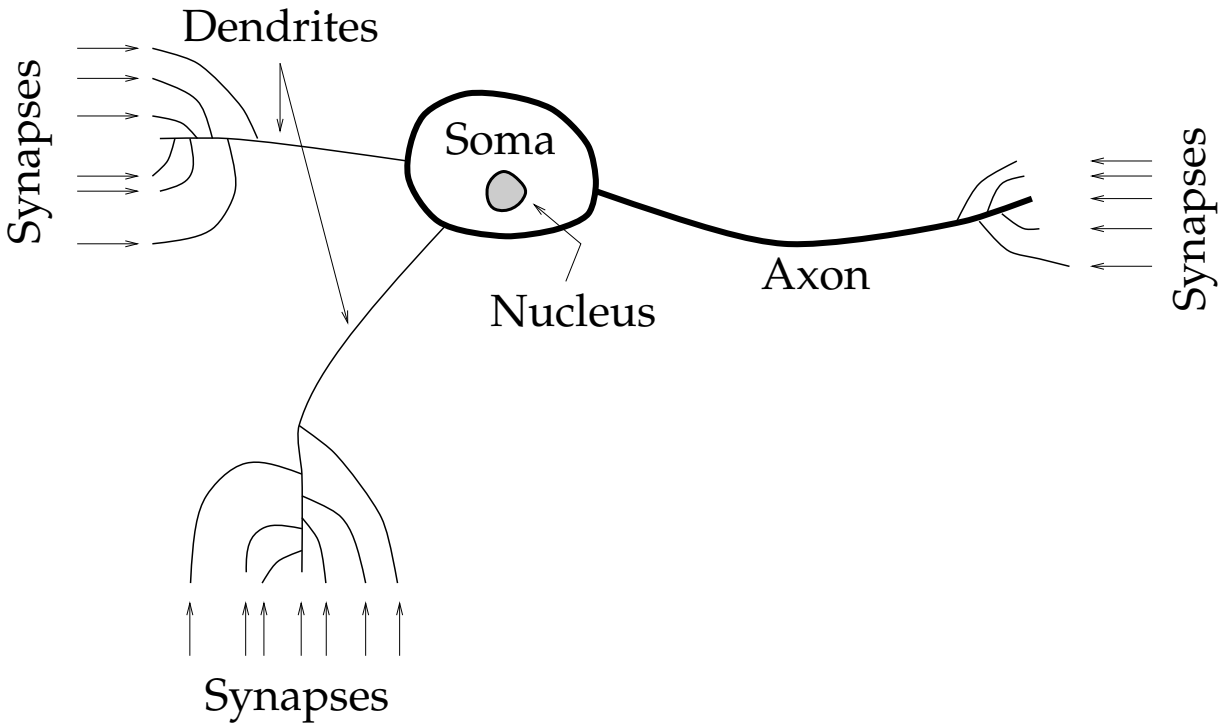
Πολύ περισσότερες πληροφορίες για τα τεχνητά νευρωνικά δίκτυα υπάρχουν στο [Hay94], το [HKP91], το [CB92], καθώς και το [JM96]. Στα [MJ95], [MMJ94], [JZ96], [Dui96] αναφέρονται μέθοδοι ομαδοποίησης και ανακάλυψης σημαντικών χαρακτηριστικών των δεδομένων με χρήση νευρωνικών δικτύων. Το [ZMIR94] περιέχει μερικές αρκετά ενδιαφέρουσες εργασίες, όπως η [JM94], πάνω στη χρήση νευρωνικών δικτύων για ομαδοποίηση - αναγνώριση δεδομένων. Επίσης, το [MR90] εξετάζει τα νευρωνικά συστήματα από την άποψη της στατιστικής φυσικής, ενώ στο [RM86a] παρουσιάζεται η πρωτοποριακή δουλειά των Rumelhart, McClelland και του Parallel Distributed Processing (PDP) Research Group στις αρχιτεκτονικές

των νευρωνικών δικτύων και στους αλγόριθμους εκμάθησης, ενώ στο [RM86b] οι ίδιοι παρουσιάζουν τα αποτελέσματα της έρευνάς τους πάνω στα μοντέλα της ψυχολογίας και της βιολογίας για τον τρόπο λειτουργίας του εγκεφάλου.

Τέλος, στο [Lau92] περιέχονται μερικές από τις πλέον σημαντικές εργασίες που έχουν δημοσιευθεί σχετικώς, όπως μία παρουσίαση του αλγόριθμου εκμάθησης με **οπίσθια μετάδοση σφάλματος** (*back-propagation*) [Wer90], τα ανταγωνιστικού τύπου δίκτυα [CG87], τα Perceptrons [WL90], τον αυτο-οργανωνόμενο χάρτη χαρακτηριστικών [Koh90] και τα δίκτυα τύπου Hopfield [Hop82].

2.1 Ο Νευρώνας

Με την πρωτοποριακή δουλειά του Ramón y Cajál [RyC11], αποδείχθηκε ότι ο ανθρώπινος εγκέφαλος αποτελείται από έναν τεράστιο αριθμό (τουλάχιστον τριάντα δισεκατομμύρια) ειδικών κυττάρων, που καλούνται **νευρώνες** (*neurons*) ή αλλιώς νευρικά κύτταρα. Μία σχηματική παράσταση του νευρώνα φαίνεται στο σχήμα 2.1. Σε αυτό παρατηρούμε ότι ο νευρώνας χωρίζεται σε τέσσερα βασικά τμήματα:



Σχήμα 2.1: Χονδρική αναπαράσταση ενός νευρώνα

1. Τους **δενδρίτες** (*dendrites*), διακλαδισμένες αποφύσεις των νευρώνων. Βρίσκονται σε επαφή μέσω των **συνάψεων** (*synapses*) με τις απολήξεις διαφόρων **νευραξόνων** (*axons*) που προέρχονται από γειτονικούς ή απομακρυσμένους νευρώνες. Οι δενδρίτες συλλέγουν τα σήματα που εκπέμπονται από τις νευραξονικές απολήξεις και τα μεταδίδουν στο αντίστοιχο κυτταρικό σώμα του νευρώνα στον οποίο ανήκουν.

2. Το **σώμα** (*soma*) του νευρώνα, δηλαδή το κυτταρόσωμά του. Περιέχει τον **πυρήνα** (*nucleus*) και άλλα οργανίδια, που επεμβαίνουν στις διάφορες χημικές συνθέσεις.

3. Τον **νευράξονα** (*axon*), *μοναδική* νηματοειδής προέκταση του νευρώνα σε αντίθεση με τις δενδριτικές αποφυάδες που είναι πολλαπλές και διακλαδισμένες. Οι **νευρικές ώσεις** (*impulsion ή influx nerveux*) του κυτταροσώματος του νευρώνα κυκλοφορούν στο νευράξονα, διευθυνόμενες προς την απόληξή του. Οι νευράξονες καταλήγουν στην προσυναπτική μεμβράνη που συμμετέχει στο σχηματισμό της σύναψης.

4. Τις συνάψεις και το **συναπτικό χάσμα** (*synaptic gap*). Οι συνάψεις είναι οργανίδια όπου διαρθρώνεται η απόληξη του νευράξονα ενός νευρώνα με ένα δενδρίτη άλλου νευρώνα, ή η απόληξη ενός νευράξονα με τη μεμβράνη ενός άλλου κυττάρου (μυικού - νευρομυική διάρθρωση - ή αδενικού κυττάρου). Στη σύναψη πραγματοποιείται η μεταβίβαση είτε της νευρικής ώσης από νευρώνα σε νευρώνα (ηλεκτρική σύναψη ή έφαψη), είτε του νευρομεταβιβαστή (χημικές συνάψεις). Στις συνάψεις διακρίνουμε την προσυναπτική και τη μετασυναπτική μεμβράνη, που χωρίζονται με μία πολύ λεπτή σχισμή, το συναπτικό χάσμα.

Οι νευρικές ώσεις που αναφέρθησαν στα ανωτέρω είναι ηλεκτρικά σήματα που παράγονται στο νευρικό κύτταρο και διαδίδονται στο νευράξονά του. **Ριπές ώσεων** (*rafales*) παράγονται στα νευρικά κύτταρα του φλοιού του εγκεφάλου σε απάντηση περιφερειακού ερεθίσματος και μπορούν να καταγραφούν με ένα μικροηλεκτρόδιο.

Αξίζει εδώ να σημειωθεί ότι δεν επικοινωνούν όλοι οι νευρώνες με τη χρήση ηλεκτρικών σημάτων, αλλά ορισμένοι χρησιμοποιούν τους **νευρομεταβιβαστές ή νευροδιαβιβαστές** (*neurotransmitters*). Οι νευρομεταβιβαστές είναι χημικές ουσίες που παράγονται στους νευρώνες και ελευθερώνονται στις απολήξεις των νευραξόνων, προκειμένου να επέμβουν στην μεταβίβαση του νευρικού σήματος στο επίπεδο των συνάψεων που λειτουργούν με χημική διαδικασία. Έχουν απομονωθεί δεκάδες νευρομεταβιβαστές που δρουν στο νευρικό σύστημα και στον εγκέφαλο, π.χ. η ακετυλοχολίνη, διάφορες κατεχολαμίνες (ντοπαμίνη, νοραδρελίνη), η σεροτονίνη και διάφορα νευροπεπτίδια. [1]

Βλέπουμε επομένως ότι ο νευράξονας ενός νευρώνα συνδέεται μέσω των συνάψεων με τους δενδρίτες άλλων νευρώνων. Κατά αυτόν τον τρόπο επιτυγχάνεται η επικοινωνία μεταξύ τους, αφού νευρικές ώσεις που παράγονται στον πυρήνα μεταφέρονται, μέσω του νευράξονα, στις συνάψεις και ακολούθως στους δενδρίτες άλλων νευρώνων όπου με τη σειρά τους προκαλούν άλλες νευρικές ώσεις.

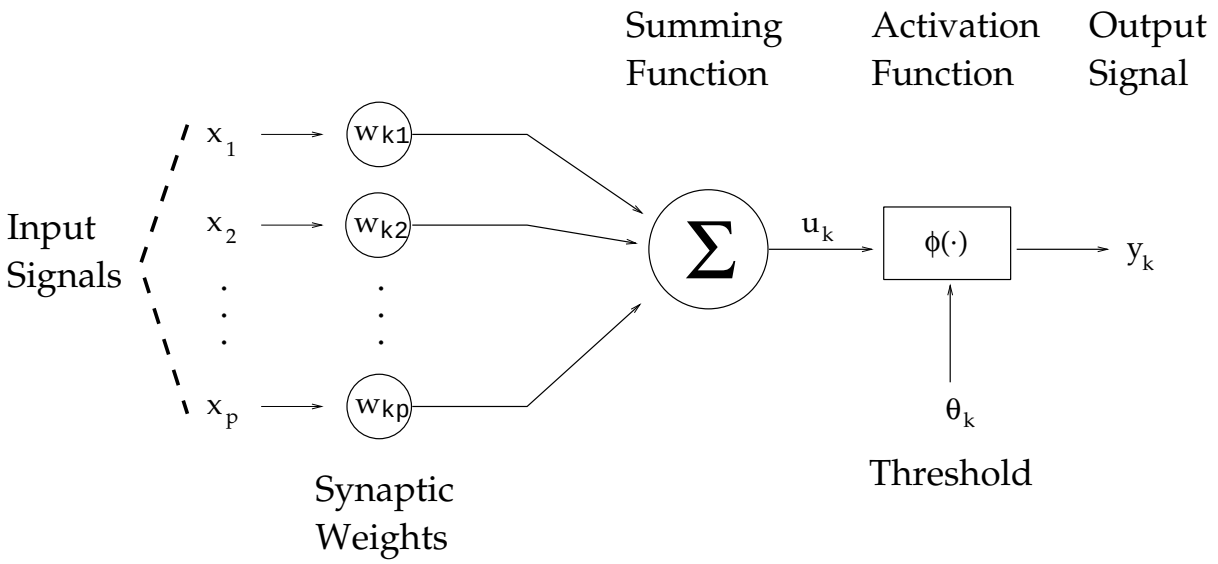
Πολύ περισσότερες πληροφορίες, τόσο για τη φυσιολογία, όσο και για τον τρόπο λειτουργίας του ανθρώπινου εγκεφάλου περιέχονται στο [Cha83].

2.2 Μαθηματικό Μοντέλο του Νευρώνα

Το μαθηματικό μοντέλο του νευρώνα μπορεί να παρασταθεί σχηματικά όπως φαίνεται στο σχήμα 2.2. Σε αυτό διακρίνονται τρία βασικά τμήματα του μοντέλου, σε αντιστοιχία με αυτά του πραγματικού νευρώνα:

1. Οι συνάψεις, κάθε μία εκ των οποίων χαρακτηρίζεται από ένα βάρος. Συγκεκριμένα, το σήμα x_j, στην είσοδο της σύναψης j του νευρώνα k, πολλαπλασιάζεται με το συναπτικό βάρος w_{kj}. Αξίζει εδώ να σημειωθεί ο τρόπος με τον οποίο σημειώνονται

[1] Ενδιαφέρον είναι ότι το κουράριο, είδος δηλητηρίου που παρασκεύαζαν ορισμένες φυλές Ινδιάνων της Νοτίου Αμερικής από διάφορα τροπικά φυτά και χρησιμοποιούσαν στα βέλη τους, έχει ως στόχο του τις μετασυναπτικές μεμβράνες των νευρομυικών συνάψεων, όπου εντοπίζεται ο υποδοχέας της ακετυλοχολίνης. Το κουράριο προσκολλάται στον υποδοχέα και παίρνοντας τη θέση του νευρομεταβιβαστή αυτού εμποδίζει τη διαβίβαση της νευρικής ώσης στο μυν παραλύοντας έτσι το θύμα.



Σχήμα 2.2: Μαθηματικό μοντέλο ενός νευρώνα

οι δείκτες στο βάρος w_{kj}. Ο πρώτος δείκτης αναφέρεται στον συγκεκριμένο νευρώνα και ο δεύτερος στη σύναψη για της οποίας το βάρος μιλάμε. Το βάρος w_{kj} είναι θετικό αν η αντίστοιχη σύναψη είναι **ενισχυτική** (*excitatory*) και αρνητικό αν είναι **αποτρεπτική** (*inhibitory*).

2. Ένας αθροιστής, που προσθέτει όλα τα βεβαρημένα σήματα εισόδου. Η λειτουργία που επιτελείται στο νευρώνα μέχρι και αυτό το σημείο είναι αυτή ενός **γραμμικού μείκτη** (*linear combiner*).

3. Μία συνάρτηση ενεργοποίησης, $\varphi(\cdot)$, η οποία περιορίζει το εύρος της εξόδου του νευρώνα. Τυπικό πεδίο εύρους της εξόδου είναι το [0,1], όμως αρκετά συχνά χρησιμοποιείται και το πεδίο [-1,1].

Το μοντέλο του νευρώνα που φαίνεται στο σχήμα 2.2 περιέχει επίσης ένα **κατώφλι** (*threshold*) θ_k που εφαρμόζεται στο νευρώνα εξωτερικώς. Αυτό χρησιμοποιείται προκειμένου να μειώσει τη συνολική είσοδο που παρέχεται στη συνάρτηση ενεργοποίησης. Μπορεί όμως να χρησιμοποιηθεί και για τον αντίθετο σκοπό, δηλαδή ώς ένας όρος **επιρροής** (*bias*), αντί για κατώφλι. Ο όρος επιρροής απλώς θα έχει αντίθετο πρόσημο από το κατώφλι.

Η λειτουργία επομένως του νευρώνα k μπορεί να περιγραφεί από τις εξισώσεις 2.1 και 2.2.

$$u_k = \sum_{j=1}^{p} w_{kj} x_j \tag{2.1}$$

$$y_k = \varphi(u_k - \theta_k) \tag{2.2}$$

όπου $x_1, x_2, \ldots, x_p$ είναι τα σήματα εισόδου, $w_{k1}, w_{k2}, \ldots, w_{kp}$ είναι τα συναπτικά βάρη του νευρώνα k, u_k είναι η έξοδος του γραμμικού μείκτη, θ_k είναι το κατώφλι, $\varphi(\cdot)$ είναι η συνάρτηση ενεργοποίησης και y_k είναι το σήμα εξόδου του νευρώνα.

Η χρήση του κατωφλιού έχει ως αποτέλεσμα την εφαρμογή ενός **αφφινικού μετασχηματισμού** (*affine transformation*) στην έξοδο u_k του γραμμικού μείκτη, όπως φαίνεται και στην εξίσωση 2.3.

$$\upsilon_k = u_k - \theta_k \tag{2.3}$$

Όπως αναφέρθηκε ήδη, το κατώφλι είναι μία εξωτερική παράμετρος του τεχνητού νευρώνα k. Μπορούμε να το χρησιμοποιήσουμε στο μοντέλο όπως στην εξίσωση 2.2 ή να συνδυάσουμε τις εξισώσεις 2.1 και 2.2 ως εξής:

$$\upsilon_k = \sum_{j=0}^{p} w_{kj} x_j \tag{2.4}$$

και

$$y_k = \varphi(\upsilon_k) \tag{2.5}$$

Στην εξίσωση 2.4 έχει προστεθεί μία ακόμα σύναψη (με δείκτη μηδέν), της οποίας η είσοδος είναι πάντοτε σταθερή

$$x_0 = -1$$

και της οποίας το βάρος είναι

$$w_{k0} = \theta_k$$

Μπορούμε επομένως να μετατρέψουμε το μοντέλο του νευρώνα k σε αυτό που φαίνεται στο σχήμα 3(α). Σε αυτό, το κατώφλι αντιπροσωπεύεται από την προσθήκη μίας νέας σύναψης με σταθερή είσοδο $x_0 = -1$ και με βάρος ίσο προς το κατώφλι θ_k. Αντίστοιχα, μπορούμε να μετατρέψουμε το μοντέλο όπως στο σχήμα 3(β), όπου θεωρώντας την είσοδο της νέας σύναψης ίση με $x_0 = +1$ και το βάρος $w_{k0} = b_k$ την μετατρέπουμε στον όρο bias. Παρόλο που τα μοντέλα των σχημάτων 3(α) και 3(β) φαίνονται διαφορετικά, από μαθηματικής άποψης θεωρούνται ισοδύναμα.

2.2.1 Συναρτήσεις Ενεργοποίησης

Η συνάρτηση ενεργοποίησης, που συμβολίζεται στα ανωτέρω ως $\varphi(\cdot)$, ορίζει την έξοδο του νευρώνα σε σχέση με το επίπεδο ενεργοποίησης των εισόδων του. Οι συναρτήσεις ενεργοποίησης χωρίζονται σε τρία βασικά είδη:

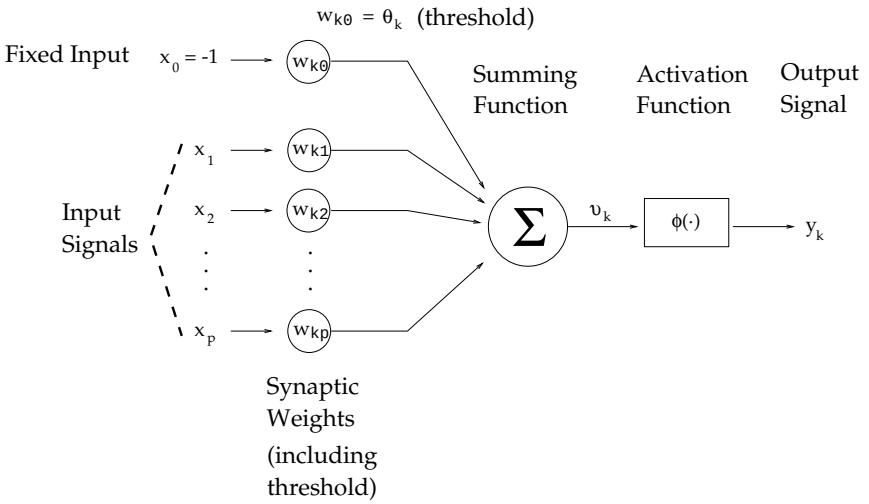
1. **Συναρτήσεις κατωφλίου** (*Threshold functions*). Οι συναρτήσεις αυτού του είδους, που γραφικώς ομοιάζουν με αυτή του σχήματος 2.4, έχουν τη γενική μορφή της εξίσωσης 2.6.

$$\varphi(\upsilon) = \begin{cases} 1 & , \upsilon \geq 0 \\ 0 & , \upsilon < 0 \end{cases} \tag{2.6}$$

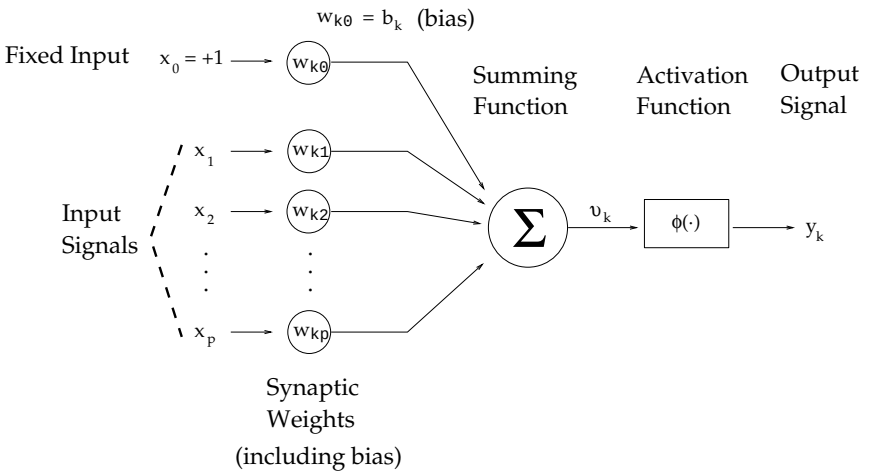
Αντιστοίχως, η έξοδος του νευρώνα k έχει τη μορφή της εξίσωσης 2.7,

$$y_k = \begin{cases} 1 & , \upsilon_k \geq 0 \\ 0 & , \upsilon_k < 0 \end{cases} \tag{2.7}$$

όπου υ_k είναι η εσωτερική ενεργοποίηση του νευρώνα, δηλαδή:



(α)



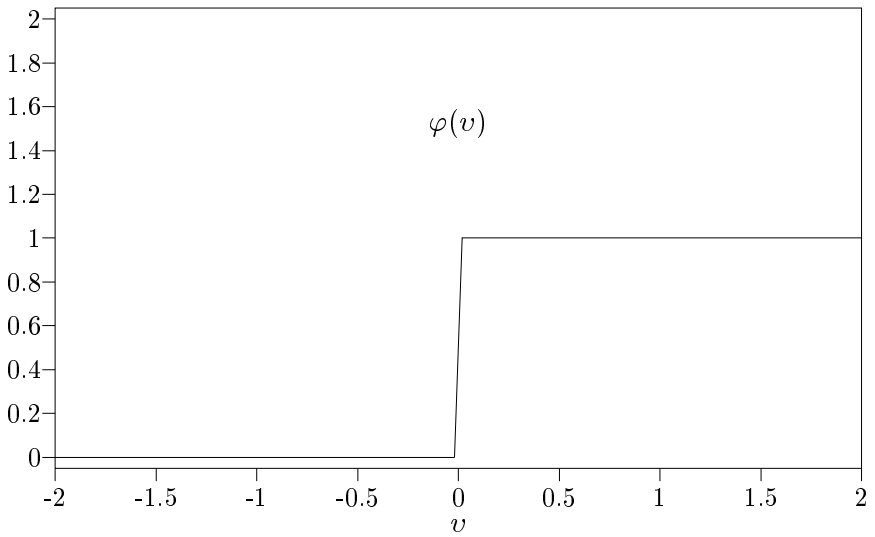
(β)

Σχήμα 2.3: Δύο παραλλαγές του μοντέλου του νευρώνα k

$$\upsilon_k = \sum_{j=1}^{p} x_{kj} x_j - \theta_k$$

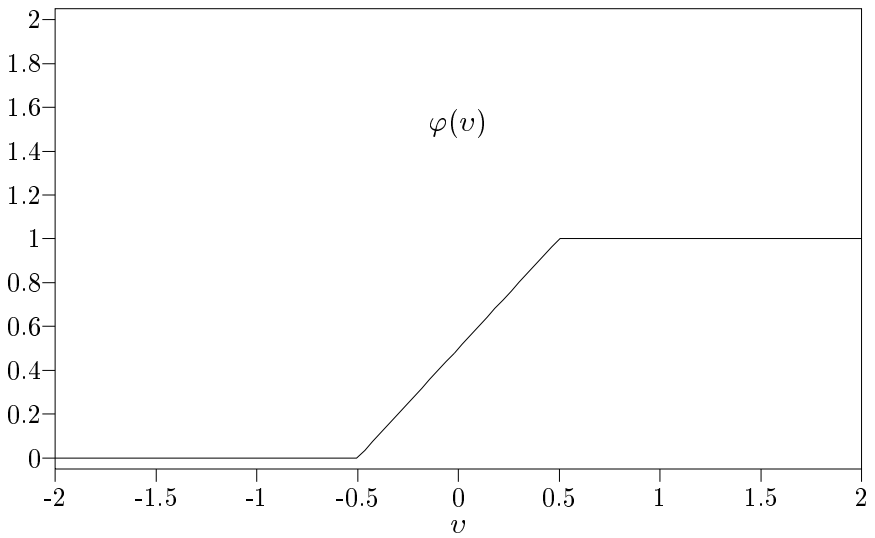
Ένας νευρώνας τέτοιου είδους αναφέρεται στη βιβλιογραφία ως μοντέλο των *McCulloch-Pitts*. Η έξοδος του νευρώνα λαμβάνει την τιμή 1 εάν η εσωτερική του ενεργοποίηση είναι μη αρνητική και την τιμή 0 ειδάλλως. Για το λόγο αυτό, λέγεται ότι το μοντέλο των *McCulloch-Pitts* διακρίνεται από την ιδιότητα **όλα ή τίποτα** (*all-or-none*).

2. **Τμηματικώς-Γραμμικές συναρτήσεις** (*Piecewise - Linear*). Μία τμηματικώς-γραμμική συνάρτηση, που γραφικώς ομοιάζει με αυτή του σχήματος 2.5, έχει εν γένει τη μορφή της εξίσωσης 2.8.



Σχήμα 2.4: Συνάρτηση κατωφλίου

$$\varphi(v) = \begin{cases} 1 & , v \geq \frac{1}{2} \\ \frac{1}{2} + v & , \frac{1}{2} > v > -\frac{1}{2} \\ 0 & , v \leq -\frac{1}{2} \end{cases} \tag{2.8}$$

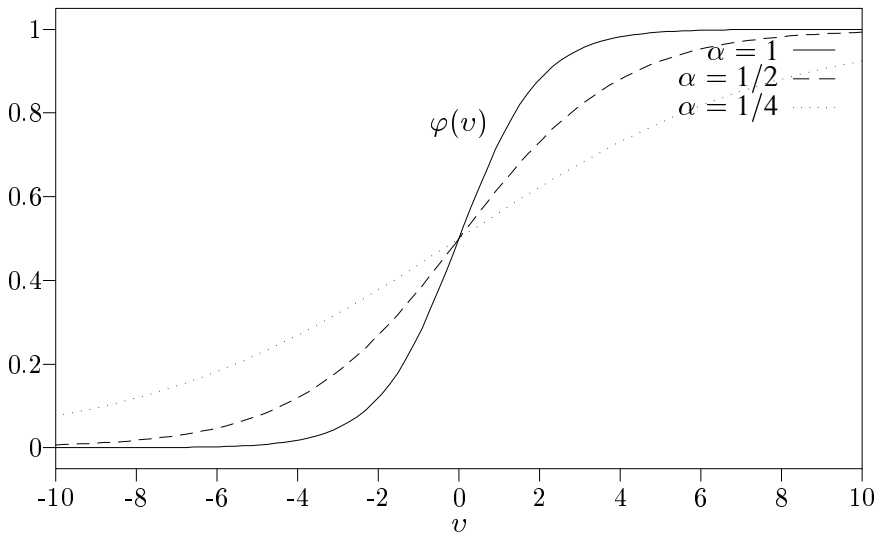


Σχήμα 2.5: Τμηματικώς-Γραμμική συνάρτηση

3. **Σιγμοειδείς συναρτήσεις** (*Sigmoid functions*). Οι σιγμοειδείς συναρτήσεις είναι οι πλέον διαδεδομένες στα τεχνητά νευρωνικά δίκτυα. Πρόκειται για μία οικογένεια αυξουσών συναρτήσεων που έχουν τις πολύ ‘‘καλές’’ μαθηματικές ιδιότητες της συνέχειας και παραγωγισημότητας σε όλο το διάστημα ορισμού τους, ακόμα και στα άκρα αυτού. Μία τέτοια συνάρτηση είναι η λεγόμενη **λογιστική συνάρτηση** (*logistic function*), η οποία έχει τη μορφή της εξίσωσης 2.9:

$$\varphi(v) = \frac{1}{1 + \exp(-\alpha v)} \tag{2.9}$$

όπου το α είναι η **παράμετρος κλίσης** (*slope parameter*) της σιγμοειδούς συνάρτησης. Μεταβάλλοντας τις τιμές της παραμέτρου α, λαμβάνουμε σιγμοειδείς συναρτήσεις διαφορετικών κλίσεων, όπως φαίνεται στο σχήμα 2.6.



Σχήμα 2.6: Τμηματικώς-Γραμμική συνάρτηση

Καθώς η παράμετρος κλίσης α προσεγγίζει το άπειρο, η σιγμοειδής συνάρτηση προσεγγίζει την συνάρτηση κατωφλίου. Ενώ όμως η συνάρτηση κατωφλίου λαμβάνει μόνο δύο τιμές, 0 ή 1, η σιγμοειδής λαμβάνει όλες τις τιμές μεταξύ του 0 και 1. Επιπλέον, όπως ήδη αναφερθεί, η σιγμοειδής συνάρτηση παραγωγίζεται, αντιθέτως με τη συνάρτηση κατωφλίου. Αυτή της η ιδιότητα είναι πολύ σημαντική για τη θεωρία των νευρωνικών δικτύων.

Οι συναρτήσεις ενεργοποίησης που ορίσθηκαν στις εξισώσεις 2.6, 2.8 και 2.9, έχουν πεδίο τιμών το διάστημα $[0, +1]$. Μερικές φορές, όμως, είναι επιθυμητό το πεδίο τιμών να είναι το διάστημα $[-1, +1]$. Στις περιπτώσεις αυτές, η συνάρτηση ενεργοποίησης λαμβάνει μία αντισυμμετρική μορφή ως προς την αρχή των αξόνων των συντεταγμένων. Πιο συγκεκριμένα, η συνάρτηση κατωφλίου της εξίσωσης 2.6 ορίζεται πλέον όπως στην εξίσωση 2.10.

$$\varphi(v) = \begin{cases} 1 & , v > 0 \\ 0 & , v = 0 \\ -1 & , v < 0 \end{cases} \tag{2.10}$$

Η συνάρτηση αυτή ονομάζεται **συνάρτηση σημείου ή προσήμου** (*signum function*). Αντιστοίχως, αντί της σιγμοειδούς συνάρτησης μπορεί να χρησιμοποιηθεί η **συνάρτηση της υπερβολικής εφαπτομένης** (*hyperbolic tangent function*), που ορίζεται όπως στην εξίσωση 2.11.

$$\varphi(v) = \tanh(\frac{v}{2}) = \frac{1 - \exp(-v)}{1 + \exp(-v)} \tag{2.11}$$

Αξίζει να σημειωθεί ότι η χρήση μίας σιγμοειδούς συνάρτησης ενεργοποίησης που λαμβάνει και αρνητικές τιμές δεν έχει μόνο πλεονεκτήματα κατά την θεωρητική ανάλυση των δικτύων, αλλά στηρίζεται και από πειραματικά ευρήματα της νευροφυσιολογίας.

2.3 Αρχιτεκτονικές Νευρωνικών Δικτύων

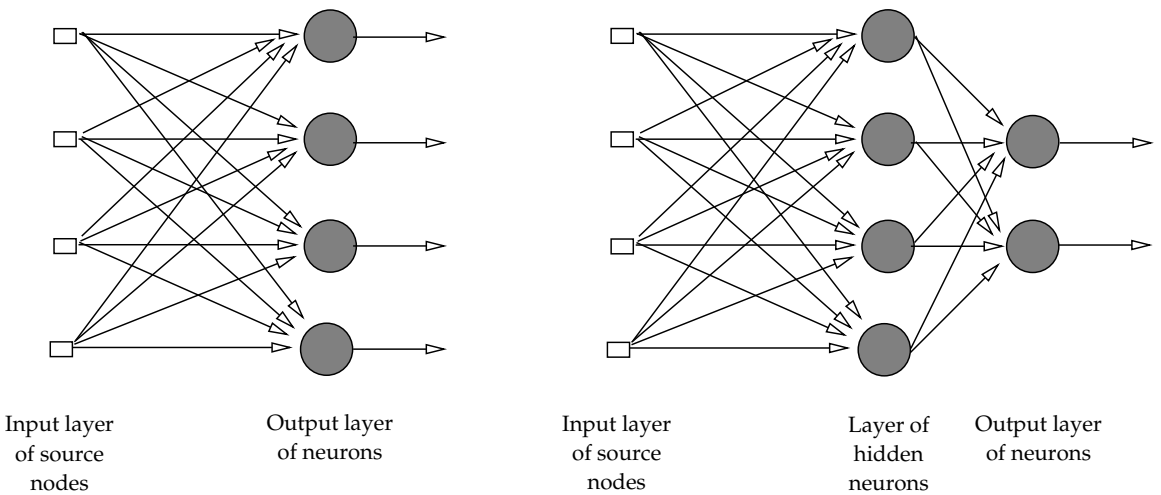
Όπως έχει ήδη αναφερθεί, οι νευρώνες του εγκεφάλου συνδέονται με διάφορους τρόπους προκειμένου να σχηματίσουν υπολογιστικά συστήματα μεγαλύτερης περιπλοκότητας.

Στην πάροδο του χρόνου έχουν προταθεί διάφοροι τρόποι δικτύωσης των τεχνητών νευρώνων, η πλειονότητα των οποίων έχει βασιστεί σε αντίστοιχους τρόπους δικτύωσης των πραγματικών νευρώνων.

Ένας διαχωρισμός που μπορεί να γίνει στις αρχιτεκτονικές των δικτύων είναι αναλόγως με τα **στρώματα** (*layers*) στα οποία χωρίζουν τους νευρώνες. Έτσι, έχουμε:

1. **Δίκτυα εμπρόσθιας μετάδοσης ενός στρώματος** (*Single-layer feedforward networks*). Σε αυτά τα δίκτυα, όπως φαίνεται και στο σχήμα 2.7(α), οι νευρώνες βρίσκονται όλοι στο ίδιο στρώμα και οι συνδέσεις είναι *αποκλειστικώς* από τις εισόδους στους νευρώνες. Δηλαδή δεν επιτρέπεται ούτε κύκλος, ούτε σύνδεση από ένα νευρώνα σε άλλον του ιδίου στρώματος.

2. **Δίκτυα εμπρόσθιας μετάδοσης πολλών στρωμάτων** (*Multilayer feedforward networks*). Η διαφορά αυτών των δικτύων από τα προηγούμενα είναι ότι διαθέτουν περισσότερα του ενός στρώματα. Όπως όμως φαίνεται και στο σχήμα 2.7(β), ούτε σε αυτά επιτρέπονται συνδέσεις νευρώνων του ιδίου στρώματος. Επίσης δεν επιτρέπεται να συνδέεται η έξοδος ενός νευρώνα σε νευρώνα χαμηλότερου στρώματος. Εξ ου και ο όρος εμπρόσθια μετάδοση που χαρακτηρίζει τον τρόπο με τον οποίο μεταδίδονται τα αρχικά σήματα εισόδου στο νευρωνικό δίκτυο.

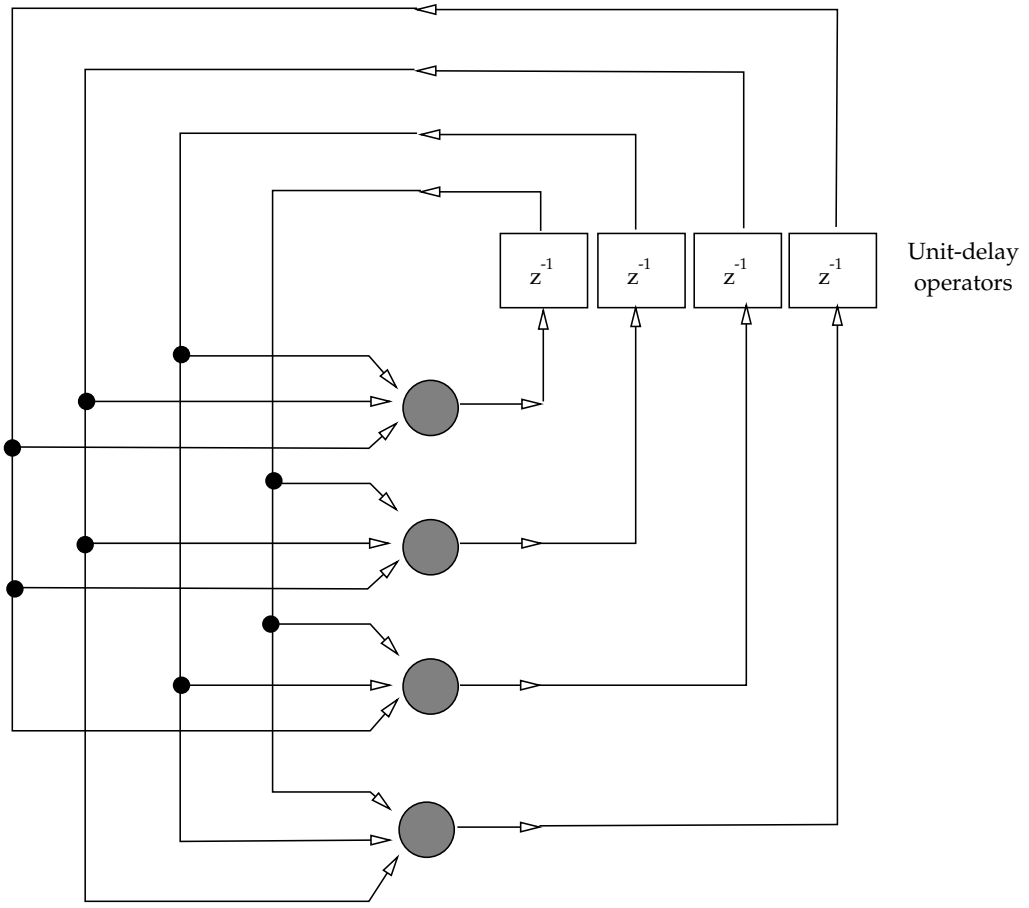


(α) Δίκτυο ενός στρώματος (β) Δίκτυο πολλών στρωμάτων

Σχήμα 2.7: Δίκτυα εμπρόσθιας μετάδοσης

3. **Αναδραστικά δίκτυα** (*Recurrent networks*). Τα αναδραστικά δίκτυα είναι παρόμοια με τα δίκτυα εμπρόσθιας μετάδοσης, αλλά όπως φαίνεται και στο σχήμα 2.8, επιτρέπουν

επιπλέον την ύπαρξη κύκλων στις συνδέσεις, καθώς και τη δυνατότητα σύνδεσης της εξόδου ενός νευρώνα είτε στον ίδιο ή και σε νευρώνα χαμηλότερου στρώματος.



Σχήμα 2.8: Αναδραστικό δίκτυο

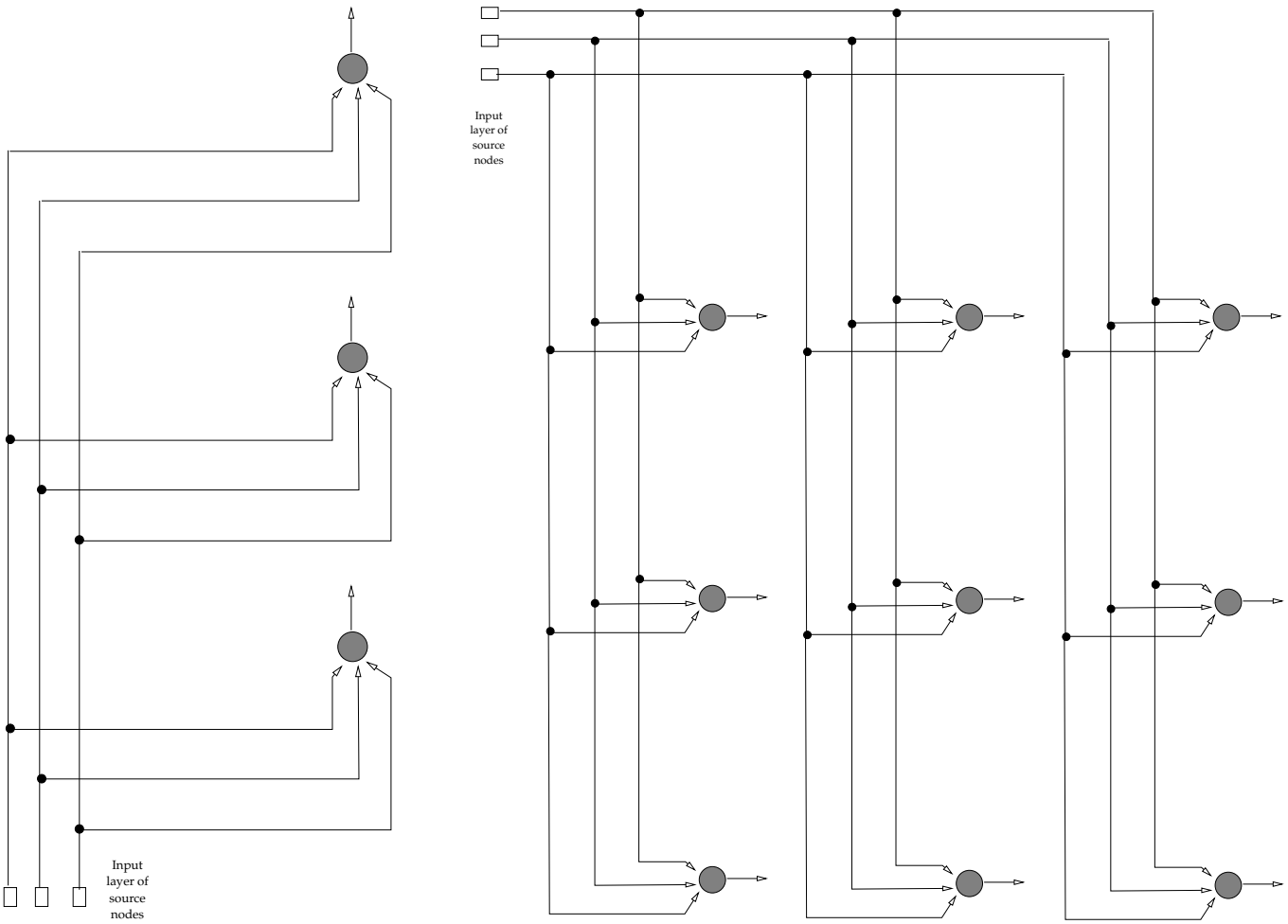
4. **Δίκτυα με δομή πίνακα** (*Lattice structures*). Στα δίκτυα αυτού του τύπου, οι νευρώνες παρατάσσονται σχηματίζοντας ένα μονοδιάστατο (βλέπε το σχήμα 2.9(α)), δισδιάστατο (βλέπε το σχήμα 2.9(β)) ή και πολυδιάστατο πίνακα. Οι είσοδοι του δικτύου συνδέονται σε όλους τους νευρώνες αυτού, όπως φαίνεται και στα προαναφερθέντα σχήματα.

   Στην πραγματικότητα, τα δίκτυα με δομή πίνακα είναι μία περίπτωση των δικτύων εμπρόσθιας μετάδοσης, με τους νευρώνες εξόδου παρατεταγμένους σε σειρές και στήλες.

2.4 Η Διαδικασία Εκμάθησης

Προτού εξετάσουμε τη διαδικασία της εκμάθησης των νευρωνικών δικτύων, θα πρέπει να δώσουμε έναν ορισμό της έννοιας της εκμάθησης.

Όπως αναφέρεται και στο [Hay94], *εκμάθηση είναι η διαδικασία μέσω της οποίας προσαρμόζονται οι ελεύθερες παράμετροι του δικτύου, μέσω της συνεχούς παροχής ερεθισμάτων από το εξωτερικό περιβάλλον. Ο τύπος της εκμάθησης καθορίζεται από τον τρόπο με τον οποίο γίνεται η προσαρμογή των παραμέτρων.*



(α) Δίκτυο με δομή μονοδιάστατου πίνακα τριών νευρώνων

(β) Δίκτυο με δομή δισδιάστατου πίνακα 3×3

Σχήμα 2.9: Δίκτυα με δομή πίνακα

Ο ορισμός αυτός της εκμάθησης συνεπάγεται την ακόλουθη ακολουθία από γεγονότα:

1. Παρέχονται ερεθίσματα στο νευρωνικό δίκτυο από το περιβάλλον.
2. Τα ερεθίσματα αυτά προκαλούν αλλαγές στο δίκτυο.
3. Το δίκτυο απαντά με διαφορετικό τρόπο στο περιβάλλον του, εξαιτίας των αλλαγών που έχουν συμβεί στην εσωτερική του δομή.

Μέχρι του παρόντος, έχουν προταθεί τρία **παραδείγματα** (*paradigms*) της διαδικασίας της εκμάθησης:

1. **Καθοδηγούμενη εκμάθηση** (*Supervised learning*). Στο παράδειγμα αυτό, εκτός από το σήμα εισόδου που δίνεται στο δίκτυο, υπάρχει και κάποιος εξωτερικός διδάσκαλος που παρέχει σε αυτό την σωστή απάντηση που θα έπρεπε να δώσει κάθε φορά. Έτσι, το δίκτυο μπορεί να υπολογίσει τη διαφορά της δικής του απάντησης από την επιθυμητή και να αναπροσαρμόσει αναλόγως τις ελεύθερες παραμέτρους του.

2. **Εκμάθηση βασισμένη στην ενδυνάμωση** (*Reinforcement learning*). Στο παράδειγμα αυτό, το δίκτυο τείνει να ενδυναμώνει τις εσωτερικές παραμέτρους που συμμετέχουν στην παραγωγή εξόδων που θεωρούνται καλές από το περιβάλλον. Δηλαδή, όπως και στην καθοδηγούμενη εκμάθηση, έτσι κι εδώ υπάρχει κάποιος εξωτερικός διδάσκαλος. Στην περίπτωση αυτή, όμως, δεν παρέχει στο δίκτυο την επιθυμητή απάντηση, αλλά μόνο μία ένδειξη που του λέει αν η απάντηση που έδωσε ήταν ικανοποιητική ή όχι. Τέτοιου είδους εκμάθηση εμφανίζεται π.χ. στην εκπαίδευση των ζώων, όπου αυτά αμοίβονται όταν κάνουν αυτό που θέλουμε και τιμωρούνται στην αντίθετη περίπτωση. Έτσι, τα ζώα τείνουν να επαναλαμβάνουν πράξεις τους που οδήγησαν σε ανταμοιβή τους, όπως το φαγητό, και να περιορίσουν τις υπόλοιπες. Χρησιμοποιείται όταν είναι αδύνατον να δώσουμε τη σωστή απάντηση στο δίκτυο (όπως στην περίπτωση των ζώων).

3. **Εκμάθηση βασισμένη στην αυτο-οργάνωση** (*Self-organised (unsupervised) learning*). Στο παράδειγμα αυτό δεν υπάρχει κάποιος εξωτερικός διδάσκαλος να καθοδηγεί το δίκτυο, αλλά αυτό προσπαθεί να αναπαραστήσει την είσοδό του με τον οικονομικότερο δυνατόν τρόπο, προσπαθώντας να βρει αναλογίες μεταξύ των διαφόρων εισόδων.

Οι προσαρμογές των βαρών στην κάθε περίπτωση μπορούν να γίνουν με διάφορους τρόπους, οι οποίοι καλούνται **κανόνες εκμάθησης** (*learning rules*). Οι βασικότεροι εξ αυτών είναι οι εξής τέσσερις:

1. Ο κανόνας **διόρθωσης σφάλματος** (*error-correction*), που εφαρμόζεται στο παράδειγμα της καθοδηγούμενης εκμάθησης. Σε αυτόν, τη χρονική στιγμή t, παρέχεται στο νευρώνα το σήμα εισόδου $\vec{x}(t)$ και η σωστή απάντηση $d_k(t)$. Έτσι, εξ αυτών μπορεί να υπολογισθεί το σφάλμα της τρέχουσας απάντησης, $y_k(t)$, το οποίο θα είναι:

$$e_k(t) = d_k(t) - y_k(t)$$

   Με βάση την τιμή αυτή του σφάλματος, επαναϋπολογίζονται τα συναπτικά βάρη συμφώνως προς την εξίσωση:

$$\vec{w}_k(t+1) = \vec{w}_k(t) + \eta * e_k(t) * \vec{x}(t)$$

   όπου η είναι μία θετική σταθερά που καθορίζει το ρυθμό της εκμάθησης.

2. Ο κατά Hebb κανόνας εκμάθησης, ονομασθείς προς τιμήν του νευροψυχολόγου Hebb που τον πρωτοδιατύπωσε [Heb49], που μπορεί να περιγραφεί συνοπτικώς ως εξής:

   (a) Εάν δύο νευρώνες ενεργοποιούνται **συγχρόνως** (*synchronously*), τότε η μεταξύ τους σύναψη ενισχύεται.

   (b) Εάν δύο νευρώνες ενεργοποιούνται **ασυγχρόνως** (*asynchronously*), τότε η μεταξύ τους σύναψη αποδυναμώνεται ή και απαλείφεται.

   Η απλούστερη συνάρτηση μεταβολής των συναπτικών βαρών που ικανοποιεί τον κανόνα εκμάθησης κατά Hebb, είναι η κάτωθι:

$$\vec{w}_k(t+1) = \vec{w}_k(t) + \eta * y_k(t) * \vec{x}(t)$$

   Η συνάρτηση αυτή αυξάνει τα συναπτικά βάρη όποτε η έξοδος $y_k(t)$ και η είσοδος $\vec{x}(t)$ έχουν το ίδιο πρόσημο και τα μειώνει αντιστοίχως όταν έχουν αντίθετα πρόσημα.

3. Ο **ανταγωνιστικός** (*competitive*) κανόνας εκμάθησης, που όπως αναφέρεται και στο [RZ86], περιλαμβάνει δύο βασικά στοιχεία:

   (a) Ένα σύνολο νευρώνων που είναι καθόλα ίδιοι, εκτός από τα συναπτικά τους βάρη τα οποία έχουν αρχικώς τυχαίως κατανεμημένες τιμές, και τα οποία τους αναγκάζουν να απαντούν διαφορετικώς στο ίδιο σήμα εισόδου.

   (b) Έναν μηχανισμό που επιτρέπει στους νευρώνες να ανταγωνίζονται μεταξύ τους για το δικαίωμα απάντησης σε ένα συγκεκριμένο υποσύνολο σημάτων εισόδου, έτσι ώστε μόνον ένας νευρώνας να απαντά κάθε φορά. Ο νευρώνας αυτός καλείται νευρώνας - νικητής.

   Υπαρχόντων των ανωτέρω δύο στοιχείων, και αφού έχει επιλεχθεί ο νευρώνας - νικητής μέσω του μηχανισμού 3b, η μεταβολή των συναπτικών βαρών γίνεται συμφώνως προς την εξίσωση:

$$\vec{w}_k(t+1) = \vec{w}_k(t) + \begin{cases} \eta * (\vec{x}(t) - \vec{w}_k(t)) & \text{εάν ο νευρώνας k είναι ο νικητής} \\ 0 & \text{αλλιώς} \end{cases}$$

4. Ο κατά Boltzmann κανόνας εκμάθησης, ονομασθείς προς τιμήν του γνωστού φυσικού Boltzmann, ο οποίος είναι ένας **στοχαστικός** (*stochastic*) κανόνας εκμάθησης που εξήλθε από τη χρήση κανόνων της Θερμοδυναμικής και της Θεωρίας της Πληροφορίας.

   Η βασική ιδέα του κανόνα αυτού είναι πως σε ένα αναδραστικό δίκτυο που οι νευρώνες λειτουργούν κατά δυαδικό τρόπο, δηλαδή απαντούν πάντοτε είτε $+1$ ή -1, μπορεί να ορισθεί μία **συνάρτηση ενέργειας** (*energy function*) E, η τιμή της οποίας καθορίζεται από τις καταστάσεις στις οποίες βρίσκονται οι νευρώνες, συμφώνως προς την εξίσωση:

$$E = -\frac{1}{2}\sum_i \sum_j w_{ij} s_i s_j \qquad \text{για } i \neq j$$

   όπου s_i είναι η κατάσταση του νευρώνα i (δηλαδή το αν απαντά θετικά ή αρνητικά) και w_{ij} είναι το βάρος της σύναψης που συνδέει το νευρώνα i στο νευρώνα j. Το γεγονός ότι $i \neq j$ σημαίνει ότι δεν επιτρέπεται αυτο-ανάδραση σε κανένα από τους νευρώνες του δικτύου.

   Το δίκτυο λειτουργεί επιλέγοντας τυχαίως κάποιον από τους νευρώνες, έστω τον νευρώνα j, και αντιστρέφοντας την κατάστασή του από s_i σε $-s_i$ με πιθανότητα

$$W(s_i \to -s_i) = \frac{1}{1 + \exp(-\frac{\Delta E_j}{T})}$$

   όπου ΔE_j είναι η μεταβολή της ενέργειας του δικτύου που θα προκληθεί από μία τέτοια αντιστροφή και T είναι μία μεταβαλλόμενη ποσότητα του αλγόριθμου, καλούμενη **θερμοκρασία** (*temperature*) του δικτύου, η οποία δεν σχετίζεται με την πραγματική θερμοκρασία αυτού, αλλά παίζει το ρόλο του ρυθμού εκμάθησης, η, που εμφανίζεται στους άλλους κανόνες εκμάθησης.

   Εάν ο κανόνας αυτός εφαρμόζεται συνεχώς, τότε το δίκτυο θα καταλήξει στην καλούμενη **κατάσταση θερμικής ισορροπίας** (*thermal equilibrium state*), όπου πλέον η

ενέργειά του θα έχει λάβει τη χαμηλότερη τιμή της και η θερμοκρασία του θα έχει πρακτικώς μηδενιστεί.

Αξίζει εδώ να σημειωθεί ότι ο κανόνας αυτός επιτυγχάνει την εύρεση του *ολικού βέλτιστου* της συνάρτησης ενέργειας του δικτύου και δεν παγιδεύεται σε τοπικά βέλτιστα αυτής. Όμως χρειάζεται πολύ χρόνο προκειμένου να φτάσει το δίκτυο στην κατάσταση θερμικής ισορροπίας.

2.5 Ομαδοποίηση με τα Νευρωνικά Δίκτυα

Δεδομένου ότι η παρούσα εργασία εξετάζει την ομαδοποίηση δεδομένων, στη συνέχεια θα εξεταστούν τα νευρωνικά δίκτυα μόνον ως προς τη δυνατότητά τους να επιτελούν το έργο αυτό.

Όπως παρουσιάζεται και στο [Lip87], πολλοί διαφορετικοί τύποι δικτύων δύνανται να χρησιμοποιηθούν για το σκοπό αυτό. Πλην όμως, μόνον ο λεγόμενος **αυτο-οργανωνόμενος χάρτης χαρακτηριστικών** (*Self-Organizing Feature Map* (SOFM)) έχει ταυτοχρόνως δύο πολύ σημαντικά χαρακτηριστικά:

1. Τη δυνατότητα να χειρίζεται δεδομένα εισόδου που λαμβάνουν συνεχείς τιμές, δηλαδή διανύσματα που ανήκουν στο χώρο $\mathbb{R}^n$.

2. Τη δυνατότητα να εφαρμόσει το παράδειγμα της εκμάθησης βασισμένης στην αυτο-οργάνωση, δίχως δηλαδή να χρειάζεται την παρουσία ενός εξωτερικού διδασκάλου.

Καθώς και τα δύο αυτά χαρακτηριστικά αποτελούν βασικούς περιορισμούς του προβλήματος ομαδοποίησης δοσοληψιών συμφώνως προς τα χαρακτηριστικά του φόρτου εργασίας τους, που εξετάζεται στην παρούσα εργασία, στη συνέχεια θα εξεταστεί η αρχιτεκτονική SOFM μόνο, καθώς και αυτή του αναπροσαρμοζόμενου K-MEANS αλγόριθμου (βλέπε το κεφάλαιο 3), που μπορεί να θεωρηθεί ως μία υποπερίπτωση του SOFM.

2.5.1 Ο Αυτο-Οργανωνόμενος Χάρτης Χαρακτηριστικών

Ένα πολύ σημαντικό χαρακτηριστικό της οργάνωσης του τμήματος του εγκεφάλου που επεξεργάζεται τα σήματα που λαμβάνουμε μέσω των αισθητηρίων οργάνων μας, είναι ότι οι νευρώνες παρουσιάζουν μία κανονικότητα ως προς τη θέση τους και συχνά η οργάνωσή τους αντικατοπτρίζει ορισμένα φυσικά χαρακτηριστικά της συγκεκριμένης αίσθησης που επεξεργάζονται. Για παράδειγμα, σε κάθε επίπεδο της **ακουστικής οδού** (*auditory pathway*), τα νευρικά κύτταρα είναι τοποθετημένα αναλόγως προς τη συγκεκριμένη συχνότητα στην οποία αντιδρούν περισσότερο.

Παρόλο που μεγάλο ποσοστό της χαμηλού επιπέδου οργάνωσης είναι προκαθορισμένη γενετικώς, είναι πιθανό κάποια από την οργάνωση στα υψηλότερα επίπεδα να δημιουργείται κατά τη διάρκεια της εκμάθησης από αλγόριθμους αυτο-οργάνωσης. Ο Kohonen [Koh82], [Koh90], δημιούργησε έναν τέτοιον αλγόριθμο, ο οποίος παράγει αυτο-οργανωνόμενους χάρτες χαρακτηριστικών, παρόμοιους με αυτούς που εμφανίζονται στον εγκέφαλο. Ο αλγόριθμος αυτός χρησιμοποιεί τον ανταγωνιστικό κανόνα εκμάθησης (βλέπε σελίδα 17).

Σκοπός του SOFM είναι να παράγει έναν **κβαντοποιητή διανυσμάτων** (*vector quantizer*), δηλαδή ένα σύστημα που να αντιστοιχεί σε συγκεκριμένα σύνολα διανυσμάτων εισόδου κάποια διανύσματα αναφοράς, τα οποία να περιγράφουν όσο το δυνατόν καλύτερα το σύνολο στο οποίο αντιστοιχούν. Τα διανύσματα αναφοράς αυτά τα αποθηκεύει στα διανύσματα των συναπτικών βαρών των νευρώνων που αποτελούν το δίκτυο.

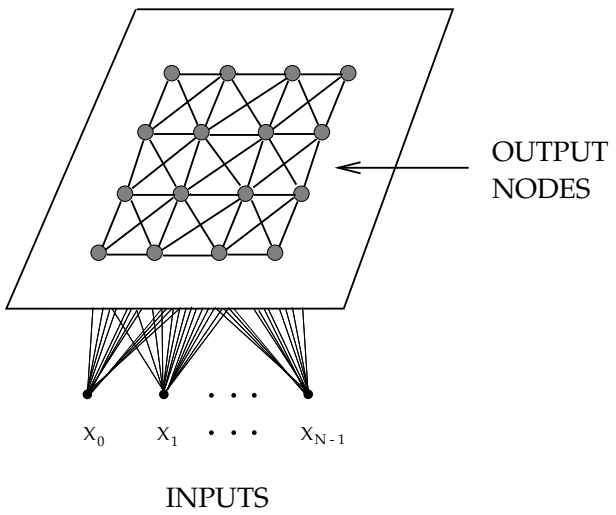
Μία σχηματική παράσταση του δικτύου που υλοποιεί τον αλγόριθμο SOFM φαίνεται στο σχήμα 2.10. Όπως φαίνεται σε αυτό, οι, M τον αριθμό, κόμβοι της εξόδου είναι παρατεταγμένοι σε ένα δισδιάστατο δίχτυ και είναι συνδεδεμένοι σε μεγάλο βαθμό μεταξύ τους. Επίσης, οι, N τον αριθμό, κόμβοι της εισόδου συνδέονται με κάθε έναν από τους κόμβους εξόδου, όπως δείχνει και το σχήμα 2.11.

Προκειμένου να κατασκευαστούν τα διανύσματα αναφοράς, ο SOFM αναγκάζει τους νευρώνες του δικτύου να ανταγωνίζονται για το ποιός εξ αυτών θα αντιπροσωπεύσει το εκάστοτε διάνυσμα εισόδου. Αυτό γίνεται κατά την εμφάνιση του διανύσματος εισόδου, όπου όλοι οι νευρώνες υπολογίζουν την απόσταση που έχουν από αυτό, δηλαδή το πόσο απέχει το διάνυσμα των συναπτικών τους βαρών από το διάνυσμα εισόδου. Νικητής θεωρείται ο νευρώνας το διάνυσμα των συναπτικών βαρών του οποίου είναι πλησιέστερο προς το διάνυσμα εισόδου. Ο νευρώνας αυτός ενημερώνει κατόπιν τα βάρη του, συμφώνως προς τον ανταγωνιστικό κανόνα εκμάθησης, προσεγγίζοντας έτσι ακόμα περισσότερο το τρέχων διάνυσμα εισόδου. Βασικό στοιχείο του SOFM, και ταυτοχρόνως σημείο διαφοροποίησής του από τον ανταγωνιστικό κανόνα εκμάθησης, είναι ότι μαζί με τον νευρώνα - νικητή, ενημερώνονται και τα διανύσματα των συναπτικών βαρών των γειτονικών του νευρώνων, αναγκάζοντάς τα με αυτόν τον τρόπο να προσεγγίσουν κι αυτά το τρέχων διάνυσμα εισόδου. Ο λόγος για τον οποίο γίνεται αυτό είναι ότι ο αλγόριθμος προσπαθεί να αναγκάσει τους γειτονικούς νευρώνες του δικτύου να αντιδρούν με παρόμοιο τρόπο, ώστε να καταφέρει να κατασκευάσει χάρτες χαρακτηριστικών παρόμοιους με αυτούς που παρατηρούνται στον εγκέφαλο και να καταλήξει επομένως σε μία τοπολογική (δισδιάστατη) κατάταξη των διανυσμάτων εισόδου.

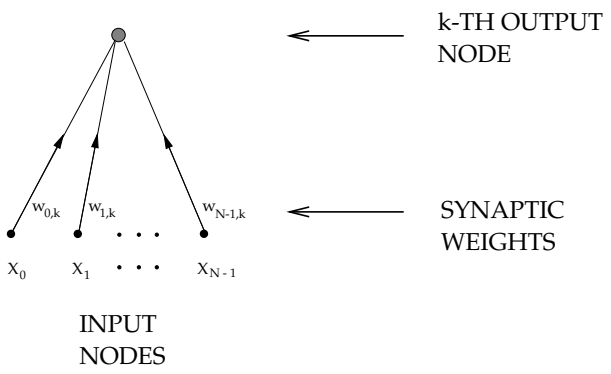
Στην προηγούμενη παράγραφο αναφέρθηκε ότι, μαζί με τον νευρώνα - νικητή, ενημερώνονται και τα βάρη των γειτονικών του νευρώνων. Η έννοια αυτή της **γειτονιάς** (*neighbourhood*) είναι πολύ βασική για τον αλγόριθμο SOFM και για αυτό χρειάζεται να υπάρχει κάποια συνάρτηση, $\mathrm{NE}_j(t)$, η οποία, δοθέντος ενός νευρώνα και της τρέχουσας χρονικής στιγμής, να δίνει ένα σύνολο από νευρώνες που αποτελούν τη γειτονιά του αρχικού νευρώνα τη συγκεκριμένη στιγμή. Η συνάρτηση αυτή θα πρέπει να φθίνει κατά την πάροδο του χρόνου. Αρχικώς δηλαδή, που το δίκτυο δεν έχει αποθηκεύσει ακόμα κάποια γνώση στα βάρη του, η γειτονιά είναι μεγάλη (συνήθως ολόκληρο το δίκτυο) ώστε να επιτρέπει τη γρήγορη προσαρμογή και εκπαίδευση όλων των νευρώνων, ενώ καθώς ο χρόνος περνά και οι νευρώνες αρχίζουν να εξειδικεύονται σε συγκεκριμένα σύνολα διανυσμάτων εισόδου, η γειτονιά μειώνεται ώστε να τους βοηθήσει να εστιαστούν πιο καλά στη συγκεκριμένη περιοχή της εισόδου που έχουν επιλέξει. Μία σχηματική παράσταση της έννοιας της γειτονιάς και του τρόπου με τον οποίο αυτή μεταβάλλεται, φαίνεται στο σχήμα 2.12.

Πρέπει να σημειωθεί εδώ, ότι τόσο το σχήμα της γειτονιάς όσο και ο τρόπος με τον οποίο αυτή μειώνεται κατά την πάροδο του χρόνου είναι στοιχεία του αλγόριθμου που εξαρτούνται από το εκάστοτε πρόβλημα. Αρκετά χρησιμοποιημένες γειτονιές είναι η τετραγωνική, που φαίνεται και στο σχήμα 2.12, και η κυψελοειδής.

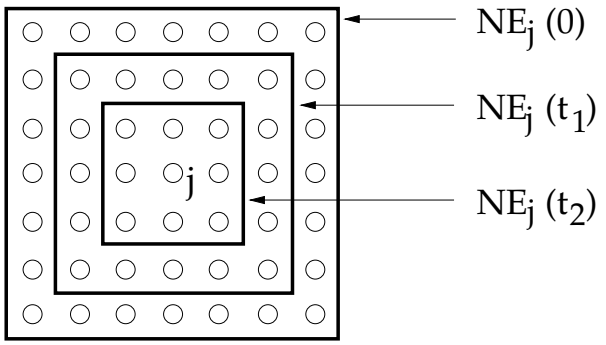
Μία περιγραφή του SOFM σε ψευδοκώδικα φαίνεται στον πίνακα 2.1. Τα βάρη των συνάψεων μεταξύ των κόμβων της εισόδου και αυτών της εξόδου αρχικώς τίθενται ίσα με μικρές τυχαίες τιμές και μετά δίνεται στο δίκτυο το πρώτο διάνυσμα εισόδου. Υπολογίζεται η Ευκλείδια απόσταση αυτού από όλους τους κόμβους εξόδου και επιλέγεται αυτός ο κόμβος εξόδου που έχει την μικρότερη. Κατόπιν, τα βάρη του κόμβου αυτού και των κόμβων που αποτελούν εκείνη τη στιγμή τη γειτονιά του προσαρμόζονται κατά τέτοιο τρόπο ώστε να προσεγγίσουν ακόμα περισσότερο το τρέχων διάνυσμα εισόδου. Η διαδικασία αυτή επαναλαμβάνεται με τα επόμενα διανύσματα εισόδου, έως ότου τα βάρη σταθεροποιηθούν όταν ο **ρυθμός εκμάθησης** (*learning rate*) έχει πλέον μηδενιστεί.



Σχήμα 2.10: Ο αυτο-οργανωνόμενος χάρτης χαρακτηριστικών



Σχήμα 2.11: Τα συναπτικά βάρη του j-οστού νευρώνα



Σχήμα 2.12: Γειτονιές στον SOFM κατά την πάροδο του χρόνου
$\mathrm{NE}_j(t)$ είναι το σύνολο των κόμβων που αποτελούν τη γειτονιά του κόμβου j, τη χρονική στιγμή t. Η γειτονιά αρχικώς έχει μεγάλο εύρος αλλά σταδιακώς συρρικνώνεται. Στο παράδειγμα αυτό, ισχύει $0 < t_1 < t_2$.

Βήμα 1. Αρχικοποίηση των βαρών του δικτύου

Αρχικοποιούνται τα συναπτικά βάρη, $\vec{w}_j, 0 \leq j < M$, των N εισόδων προς τους M κόμβους εξόδου, που φαίνονται στο σχήμα 2.11, με μικρές τυχαίες τιμές.

Έτσι, καλύπτεται η απαίτηση 3a (σελίδα 17) του ανταγωνιστικού κανόνα εκμάθησης για την αρχική διαφοροποίηση των νευρώνων.

Βήμα 2. Παρουσίαση ενός νέου δεδομένου εισόδου, $\vec{x}(t)$, στο δίκτυο

Βήμα 3. Υπολογισμός των αποστάσεων από όλους τους κόμβους του δικτύου

Υπολογίζονται οι αποστάσεις, d_j, μεταξύ του δεδομένου της εισόδου, $\vec{x}(t)$, και του διανύσματος των συναπτικών βαρών, $\vec{w}_j(t)$, κάθε κόμβου εξόδου j χρησιμοποιώντας την εξίσωση

$$d_j = ||\vec{x}(t) - \vec{w}_j(t)||^2$$

όπου $\vec{x}(t)$ είναι το διάνυσμα εισόδου τη χρονική στιγμή t και $\vec{w}_j(t)$ είναι το διάνυσμα των συναπτικών βαρών του j-οστού κόμβου εξόδου την ίδια χρονική στιγμή t. (Η απόσταση d_j δεν είναι άλλη από το τετράγωνο της Ευκλείδιας απόστασης των δύο διανυσμάτων $\vec{x}(t)$ και $\vec{w}_j(t)$).

Βήμα 4. Επιλογή του κόμβου εξόδου με την ελάχιστη απόσταση

Επιλέγεται ο κόμβος j^* ως ο κόμβος εξόδου που έχει την ελάχιστη απόσταση d_j.

Βήμα 5. Προσαρμογή των βαρών των συνάψεων του κόμβου j^* και των γειτόνων του

Προσαρμόζονται τα βάρη των συνάψεων του κόμβου j^* και όλων των κόμβων που ανήκουν στη γειτονιά του, όπως αυτή ορίζεται από τη συνάρτηση $\mathrm{NE}_{j^*}(t)$ που γραφικώς παρουσιάζεται στο σχήμα 2.12. Τα νέα βάρη δίνονται από την εξίσωση

$$\vec{w}_j(t+1) = \vec{w}_j(t) + \eta(t) * (\vec{x}(t) - \vec{w}_j(t))$$

$$\forall j \in \mathrm{NE}_{j^*}(t)$$

Ο όρος $\eta(t)$ ονομάζεται **ρυθμός εκμάθησης** (*learning rate*) $(0 < \eta(t) < 1)$ και μειώνεται κατά την πάροδο του χρόνου. Η παρουσία του στην πιο πάνω εξίσωση έχει σκοπό να επιτρέπει τη γρήγορη προσαρμογή των βαρών προς τα δεδομένα εισόδου στην αρχή, που το δίκτυο βρίσκεται ακόμα μακρυά από την τελική του κατάσταση, βοηθώντας το έτσι να αποκτήσει κάποια αρχική γνώση και με την πάροδο του χρόνου, που το δίκτυο πλησιάζει όλο και περισσότερο την τελική του κατάσταση, να επιτρέπει μόνο μικρές αλλαγές στα βάρη, ώστε το δίκτυο να διατηρεί τη γνώση που έχει αποκτήσει έως εκείνη τη χρονική στιγμή και να κάνει μόνο **μικρές αλλαγές** (*refinements*) σε αυτή.

Βήμα 6. Επιστροφή στο Βήμα 2

Πίνακας 2.1: Ο αλγόριθμος SOFM

Κεφάλαιο 3

Δυναμικώς Αναπροσαρμοζόμενος K-MEANS

Ο K-MEANS [Mac67], [Har75], [VR92] είναι ένας ευρέως διαδεδομένος αλγόριθμος ομαδοποίησης δεδομένων. Η μεγάλη διάδοση και χρήση του οφείλεται στην απλότητά του, στο ότι δύναται να χρησιμοποιηθεί σε πολλά διαφορετικά πεδία εφαρμογών και στο ότι είναι πολύ γρήγορος.

Οι ιδιότητές του αυτές οδήγησαν έως και στην υλοποίησή του με χρήση νευρωνικών δικτύων. Στο κεφάλαιο αυτό πρόκειται να παρουσιαστεί ο κλασσικός αλγόριθμος, ορισμένες παραλλαγές του, το πώς αυτός μπορεί να υλοποιηθεί με ένα νευρωνικό δίκτυο, τα προβλήματα που έχει ένα τέτοιο δίκτυο, ορισμένες προτεινόμενες βελτιστοποιήσεις που μπορούν να γίνουν σε αυτό για να επιδεικνύει καλύτερη συμπεριφορά, καθώς και πώς τελικώς υλοποιήθηκε αυτό στην παρούσα εργασία.

3.1 Ο Κλασσικός K-MEANS

Ο αλγόριθμος K-MEANS σχεδιάστηκε για την ομαδοποίηση δεδομένων σε ομάδες με κοινά μεταξύ τους χαρακτηριστικά. Η μόνη είσοδος που χρειάζεται από το χρήστη, εκτός φυσικά των ίδιων των προς ομαδοποίηση δεδομένων, είναι ο αριθμός K των επιθυμητών ομάδων.

Τις ομάδες που σχηματίζει τις αναπαριστά με το **κέντρο** (*centroid*) τους, που δεν είναι τίποτα άλλο από τον μέσο όρο των δεδομένων που έχουν αποδοθεί στην εκάστοτε ομάδα. Από το γεγονός αυτό πήρε, δε, και το όνομά του, που σε ελεύθερη μετάφραση είναι ‘‘K-Μέσοι όροι’’.

Μία γενική περιγραφή του αλγόριθμου φαίνεται στον πίνακα 3.1. Στον πίνακα αυτόν και στο κείμενο που ακολουθεί, με K συμβολίζεται ο αριθμός των ομάδων στις οποίες θα χωριστούν τα δεδομένα, με S_i το πλήθος των δεδομένων που έχουν αποδοθεί στη i-οστή ομάδα, με $\vec{c}_i$ το κέντρο της i-οστής ομάδας, και με $\vec{x}_{i_j}$ το j-οστό δεδομένο της i-οστής ομάδας. Τα δεδομένα και τα κέντρα των ομάδων γράφονται ως διανύσματα, διότι συνήθως είναι πολυδιάστατες ποσότητες, δηλαδή ανήκουν στο χώρο $\mathbb{R}^n$. Τέλος, με το $||\vec{\alpha} - \vec{\beta}||$ συμβολίζεται η δεύτερη νόρμα των n-διάστατων διανυσμάτων $\vec{\alpha}$ και $\vec{\beta}$, δηλαδή η Ευκλείδια απόστασή τους ($||\vec{\alpha} - \vec{\beta}|| = \sqrt{\sum_{i=1}^{n}(\alpha_i - \beta_i)^2}$).

Το πόσο καλή επιλογή έγινε για τις ομάδες των δεδομένων, μπορεί να φανεί εάν εξεταστεί η διασπορά της κάθε ομάδας. Η διασπορά είναι ο μέσος όρος των αποστάσεων των δεδομένων που αποτελούν την ομάδα από το κέντρο (μέσο όρο) της ομάδας. Ορίζεται, δε, συμφώνως προς την εξίσωση 3.1:

[Βήμα 1]:

Επίλεξε αρχικώς μία τυχαία ομαδοποίηση των δεδομένων σε K ομάδες.

Επανέλαβε

[Βήμα 2]:

Ενημέρωσε τα κέντρα $\vec{c}_i, i = 1, \ldots, K$ των ομάδων, θέτοντάς τα ίσα με τους αντίστοιχους μέσους όρους των ομάδων, δηλαδή:

$$\vec{c}_i = \frac{\sum_{j=1}^{S_i} \vec{x}_{i_j}}{S_i}$$

όπου, S_i είναι το πλήθος των δεδομένων που έχουν αποδοθεί στη i-οστή ομάδα, $\vec{c}_i$ είναι το κέντρο της i-οστής ομάδας, και $\vec{x}_{i_j}$ είναι το j-οστό δεδομένο της i-οστής ομάδας.

[Βήμα 3]:

Για κάθε δεδομένο, $\vec{x}_k$, βρες την ομάδα εκείνη το κέντρο της οποίας είναι πιο κοντά του (συμφώνως προς την Ευκλείδια απόσταση), δηλαδή βρες το δείκτη i^* που ικανοποιεί τη σχέση:

$$d_{i^*} = \min_i(||\vec{x}_k - \vec{c}_i||), i = 1, \ldots, K$$

[Βήμα 4]:

Μετάφερε το δεδομένο $\vec{x}_k$ στην ομάδα υπ' αριθμόν i^*[a].

Έως ότου:

Δεν γίνονται πλέον αλλαγές στα κέντρα των ομάδων.

[a] Ο αλγόριθμος διατηρεί για κάθε ομάδα το σύνολο των δεδομένων που την αποτελούν.

Πίνακας 3.1: Γενική περιγραφή του κλασσικού αλγόριθμου K-MEANS.

$$v_i \stackrel{\text{def}}{=} \frac{\sum_{j=1}^{S_i} ||\vec{x}_{i_j} - \vec{c}_i||^2}{S_i} \tag{3.1}$$

Προφανώς, όσο λιγότερα στοιχεία έχει μία ομάδα, τόσο μικρότερη θα είναι εν γένει η διασπορά της. Όμως όταν πρέπει να διαχωριστούν τα δεδομένα σε ένα σύνολο από ομάδες, τότε η αφαίρεση κάποιων εξ αυτών από μία ομάδα και η μεταφορά τους σε κάποια άλλη μειώνει μεν τη διασπορά της πρώτης ομάδας, αλλά αυξάνει τη διασπορά της δεύτερης.

Για το λόγο αυτό, ο K-MEANS σχεδιάστηκε ώστε η συνάρτηση που προσπαθεί να ελαχιστοποιήσει να είναι το **μέσο τετραγωνικό σφάλμα** (*mean square error (MSE)*), δηλαδή το άθροισμα των διασπορών όλων των ομάδων, όπως φαίνεται και στην εξίσωση 3.2.

$$\text{MSE}(K) = \sum_{i=1}^{K} v_i \tag{3.2}$$

Κατ' αυτόν τον τρόπο, η μεταφορά ενός δεδομένου από μία ομάδα σε κάποια άλλη

επιτρέπεται μόνον εάν η μείωση της διασποράς της πρώτης ομάδας είναι μεγαλύτερη από την αύξηση της διασποράς της δεύτερης ομάδας.

Το γεγονός αυτό έχει και ένα άλλο αξιοσημείωτο στοιχείο. Χάρη σε αυτόν τον τρόπο λειτουργίας του αλγόριθμου, είναι εύκολο να αποδειχθεί ότι αυτός συγκλίνει πάντοτε, δίχως ποτέ να εισέρχεται σε **ατέρμονα βρόχο** (*endless loop*), καθώς μετακινεί δεδομένα από μία ομάδα σε μίαν άλλη (οπότε και συνεχίζει να εκτελείται), εάν και μόνον εάν, η μετακίνηση αυτή θα οδηγήσει σε μείωση του μέσου τετραγωνικού σφάλματος. Καθώς, δε, το μέσο τετραγωνικό σφάλμα είναι μία μη αρνητική ποσότητα, είναι προφανές ότι δεν μπορεί να μειώνεται επ' άπειρον.

Αυτό του το πλεονέκτημα αποτελεί όμως ταυτοχρόνως και το μεγάλο του μειονέκτημα, καθώς ο K-MEANS, κατά την αναζήτησή του για το βέλτιστο διαχωρισμό των δεδομένων σε ομάδες, είναι δυνατόν να εγκλωβιστεί σε κάποιο τοπικό ελάχιστο του **χώρου αναζήτησης** (*search space*). Έτσι, όταν ο K-MEANS φθάσει σε ένα τέτοιο τοπικό ελάχιστο, τότε θα παραμείνει σε αυτό, καθώς για να ξεφύγει θα πρέπει να μετακινήσει κάποια δεδομένα σε ομάδες που είναι τοπικά υποβέλτιστες. Κάτι τέτοιο όμως, όπως δείχθηκε, δεν γίνεται ποτέ. Προκειμένου να ξεπεραστεί αυτό του το μειονέκτημα, συνήθως οι ερευνητές τρέχουν τον αλγόριθμο αρκετές φορές με τα ίδια δεδομένα, επιλέγοντας κάθε φορά διαφορετικές αρχικές τυχαίες ομαδοποιήσεις και επιλέγοντας στο τέλος εκείνη την ομαδοποίηση που έχει το μικρότερο μέσο τετραγωνικό σφάλμα.

3.1.1 Παραλλαγές του Κλασσικού K-MEANS

Υπάρχουν διάφορες παραλλαγές του κλασσικού αλγόριθμου K-MEANS, που αξίζει να αναφερθούν. Τα σημεία του αλγόριθμου που μπορούν να αλλαχθούν είναι τα εξής δύο [Har75]:

1. Η επιλογή των αρχικών ομάδων (βήμα 1 στον αλγόριθμο του πίνακα 3.1).
2. Ο κανόνας ενημέρωσης των κέντρων των ομάδων κατά τη μετακίνηση δεδομένων (βήμα 2 στον αλγόριθμο του πίνακα 3.1).

Όσον αφορά την επιλογή των αρχικών ομάδων, υπάρχουν τρεις περιπτώσεις:

1. Οι αρχικές ομάδες μπορούν να επιλεχθούν τυχαίως. Τότε ο αλγόριθμος επαναλαμβάνεται με διαφορετικές αρχικές ομαδοποιήσεις προκειμένου να εξαχθεί κάποιο συμπέρασμα για τη θέση του ολικού βελτίστου από τη διάταξη στο χώρο των τοπικών ελαχίστων που φθάνει κάθε φορά ο αλγόριθμος. Προκειμένου να χρησιμοποιηθεί αυτή η τεχνική, θα πρέπει να υπάρχει κάποια θεωρία για τη κατανομή που συνδέει τα τοπικά και ολικά βέλτιστα σημεία του χώρου αναζήτησης.
2. Μπορεί να επιλεχθεί μία συγκεκριμένη μεταβλητή των δεδομένων, να χωρισθεί αυτή σε K ίσα διαστήματα και να χωρισθούν τα δεδομένα σε K ομάδες με βάση το διάστημα που ανήκει η αντίστοιχη μεταβλητή του κάθε δεδομένου.
3. Οι αρχικές K ομάδες μπορούν να είναι ίδιες με αυτές που είχε καταλήξει ο αλγόριθμος για $K-1$ ομάδες, με τη διαφορά ότι το δεδομένο εκείνο που απείχε περισσότερο από το κέντρο της ομάδας του, τώρα θα αποτελεί ξεχωριστή ομάδα.

Αντιστοίχως, για τον κανόνα ενημέρωσης υπάρχουν οι εξής δύο περιπτώσεις:

1. Τα κέντρα των ομάδων μπορούν να επαναϋπολογίζονται μετά από κάθε μετακίνηση δεδομένου από μία ομάδα σε μία άλλη.

2. Τα κέντρα των ομάδων μπορούν να επαναϋπολογίζονται στο τέλος όλων των μετακινήσεων δεδομένων από μία ομάδα σε μία άλλη.

Θα πρέπει να σημειωθεί πως ο κλασσικός αλγόριθμος K-MEANS που έχει υλοποιηθεί στα πλαίσια του CLUE [Lab95], κάνει τυχαία επιλογή των αρχικών ομάδων και ενημερώνει τα κέντρα των ομάδων στο τέλος όλων των μετακινήσεων προκειμένου να επιταχυνθεί η λειτουργία του.

3.2 Υλοποίηση του K-MEANS με χρήση Νευρωνικών Δικτύων

Η μεγάλη δημοτικότητα του K-MEANS, οδήγησε και στην μετατροπή του προκειμένου να μπορεί να υλοποιηθεί με ένα νευρωνικό δίκτυο [Llo82],[MD89], [HL88]. Αυτό έγινε προκειμένου να μπορεί να χρησιμοποιηθεί ο K-MEANS και σε ακολουθίες από δεδομένα, δηλαδή και στις περιπτώσεις εκείνες όπου δεν υπάρχουν όλα τα δεδομένα διαθέσιμα εξαρχής όπως απαιτεί ο κλασσικός K-MEANS, αλλά αυτά γίνονται διαθέσιμα κατά την πάροδο του χρόνου.

Το νευρωνικό δίκτυο που υλοποιεί τον K-MEANS, το οποίο και ονομάζεται αναπροσαρμοζόμενος K-MEANS προκειμένου να διαχωρίζεται από τον κλασσικό K-MEANS, έχει K τον αριθμό νευρώνες, οι οποίοι αντιστοιχούν στις K ζητούμενες ομάδες των δεδομένων. Δεδομένου ότι σκοπός του K-MEANS είναι να δώσει τα κέντρα των K ομάδων, αυτά αποθηκεύονται στα διανύσματα των βαρών των συνάψεων των νευρώνων.

Η υλοποίηση του K-MEANS με ένα τεχνητό νευρωνικό δίκτυο παρουσιάζεται στον πίνακα 3.2. Ο δείκτης εγκλεισμού που αναφέρεται στον πίνακα αυτό, δεν είναι τίποτα άλλο από μία ‘‘μαθηματικοποιημένη’’ έκφραση του ‘‘επίλεξε την ομάδα, το κέντρο της οποίας είναι πλησιέστερο στο τρέχον δεδομένο εισόδου’’. Πράγματι, όπως φαίνεται από τις εξισώσεις 3.3 και 3.4, για όλους τους νευρώνες πλην του i-οστού, το $M_j(\vec{x})$ είναι ίσο με μηδέν, οπότε τα βάρη τους παραμένουν σταθερά. Όμως, το $M_i(\vec{x})$ είναι ίσο με τη μονάδα, οπότε το διάνυσμα βαρών του i-οστού νευρώνα θα μεταβληθεί κατά την ποσότητα $\eta * [\vec{x}(T) - \vec{c}_i(T)]$.

Ξεχνώντας προσωρινώς το συντελεστή η, η ποσότητα που προστίθεται στο διάνυσμα βαρών δεν είναι τίποτα άλλο, παρά η διαφορά του δεδομένου εισόδου από το τρέχον διάνυσμα βαρών. Ουσιαστικώς δηλαδή, μεταβάλλεται το διάνυσμα βαρών προς την κατεύθυνση του δεδομένου εισόδου, προσπαθώντας να το προσεγγίσει.

Επιστρέφοντας τώρα στο συντελεστή η, θα πρέπει να σημειωθεί ότι στην αρχική υλοποίηση του K-MEANS με ένα νευρωνικό δίκτυο, αυτός ήταν σταθερός καθ’ όλη την εκπαίδευση του δικτύου και περιοριζόταν στο διάστημα $[0, 1]$. Εξετάζοντας τις ακραίες τιμές του διαστήματος στο οποίο λαμβάνει τιμές το η, παρατηρείται ότι εάν είναι ίσο με μηδέν, τότε τα βάρη του δικτύου θα παραμένουν πάντοτε σταθερά, οπότε και δεν θα μαθαίνει το δίκτυο τίποτα από τα δεδομένα που του παρουσιάζονται. Αντιθέτως, εάν είναι ίσο με τη μονάδα, τότε τα βάρη θα τίθενται ίσα με το εκάστοτε δεδομένο εισόδου, ξεχνώντας ουσιαστικώς όλα τα προηγούμενα δεδομένα που είχαν παρουσιαστεί. Επομένως, ο συντελεστής η καθορίζει το ποσοστό με το οποίο λαμβάνεται υπόψη το τρέχον δεδομένο εισόδου σε σχέση με τα προηγούμενά του προκειμένου να υπολογισθεί η τρέχουσα τιμή των βαρών. Χρησιμεύει, δηλαδή, για να υλοποιηθεί ένα είδος μνήμης, όπως και ο αντίστοιχος συντελεστής $\eta(t)$ που αναφερόταν στην περιγραφή του αλγόριθμου SOFM στον πίνακα 2.1 του υποκεφαλαίου 2.5.1.

Η σχέση με το υποκεφάλαιο 2.5.1 δεν περιορίζεται μόνο στην παρουσία του συντελεστή αυτού της μνήμης. Εάν παραβληθούν οι δύο αλγόριθμοι (πίνακας 2.1 και πίνακας 3.2) τότε είναι φανερό ότι ο αναπροσαρμοζόμενος K-MEANS αποτελεί μία υποπερίπτωση του SOFM

Έστω K νευρώνες και έστω $\vec{c}_1, \ldots, \vec{c}_K$ τα διανύσματα των συναπτικών τους βαρών, τα οποία τελικώς θα περιέχουν τα διανύσματα περιγραφής των K ομάδων (δηλαδή τα κέντρα τους).
Αν $\vec{x}(T)$ και $\vec{c}_i(T)$ είναι τα διανύσματα εισόδου (δεδομένο) και το διάνυσμα των συναπτικών βαρών του i-οστού νευρώνα τη χρονική στιγμή T, και $d(\vec{x}, \vec{c}_i) = ||\vec{x} - \vec{c}_i||^2$ είναι το τετράγωνο της Ευκλείδιας απόστασης των δύο, τότε το νευρωνικό δίκτυο επαναπροσδιορίζει τις τιμές των βαρών συμφώνως προς την εξίσωση 3.3.

$$\vec{c}_i(T+1) = \vec{c}_i(T) + M_i(\vec{x}(T)) * \eta * [\vec{x}(T) - \vec{c}_i(T)] \tag{3.3}$$

όπου η είναι ο **ρυθμός εκμάθησης** (*learning rate*) του δικτύου, που καθορίζει την ταχύτητα και ακρίβεια της προσέγγισης των δεδομένων, και το $M_i(\cdot)$ είναι ένας **δείκτης εγκλεισμού** (*membership indicator*) που καθορίζει το κατά πόσον το δεδομένο εισόδου $\vec{x}(T)$ ανήκει στην i-οστή ομάδα.
Στο δίκτυο αυτό, ο ρυθμός εκμάθησης, η, ορίζεται να είναι μία σταθερά και ο δείκτης εγκλεισμού M_i ορίζεται συμφώνως προς την εξίσωση 3.4:

$$M_i(\vec{x}) = \begin{cases} 1 & \text{εάν } d(\vec{x}, \vec{c}_i) \leq d(\vec{x}, \vec{c}_j) \ \forall j \neq i \\ 0 & \text{αλλιώς} \end{cases} \tag{3.4}$$

Πίνακας 3.2: Υλοποίηση του K-MEANS με ένα τεχνητό νευρωνικό δίκτυο.

αλγόριθμου, στην οποία η γειτονιά ενός νευρώνα είναι πάντοτε σταθερή και ίση με τον ίδιο τον νευρώνα.

Ο λόγος που χρησιμοποιείται ο αναπροσαρμοζόμενος K-MEANS αντί του SOFM, είναι ότι παρόλο που ο δεύτερος έχει μεγαλύτερη ευελιξία, εντούτοις μετά την εφαρμογή του δεν είναι γνωστό ποιοί νευρώνες έχουν εκπαιδευτεί να αναγνωρίζουν στοιχεία της ίδιας ομάδας.

Έτσι, μετά την εφαρμογή του SOFM πρέπει να εφαρμοσθεί ένας ακόμα αλγόριθμος ο λεγόμενος **LVQ** (*Learning Vector Quantization*) [Koh90], που διαχωρίζει τους νευρώνες, πλέον, ανά ομάδες. Ο αλγόριθμος αυτός όμως είναι αλγόριθμος εκμάθησης με διδάσκαλο, απαιτεί δηλαδή την ύπαρξη δεδομένων για τα οποία είναι ήδη γνωστές οι ομάδες στις οποίες ανήκουν αυτά, ώστε να δοκιμάσει, ουσιαστικώς, τους νευρώνες του SOFM και να δει ποιοί εξ αυτών αντιδρούν σε συγκεκριμένες ομάδες. Εάν δεν υπάρχουν τέτοιου είδους προ-ομαδοποιημένα δεδομένα, τότε είναι αδύνατο να ερμηνευθεί ο χάρτης χαρακτηριστικών τον οποίο έχει κατασκευάσει ο SOFM, προκειμένου να εξαχθούν οι περιγραφές των ομάδων.

3.3 Προβλήματα του Αναπροσαρμοζόμενου K-MEANS

Ο αναπροσαρμοζόμενος K-MEANS, όπως και ο κλασσικός K-MEANS, έχει το πρόβλημα ότι μπορεί να παγιδευθεί σε κάποιο τοπικό ελάχιστο, που να απέχει αρκετά από το ολικό βέλτιστο. Αυτό οφείλεται στο γεγονός ότι, κατά την αρχικοποίηση, κάποια από τα διανύσματα βαρών μπορεί να πάρουν τιμές που να βρίσκονται σε περιοχές του χώρου αναζήτησης οι οποίες περιέχουν λίγα ή και καθόλου στοιχεία. Έτσι, κατά την εκπαίδευση του δικτύου, οι νευρώνες αυτοί δεν πρόκειται ποτέ να βρεθούν κοντά σε κάποιο από τα δεδομένα εισόδου και επομένως, τα διανύσματα βαρών τους θα διατηρήσουν τις αρχικές τυχαίες τιμές τους. Ένας κλασσικός τρόπος υπέρβασης της υποχρησιμοποίησης αυτής είναι να χρησιμοποιηθεί η λεγόμενη **διαρρέουσα εκμάθηση** (*leaky learning*) [RZ86] όπου, σε συνδυασμό με την ενημέρωση του πλησιέστερου προς το δεδομένο εισόδου νευρώνα κάθε

φορά, ενημερώνονται και οι υπόλοιποι νευρώνες αλλά με μικρότερους ρυθμούς εκμάθησης.

Μίαν άλλη προσέγγιση είναι η χρήση του **κανόνα εκμάθησης με τύψεις** (*conscience learning law*) [Des88], όπου ο καθορισμός του πλησιέστερου νευρώνα προς το τρέχον διάνυσμα εισόδου χρησιμοποιεί μία νόρμα που ενισχύει τους νευρώνες που είχαν απαντήσει σε λιγότερα δεδομένα εισόδου στο παρελθόν. Το όνομα του κανόνα προκύπτει από το ότι οι νευρώνες που έχουν απαντήσει στα περισσότερα δεδομένα νοιώθουν ''τύψεις'' για αυτό και παραχωρούν τη σειρά τους στους υπόλοιπους.

Όμως, όπως συχνά συμβαίνει, η θεραπεία είναι χειρότερη από την ασθένεια. Κι αυτό γιατί οι δύο μέθοδοι αυτές οδηγούν σε ομαδοποιήσεις που δεν είναι βέλτιστες (μερικές φορές, δε, δεν πλησιάζουν καν στην βέλτιστη) συμφώνως προς τη συνάρτηση κόστους που ορίζεται από το μέσο τετραγωνικό σφάλμα κι αυτό διότι η χρήση τους έχει ως αποτέλεσμα την παραλλαγή της συνάρτησης κόστους που προσπαθεί να ελαχιστοποιήσει το δίκτυο. Εκτός αυτού, η διαρρέουσα εκμάθηση αυξάνει τους συνολικούς υπολογισμούς που πρέπει να επιτελεί το δίκτυο για κάθε δεδομένο εισόδου, αφού τώρα πλέον πρέπει να ενημερώνονται όλα τα διανύσματα βαρών.

Ένα άλλο πρόβλημα του αναπροσαρμοζόμενου K-MEANS έχει να κάνει με το ρυθμό εκμάθησης η που χρησιμοποιείται σε αυτόν. Στην επιλογή του, αναγκαστικώς θα υπάρχει μία **αντιστάθμιση** (*tradeoff*) μεταξύ της επίδρασής του στην **δυναμική επίδοση** (*dynamic performance*) (δηλαδή το ρυθμό σύγκλισης) και την **επίδοση στη σταθερή κατάσταση** (*steady-state performance*) (δηλαδή την τελική απόκλιση από την βέλτιστη λύση, που παρατηρείται στο τέλος της εκμάθησης) του δικτύου. Δηλαδή, όταν χρησιμοποιείται ένας σταθερός ρυθμός εκμάθησης, αυτός θα πρέπει να είναι αρκετά μικρός προκειμένου να συγκλίνει η διαδικασία της εκμάθησης. Όσο μικρότερος είναι ο ρυθμός εκμάθησης, τόσο μικρότερη είναι η τελική απόκλιση από τη βέλτιστη λύση, αλλά ταυτοχρόνως τόσο μικρότερος θα είναι και ο ρυθμός με τον οποίο επιτελείται η εκμάθηση. Καθώς ο βέλτιστος ρυθμός εκμάθησης εξαρτάται από τις ιδιότητες του συγκεκριμένου προβλήματος κάθε φορά, δεν είναι δυνατή η επιλογή του εξ αρχής, αλλά το σύνηθες είναι να επιλέγεται αρχικώς ένας σχετικώς μικρός ρυθμός και κατόπιν να βελτιώνεται αυτός με διαδοχικούς πειραματισμούς.

Εξ αιτίας της δυσκολίας της λύσης αυτής, έχουν μελετηθεί διάφορες τεχνικές για την εφαρμογή ενός μεταβαλλόμενου ρυθμού εκμάθησης. Στο [DM90a], προτάθηκε για πρώτη φορά ο ρυθμός εκμάθησης του κάθε νευρώνα να είναι αντιστρόφως ανάλογος προς την τετραγωνική ρίζα του αριθμού των δεδομένων που έχουν αποδοθεί σε αυτόν, θεωρώντας ότι όσα περισσότερα δεδομένα έχουν αποδοθεί σε έναν νευρώνα, τόσο περισσότερο έχει μάθει αυτός. Καθώς η σύγκλιση της μεθόδου αυτής ήταν πολύ αργή, προτάθηκε από τους ίδιους μία μέθοδος **αναζήτησης-σύγκλισης** (*search-then-converge*), όπου $\eta(t) = \frac{\eta_0}{(1+t/\tau)}$ [DM90b]. Με τη μέθοδο αυτή, ο ρυθμός εκμάθησης μένει κοντά στην τιμή η_0 για ένα χρονικό διάστημα τ και κατόπιν μειώνεται με ρυθμό $\frac{1}{t}$. Για πολλά προβλήματα, αυτή η μέθοδος συγκλίνει με ακρίβεια και σε μικρό χρονικό διάστημα. Όμως, δεν είναι δυνατόν να επιλέγουμε εκ προοιμίου την βέλτιστη τιμή του χρονικού διαστήματος τ για κάθε πρόβλημα. Επίσης, τέτοιου είδους μέθοδοι, με προκαθορισμένους ρυθμούς εκμάθησης, δεν είναι αρκετά ευέλικτες προκειμένου να χρησιμοποιούνται σε προβλήματα, τα χαρακτηριστικά των οποίων μεταβάλλονται κατά τη διάρκεια του χρόνου.

3.4 Βέλτιστος Αναπροσαρμοζόμενος K-MEANS

Στην εργασία [CS95] οι συγγραφείς παραθέτουν τα αποτελέσματα της προσπάθειάς τους να επιλύσουν τα δύο αυτά προβλήματα του αναπροσαρμοζόμενου K-MEANS. Η εργασία τους είχε, σε μεγάλο βαθμό, ως έναυσμα την εργασία [Ger79]. Στην τελευταία αποδεικνύεται

ότι για μία **λεία** (*smooth*) κατανομή P των δεδομένων και για μεγάλο αριθμό ομάδων K, η βέλτιστη ομαδοποίηση συμφώνως προς τη συνάρτηση MSE, θα δώσει ομάδες με ίδιες διασπορές v_i. Θεώρησαν ότι θα είχε αξία να προσπαθήσει κανείς να καταλήξει σε ομάδες με ίδιες διασπορές, ακόμα κι αν το K δεν είναι αρκετά μεγάλο ή η P δεν είναι λεία. Ως εκ τούτου, διαμόρφωσαν τον αναπροσαρμοζόμενο K-MEANS σε δύο σημεία του, τα οποία περιγράφονται ακολούθως.

3.4.1 Βέλτιστη χρήση των νευρώνων του δικτύου

Προκειμένου να επιλύσουν το πρόβλημα της παγίδευσης των νευρώνων σε περιοχές με λίγα ή και καθόλου δεδομένα, που όπως δείχθηκε στο υποκεφάλαιο 3.3 οδηγεί σε ομαδοποιήσεις που απέχουν πολύ από τη βέλτιστη, προτείνουν την αντικατάσταση της Ευκλείδιας απόστασης από μία συνάρτηση απόστασης που λαμβάνει υπόψη της και τη διασπορά των στοιχείων των ομάδων. Η συνάρτηση αυτή παρουσιάζεται στην εξίσωση 3.5.

$$d_{bias}(\vec{x}, \vec{c}_i) = v_i * ||\vec{x} - \vec{c}_i||^2 \tag{3.5}$$

Στην 3.5, η ποσότητα v_i είναι η διασπορά της i-οστής ομάδας, δηλαδή ο μέσος όρος των αποστάσεων των στοιχείων της ομάδας αυτής από το κέντρο της (εξίσωση 3.1).

Καθώς όσο αυξάνεται το μέγεθος μίας ομάδας εν γένει αυξάνει και η διασπορά της, η χρήση της απόστασης αυτής έχει ως αποτέλεσμα την προτίμηση των ομάδων εκείνων που έχουν λίγα στοιχεία εν σχέσει με τις υπόλοιπες. Έτσι, αναγκάζουν όλους τους νευρώνες να συμμετάσχουν στην ομαδοποίηση, βοηθώντας όσους έχουν εγκλωβιστεί σε κάποια στείρα περιοχή του χώρου αναζήτησης να απομακρυνθούν από αυτή.

Καθώς το νευρωνικό δίκτυο δεν διατηρεί κάπου τα στοιχεία που αποτελούν κάθε ομάδα, δεν είναι δυνατόν να υπολογίζεται η ακριβής τιμή της διασποράς της κάθε ομάδας. Έτσι, αυτή προσεγγίζεται συμφώνως προς την εξίσωση 3.6:

$$v_i(T+1) = \alpha * v_i(T) + (1-\alpha) * M_i(\vec{x}(T)) * ||\vec{x}(T) - \vec{c}_i(T)||^2 \tag{3.6}$$

Ο συντελεστής α της εξίσωσης 3.6 είναι μία σταθερά κατά τι μικρότερη της μονάδας. Όσο πιο κοντά είναι το α στη μονάδα, τόσο πιο ακριβής είναι η προσέγγιση της v_i, αλλά τόσο πιο αργά θα γίνεται η προσέγγιση αυτή. Οι συγγραφείς της εργασίας [CS95] στα πειράματά τους θέτουν το α ίσο με 0.9999.

Ο προσεκτικός αναγνώστης θα παρατηρήσει ότι με την αλλαγή της συνάρτησης απόστασης που προτείνεται μεταβάλλεται αναγκαστικώς και η συνάρτηση που τελικώς θα βελτιστοποιήσει το νευρωνικό δίκτυο.

Πράγματι, η συνάρτηση που ελαχιστοποιεί τώρα το δίκτυο είναι αυτή που ορίζεται στην εξίσωση 3.7. Αν ληφθεί υπόψη και ο ορισμός της διασποράς, τότε παίρνει τη μορφή της εξίσωσης 3.8:[1]

$$\text{VWMSE}(K) = \sum_{i=1}^{K} v_i * \frac{\sum_{j=1}^{S_i} ||\vec{x}_{i_j} - \vec{c}_i||^2}{S_i} \tag{3.7}$$

$$\text{VWMSE}(K) = \sum_{i=1}^{K} v_i^2 \tag{3.8}$$

Πλην όμως, οι συγγραφείς απέδειξαν ότι εάν ο αριθμός K των ομάδων είναι αρκετά μεγάλος και η κατανομή των προς ομαδοποίηση δεδομένων είναι **λεία** (*smooth*), τότε η

[1] Το όνομα VWMSE προκύπτει από το ''variance-weighted mean square error''.

συνάρτηση που ελαχιστοποιεί το δίκτυο (εξίσωση 3.8), είναι ισοδύναμη με την αρχική (εξίσωση 3.2), δηλαδή με το μέσο τετραγωνικό σφάλμα. Για διευκόλυνση, η απόδειξη παρατίθεται και στο παράρτημα Α. Η παραδοχή που έκαναν είναι ότι και στις περιπτώσεις εκείνες που δεν ισχύει κάποια από τις δύο αυτές συνθήκες, η βέλτιστη ομαδοποίηση ως προς τη συνάρτηση VMSE δεν θα απέχει πολύ από την MSE.

3.4.2 Δυναμική Ρύθμιση του Ρυθμού Εκμάθησης

Όσον αφορά το πρόβλημα της επιλογής του ρυθμού εκμάθησης, παρατήρησαν ότι η βέλτιστη λύση θα πρέπει να επιτρέπει τη δυναμική **ρύθμισή** (*adjustment*) του, αναλόγως με το πόσο απέχει το δίκτυο από τη βέλτιστη λύση.

Συγκεκριμένα, όταν το δίκτυο απέχει πολύ από τη βέλτιστη λύση, τότε ο ρυθμός εκμάθησης θα πρέπει να αυξάνεται ώστε να βελτιωθεί γρήγορα η τρέχουσα ομαδοποίηση, ενώ όταν είναι κοντά στη βέλτιστη λύση, τότε θα πρέπει να ελαττώνεται ώστε να μπορέσει να προσεγγισθεί η βέλτιστη ομαδοποίηση με μεγαλύτερη ακρίβεια.

Για να επιτευχθεί κάτι τέτοιο θα πρέπει πρώτα να ορισθεί η ποιότητα της εκάστοτε ομαδοποίησης, στην οποία έχει καταλήξει το δίκτυο, ώστε να μπορεί να μετρηθεί και η απόστασή της από τη βέλτιστη ομαδοποίηση.

Δεδομένου ότι στη βέλτιστη ομαδοποίηση οι διασπορές των ομάδων v_i είναι ίσες, βάσισαν το μέτρο της ποιότητας της τρέχουσας ομαδοποίησης στο πόσο όμοιες είναι οι διασπορές μεταξύ τους.

Καθώς από τη Θεωρία Πληροφορίας γνώριζαν ότι η **εντροπία** (*entropy*) ενός συστήματος με K καταστάσεις ελαχιστοποιείται όταν οι πιθανότητες όλων των καταστάσεων είναι ίδιες μεταξύ τους και ίσες με $\frac{1}{K}$, κανονικοποίησαν τις διασπορές συμφώνως προς την εξίσωση 3.9, ώστε να έχουν αυτές τιμές στο διάστημα $[0, 1]$, όπως και οι πιθανότητες πάνω στις οποίες εφαρμόζεται η συνάρτηση της εντροπίας, και κατόπιν εφήρμοσαν τη συνάρτηση εντροπίας σε αυτές, όπως φαίνεται στην εξίσωση 3.10:

$$v_{i,norm} = \frac{v_i}{\sum_{j=1}^{K} v_j} \tag{3.9}$$

$$H(v_1, v_2, \ldots, v_K) = \sum_{i=1}^{K} -v_{i,norm} * \ln(v_{i,norm}) \tag{3.10}$$

Η συνάρτηση 3.10 μεγιστοποιείται, όπως ήδη αναφέρθηκε, όταν $v_i = v_j \ \forall i, j \Rightarrow v_{i,norm} = \frac{1}{K} \ \forall i$, οπότε και η τιμή της γίνεται ίση με $\ln(\mathrm{K})$.

Έτσι, ο ρυθμός εκμάθησης μπορεί να ορισθεί όπως στην εξίσωση 3.11. Ο αριθμητής του κλάσματος αποδίδει την απόσταση που έχει η ποιότητα της τρέχουσας ομαδοποίησης από αυτή της βέλτιστης και ο παρονομαστής χρησιμεύει στο να κανονικοποιήσει την τιμή του ρυθμού εκμάθησης, ώστε αυτή να βρίσκεται στο διάστημα $[0, 1]$.

$$\eta = \frac{\ln(K) - H(v_1, v_2, \ldots, v_K)}{\ln(K)} \tag{3.11}$$

3.5 Ο αναπροσαρμοζόμενος K-MEANS στο CLUE

Κατά τη διεξαγωγή κάποιων αρχικών πειραμάτων με τεχνητώς κατασκευασμένα δεδομένα (τα δεδομένα που χρησιμοποιήθησαν περιγράφονται στο υποκεφάλαιο 5.1) προκειμένου να αξιολογηθεί η επίδοση του δικτύου, παρατηρήθηκε μία ανεπιθύμητη συμπεριφορά. Δηλαδή, το δίκτυο τα ομαδοποιούσε έτσι ώστε κάθε ομάδα να έχει παρόμοια διασπορά, μετακινώντας

δεδομένα σε διαφορετική ομάδα από αυτήν που πραγματικά ανήκαν, όταν η τελευταία ‘‘μεγάλωνε’’ αισθητά.

Το φαινόμενο αυτό της εσφαλμένης ομαδοποίησης εμφανιζόταν ακόμα και όταν οι ομάδες είχαν το ίδιο περίπου πλήθος στοιχείων, εξαρτόταν, δε, από τη σειρά με την οποία παρουσιάζονταν τα δεδομένα στο νευρωνικό δίκτυο, καθώς και από την κατανομή των τιμών που είχαν τα δεδομένα.

Οι παρατηρήσεις αυτές, της εξάρτησης της συνάρτησης απόστασης από το μέγεθος των διαφόρων ομάδων καθώς και από τη σειρά παρουσίασης των δεδομένων στο νευρωνικό δίκτυο, οδήγησαν στο συμπέρασμα ότι η προτεινόμενη συνάρτηση απόστασης δεν αποδίδει τόσο καλά με τα δεδομένα που χρησιμοποιούνται στην παρούσα εργασία, όσο με εκείνα που είχαν χρησιμοποιηθεί στην [CS95]. Ως εκ τούτου, εκτός από το δίκτυο που περιγράφηκε ανωτέρω, το οποίο θα αναφέρεται στο εξής ως NN-VWMSE, υλοποιήθηκε στο CLUE και ένα παραπλήσιο νευρωνικό δίκτυο το οποίο χρησιμοποιεί την κλασσική Ευκλείδια απόσταση αντί της 3.5, διατηρεί όμως τη μέθοδο της δυναμικής ρύθμισης του ρυθμού εκμάθησης. Το δίκτυο αυτό επέδειξε καλύτερη συμπεριφορά στα ίδια τεχνητά δεδομένα. Το δεύτερο αυτό δίκτυο θα αναφέρεται στο εξής ως NN-MSE.

Τα αποτελέσματα των πειραμάτων βρίσκονται στο τμήμα 5.2 του παραρτήματος Δ για τον αναπροσαρμοζόμενο K-MEANS που ελαχιστοποιεί τη συνάρτηση VWMSE και στο τμήμα 5.3 του ιδίου παραρτήματος για τον αναπροσαρμοζόμενο K-MEANS που ελαχιστοποιεί τη συνάρτηση MSE.

Κεφάλαιο 4

Γραφοθεωρητική Μέθοδος Ομαδοποίησης ‘‘Εν Πτήσει’’

Στα πλαίσια της εργασίας αυτής σχεδιάστηκε και υλοποιήθηκε και άλλη μία μέθοδος ομαδοποίησης ‘‘εν πτήσει’’, η οποία αποτελεί προέκταση μίας γραφοθεωρητικής μεθόδου ομαδοποίησης **ομαδικής επεξεργασίας** (*batch processing*).

Στο παρόν κεφάλαιο πρόκειται να δοθούν τα γενικά στοιχεία των γραφοθεωρητικών μεθόδων ομαδοποίησης, μία περιγραφή της αρχικής μεθόδου ομαδικής επεξεργασίας, καθώς και πώς αυτή προεκτάθηκε προκειμένου να μπορεί να κάνει ομαδοποίηση δεδομένων ‘‘εν πτήσει’’.

4.1 Γενικά Στοιχεία Γραφοθεωρητικών Μεθόδων Ομαδοποίησης

Οι γραφοθεωρητικές μέθοδοι ομαδοποίησης προσεγγίζουν το πρόβλημα της εύρεσης των ομάδων, στις οποίες ανήκουν τα δεδομένα, θεωρώντας κάθε ένα εξ αυτών ως κόμβο ενός πλήρως συνδεδεμένου γράφου. Στις ακμές του γράφου αποδίδονται ως βάρη οι Ευκλείδιες αποστάσεις των δεδομένων που συνδέουν αυτές.

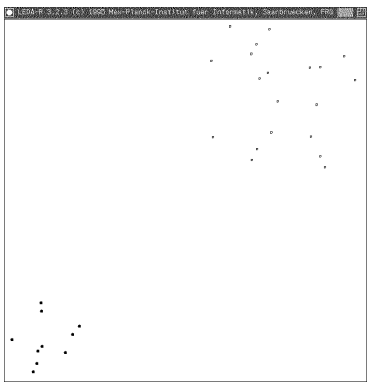
Στην εργασία [Zah71] περιγράφεται ένα σύνολο διαφορετικών μεθόδων που μπορούν να χρησιμοποιηθούν προκειμένου από τον προαναφερθέντα γράφο να αποκτηθούν οι ζητούμενες ομάδες των δεδομένων.

Οι περιγραφόμενες μέθοδοι προέκυψαν από την προσπάθεια προσομοίωσης του τρόπου με τον οποίο ομαδοποιεί το ανθρώπινο μάτι ένα σύνολο από σημεία στο δισδιάστατο χώρο. Παρόλα ταύτα, η χρήση τους δεν περιορίζεται μόνο σε δισδιάστατους χώρους αλλά δύνανται να χρησιμοποιηθούν και σε χώρους μεγαλύτερης διάστασης.

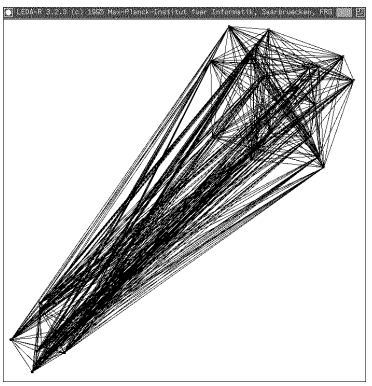
Ένα βασικό πλεονέκτημά τους είναι η εύκολη περιγραφή και επεξήγηση από έναν άνθρωπο των ομαδοποιήσεων που παράγουν, καθώς ακολουθούν τις ίδιες αρχές με τις οποίες οργανώνουν οι ανθρώπινες αισθήσεις σύνολα δισδιάστατων σημείων. Δηλαδή, τείνουν να ενώνουν τα δεδομένα σε ομάδες βασιζόμενες στην εγγύτητα των δεδομένων καθώς και στην πυκνότητά τους.

Οι μέθοδοι αυτές, όπως παρουσιάζονται στο [Zah71], έχουν ως κοινό χαρακτηριστικό τους το ότι βασίζονται στη χρήση του **ελαχίστου ζευγνύοντος δένδρου** (*minimum spanning tree*) του πλήρους γράφου. Δηλαδή, εφαρμόζουν αρχικώς στον πλήρες γράφο κάποιον από τους γνωστούς αλγόριθμους εύρεσης του ελαχίστου ζευγνύοντος δένδρου, όπως είναι οι [Kru56] και [Pri57], και κατόπιν χρησιμοποιούν μόνον αυτό, αγνοώντας τις υπόλοιπες ακμές του γράφου (οι δύο αυτοί αλγόριθμοι υπολογισμού του ελαχίστου ζευγνύοντος δένδρου ενός γράφου παρατίθενται στο παράρτημα Β). Μία σχηματική παράσταση ενός

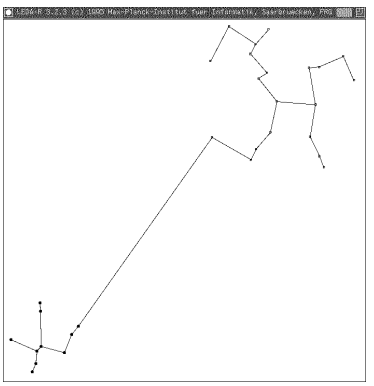
συνόλου δεδομένων που αποτελείται από δύο ομάδες φαίνεται στο σχήμα 4.1(α), ενώ στο σχήμα 4.1(β) φαίνεται ο πλήρης γράφος που σχηματίζεται από αυτά τα δεδομένα και στο σχήμα 4.1(γ) το ελάχιστο ζευγνύον δένδρο αυτού. Οι εικόνες του σχήματος 4.1 δημιουργήθηκαν με τη χρήση της LEDA [MN89], [MN95], [Näh90], μίας βιβλιοθήκης που προσφέρει δομές δεδομένων και αλγόριθμους για προβλήματα Συνδυαστικής και Γεωμετρίας, η οποία έχει χρησιμοποιηθεί σε μεγάλο βαθμό κατά την ανάπτυξη των αλγόριθμων που περιγράφονται στο κεφάλαιο αυτό.



(α) Δύο ομάδες δεδομένων



(β) Ο πλήρης γράφος των δεδομένων



(γ) Το ελάχιστο ζευγνύον δένδρο του πλήρους γράφου

Σχήμα 4.1: Λειτουργία της γραφοθεωρητικής μεθόδου: κατασκευή πλήρους γράφου και του ελαχίστου ζευγνύοντος δένδρου αυτού από ένα σύνολο δεδομένων

Βασιζόμενες στο ελάχιστο ζευγνύον δένδρο και όχι στον αρχικό γράφο, επιτυγχάνουν μία σημαντική συμπίεση της υπάρχουσας πληροφορίας, αφού από $\frac{n*(n-1)}{2}$ ακμές του πλήρους γράφου, καταλήγουν να χρησιμοποιούν μόνο τις $n-1$, όπου n είναι ο αριθμός των κόμβων - δεδομένων.

Επιτυγχάνουν επίσης, μία καλύτερη περιγραφή της δομής της υπάρχουσας πληροφορίας, αφού τώρα είναι πολύ πιο εύκολο να εντοπιστούν τα δεδομένα εκείνα που είναι πλησιέστερα σε κάποιο άλλο. Αυτό φαίνεται κι από το σχήμα 4.1, όπου για μόνο 10 δεδομένα της μίας ομάδας και 20 δεδομένα της άλλης, ο πλήρης γράφος δεν δίνει καμία πρακτική οπτική πληροφορία για τις συσχετίσεις των δεδομένων (σχήμα 4.1(β)), αφού περιέχει 435 ακμές, ενώ το ελάχιστο ζευγνύον δένδρο, με μόλις 29 ακμές, βοηθά πολύ περισσότερο στην κατανόηση των συσχετίσεων μεταξύ των δεδομένων (σχήμα 4.1(γ)).

Η βασική ιδέα στην οποία στηρίζονται οι μέθοδοι αυτές, είναι ότι οι ακμές του ζευγνύοντος δένδρου χωρίζονται σε δύο σύνολα, το σύνολο των ακμών που συνδέουν δεδομένα που ανήκουν στην ίδια ομάδα και το σύνολο των ακμών που συνδέουν δεδομένα που ανήκουν σε διαφορετικές ομάδες.

Επομένως, το πρόβλημα τώρα ορίζεται ως η εύρεση του συνόλου αυτού των ακμών που συνδέουν διαφορετικές ομάδες και η απαλοιφή τους από το ελάχιστο ζευγνύον δένδρο προκειμένου να προκύψουν οι ζητούμενες ομάδες.

4.2 Αρχική Γραφοθεωρητική Μέθοδος Ομαδικής Επεξεργασίας

Μεταξύ των διαφόρων προτεινόμενων μεθόδων ομαδοποίησης ομαδικής επεξεργασίας και δεδομένης της φύσης των δεδομένων τα οποία εμφανίζονται στο πρόβλημα της ομαδο-

ποίησης των δοσοληψιών συμφώνως προς τα χαρακτηριστικά του φόρτου εργασίας τους, επιλέχθηκε τελικώς μία μέθοδος η οποία εφαρμόζει στο ελάχιστο ζευγνύον δένδρο την διαδικασία διάσπασης αυτού σε υποδένδρα, που αντιπροσωπεύουν τις ζητούμενες ομάδες, που φαίνεται στον πίνακα 4.1.

Για κάθε ακμή, e, του δένδρου:

[Βήμα 1]:

Υπολόγισε τον μέσο όρο των μηκών των γειτονικών ακμών της e.

Έστω $\mathcal{M}(e)$ η τιμή αυτή του μέσου όρου.

Ως γειτονική ακμή της e θεωρείται κάθε ακμή του γράφου που βρίσκεται έως $\mathcal{L}$ βήματα μακριά από την e και στις μεταξύ τους ακμές δεν υπάρχει κάποια ακμή που να συνδέει διαφορετικές ομάδες. Δηλαδή, αν κατά τον υπολογισμό του μέσου όρου συναντηθεί μία συνδετική ακμή, αυτή και οι ακμές που την ακολουθούν προς την κατεύθυνση *αυτή και μόνον αυτή* αγνοούνται και συνεχίζεται ο υπολογισμός του μέσου όρου με τις ακμές που εκτείνονται προς τις άλλες κατευθύνσεις.

[Βήμα 2]:

Εάν w(e) είναι το μήκος της ακμής e, τότε:

Εάν $\frac{w(e)}{M(e)} > \mathcal{P}$, τότε χαρακτήρισε την e ως ακμή που συνδέει διαφορετικές ομάδες, διασπώντας ουσιαστικώς το δένδρο και δημιουργώντας μία νέα ομάδα.

Πίνακας 4.1: Γραφοθεωρητικός - ομαδικής επεξεργασίας - αλγόριθμος διάσπασης ενός ελαχίστου ζευγνύοντος δένδρου σε υποδένδρα.

Στον αλγόριθμο του πίνακα 4.1, τα $\mathcal{L}$ (μέγιστος αριθμός βημάτων προκειμένου να μεταβούμε σε μία γειτονική ακμή, ξεκινώντας από την τρέχουσα) και $\mathcal{P}$ (ποσοστό υπέρβασης του μέσου όρου των βαρών των γειτονικών ακμών, προκειμένου να θεωρηθεί μία ακμή ως συνδέουσα δύο διαφορετικές ομάδες δεδομένων) είναι παράμετροι του αλγόριθμου. Η τιμή του $\mathcal{P}$ κυμαίνεται συνήθως μεταξύ του 1.5 και του 2, ενώ η τιμή του $\mathcal{L}$ εξαρτάται από τη φύση του εκάστοτε προβλήματος.

Η ιδέα πίσω από αυτόν τον αλγόριθμο είναι ότι οι ακμές που ενώνουν δεδομένα της ίδιας ομάδας θα έχουν εν γένει παρόμοια μήκη - βάρη, ενώ αυτές που συνδέουν δεδομένα διαφορετικών ομάδων θα έχουν μήκη σημαντικά μεγαλύτερα από τις πρώτες. Έτσι, καθώς μία ακμή που συνδέει διαφορετικές ομάδες γειτονεύει υποχρεωτικώς με κάποιες από τις εσωτερικές ακμές των ομάδων αυτών, η διαφορά αυτή θα φανεί κατά τη σύγκριση του μήκους της με τον μέσο όρο των μηκών των γειτονικών της ακμών.

Ο αλγόριθμος του πίνακα 4.1 σταματά όταν πλέον έχει ελέγξει όλες τις ακμές που αποτελούν το ελάχιστο ζευγνύον δένδρο ($n-1$ το πλήθος).

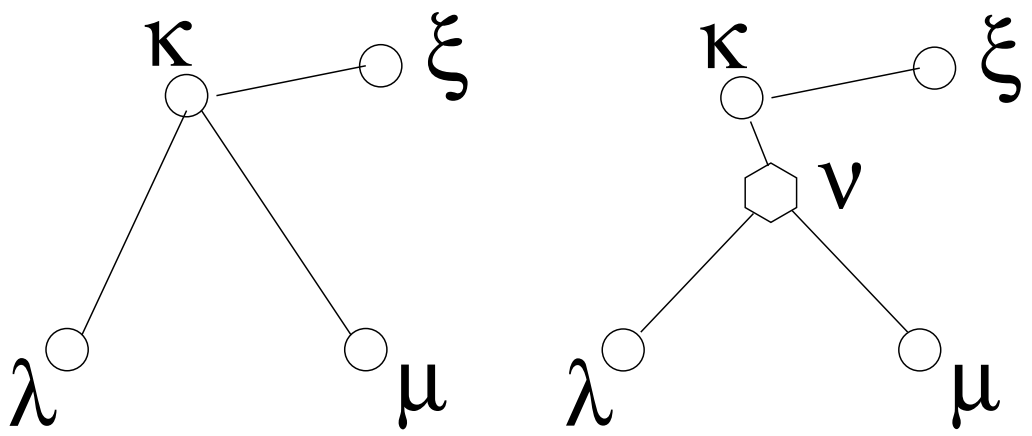
Η μέθοδος αυτή έχει το πλεονέκτημα ότι δεν χρειάζεται να καθορισθεί ο αριθμός των ομάδων που υπάρχουν στο σύνολο των στοιχείων, αλλά παράγει η ίδια το καλύτερο σύνολο ομάδων.

Προκειμένου να γίνει μία αρχική μελέτη της συμπεριφοράς της μεθόδου αυτής, χρησιμοποιήθησαν τα τεχνητώς κατασκευασμένα δεδομένα, που περιγράφονται στο υποκεφάλαιο 5.1, όπως είχε γίνει και με τα νευρωνικά δίκτυα που περιγράφονται στο κεφάλαιο 3. Τα αποτελέσματα των πειραμάτων αυτών παρουσιάζονται στο υποκεφάλαιο 5.4.

4.3 Μετατροπή της Μεθόδου Ομαδικής Επεξεργασίας σε Μέθοδο Ομαδοποίησης ''Εν Πτήσει''

Καθώς στην παρούσα εργασία το ενδιαφέρον εστιάζεται στη χρήση μεθόδων ομαδοποίησης πάνω σε δυναμικώς μεταβαλλόμενες ακολουθίες δεδομένων, έγινε μία προσπάθεια επέκτασης της προηγούμενης μεθόδου προκειμένου να μπορεί να κάνει ομαδοποίηση ''εν πτήσει''.

Στην τελική μέθοδο, όταν εμφανίζεται ένα νέο δεδομένο, ν, προς ομαδοποίηση, τότε αναζητείται ο κόμβος - δεδομένο εκείνος του δένδρου, κ, που απέχει την μικρότερη απόσταση από αυτό. Προσθέτοντας την μεταξύ τους ακμή, (κ, ν), στο δένδρο, θα δημιουργηθεί ένα νέο ζευγνύον δένδρο, το οποίο θα καλύπτει και τους $n+1$ κόμβους. Προκειμένου αυτό να είναι και ελάχιστο ζευγνύον δένδρο, θα πρέπει να ελεγχθεί κατά πόσον η απόσταση του ν από τους γειτονικούς κόμβους τού κ είναι μεγαλύτερη από την απόσταση του ίδιου του κ από αυτούς. Αν για κάποιο κόμβο λ, που συνδεόταν με τον κ, ισχύει $w((\nu, \lambda)) < w((\kappa, \lambda))$, τότε η ακμή (κ, λ) θα πρέπει να διαγραφεί και να αντικατασταθεί από την ακμή (ν, λ). Το σχήμα 4.2 παρουσιάζει ένα παράδειγμα της περίπτωσης αυτής.



(α) Το αρχικό ελάχιστο ζευγνύον δένδρο (β) Το τελικό ελάχιστο ζευγνύον δένδρο

Σχήμα 4.2: Προσθήκη ενός νέου δεδομένου ν και μετατροπή του ελαχίστου ζευγνύοντος δένδρου

Το ελάχιστο ζευγνύον δένδρο για τα τρία δεδομένα κ, λ, μ και ξ του σχήματος (α) μετατρέπεται στο ελάχιστο ζευγνύον δένδρο του σχήματος (β), όταν προστίθεται το νέο δεδομένο ν. Οι ακμές (κ, λ) και (κ, μ) αντικαθίστανται από τις (ν, λ) και (ν, μ) αντιστοίχως. *Προσοχή: Η ακμή (κ, ν) προστίθεται πάντοτε!*

Στο σημείο αυτό, θα πρέπει οι ομάδες, που διαχωρίζονταν από ακμές που αντικαταστάθηκαν, να ενοποιηθούν. Δηλαδή, αν οι ομάδες στο σχήμα 4.2(α) ήταν οι κ, λ, ξ και μ, τότε, ακριβώς μετά την προσθήκη του ν και την δημιουργία του νέου ελαχίστου ζευγνύοντος δένδρου, όπως στο σχήμα 4.2(β), θα υπάρχει μόνον μία ομάδα, η $\kappa, \lambda, \mu, \nu, \xi$.

Αφού έχει δημιουργηθεί το νέο ελάχιστο ζευγνύον δένδρο, εφαρμόζεται ο αλγόριθμος του πίνακα 4.1, αλλά μόνον τοπικώς στην ομάδα στην οποία προστέθηκε το νέο δεδομένο, προκειμένου να διασπαστεί αυτή αν χρειάζεται.

Στο σημείο αυτό θα πρέπει να σημειωθεί ότι κατά την προσθήκη του νέου δεδομένου είναι δυνατόν να διαταραχθούν οι μέχρι στιγμής συσχετίσεις μεταξύ των ομάδων. Δηλαδή, μπορεί η προσθήκη του νέου δεδομένου κοντά σε μία ακμή, που μέχρι εκείνη τη στιγμή θεωρούνταν ότι συνδέει δύο διαφορετικές ομάδες, να μεταβάλει τον μέσο όρο των βαρών

των γειτονικών της ακμών. Θα πρέπει επομένως να υπάρχει η δυνατότητα αποχαρακτήρισης ακμών ως συνδετικών και μετατόπισης των ορίων των ομάδων ή ακόμα και σύμπτυξης ομάδων.

Προκειμένου να εξετάζει ο αλγόριθμος μόνον τις ακμές εκείνες που υπάρχει πιθανότητα να χρειάζονται αποχαρακτηρισμό και όχι όλες τις ακμές του γράφου, ο αλγόριθμος κρατά το σύνολο εκείνων των συνδετικών ακμών μεταξύ διαφορετικών ομάδων που διαπερνά καθώς προσπαθεί να υπολογίσει τους μέσους όρους.

Κατόπιν, όταν έχει τελειώσει ο αλγόριθμος του πίνακα 4.1, αρχίζει να εξετάζει τις συνδετικές ακμές που είχε συναντήσει και να επανεξετάζει το κατά πόσον αυτές αποτελούν πράγματι συνδετικές ακμές. Εάν για κάποια εξ αυτών βρει ότι οι συσχετίσεις έχουν πλέον αλλάξει, την αποχαρακτηρίζει, συνδέοντας έτσι τις πρώην διαφορετικές ομάδες σε μία κοινή. Κατόπιν, εφαρμόζει αναδρομικώς όλο τον αλγόριθμο στη νέα αυτή ομάδα, σημειώνοντας τις συνδετικές ακμές που συναντά, δημιουργώντας καινούργιες, κ.ο.κ.

Με τη μέθοδο αυτή επιτυγχάνεται η ενσωμάτωση του νέου δεδομένου και ο χαρακτηρισμός του με σχετικώς μικρό υπολογιστικό κόστος.

Ο τελικός αλγόριθμος παρουσιάζεται στους πίνακες 4.2 και 4.3.

Τελειώνοντας την περιγραφή του αλγόριθμου αυτού, αξίζει να σημειωθεί ότι έχει πολλά κοινά σημεία με έναν άλλον, αρκετά γνωστό αλγόριθμο ομαδοποίησης, τον *Single Linkage*. Ο αλγόριθμος αυτός περιγράφεται διεξοδικώς στο [Har75]. Το σημείο που πρέπει να προσεχθεί ιδιαιτέρως σε αυτόν, είναι ότι δεν προσπαθεί να ελαχιστοποιήσει το μέσο τετραγωνικό σφάλμα, όπως κάνουν ο K-MEANS και ο NN-MSE (και στην ουσία και ο NN-VWMSE), αλλά η ποσότητα που ελαχιστοποιεί είναι το άθροισμα των διαμέτρων των ομάδων, όπως φαίνεται στην εξίσωση 4.1.

$$\text{GT-error}(K) = \sum_{i=1}^{K} \text{diameter}(\text{Cluster}_i) \tag{4.1}$$

Ως διάμετρος μίας ομάδας, $\text{diameter}(\cdot)$, ορίζεται ο ελάχιστος αριθμός G, για τον οποίον όλα τα ζεύγη των στοιχείων της ομάδας αυτής συνδέονται μεταξύ τους με ένα μονοπάτι ακμών, που καμίας το βάρος δεν υπερβαίνει το G. Έτσι, η βέλτιστη ομαδοποίηση θα είναι τα **μέγιστα συνδεδεμένα τμήματα** (*maximal connected components*) του γράφου, όπου δύο κόμβοι θα συνδέονται όταν η απόστασή τους δεν θα υπερβαίνει το G, για κάποιο G.

Το γεγονός αυτό, οδηγεί τον αλγόριθμο στην κατασκευή ομάδων με σχήμα ''λουκάνικου'', αντί για τις **κοίλες** (*convex*) ομάδες στις οποίες καταλήγουν οι K-MEANS, NN-MSE και NN-VWMSE, λόγω της συνάρτησης μέσου τετραγωνικού σφάλματος που ελαχιστοποιούν.

Πληροφοριακά, όπως αναφέρεται και στο [Har75], μία συνάρτηση που θα προσέγγιζε περισσότερο το μέσο τετραγωνικό σφάλμα, αλλά θα χρησιμοποιούσε επίσης τη διάμετρο των ομάδων, είναι αυτή της εξίσωσης 4.2.

$$\text{MSE-diameter}(K) = \sum_{i=1}^{K} (\text{diameter}(\text{Cluster}_i) \times \text{Number of Elements}(\text{Cluster}_i)) \tag{4.2}$$

Αυτές τις τελευταίες παρατηρήσεις θα πρέπει να τις έχει κάποιος υπόψη του όταν θα συγκρίνει τους διάφορους αλγόριθμους μεταξύ τους. Εν τέλει, η επιλογή κάποιου αλγόριθμου θα έχει να κάνει πρώτα από όλα από το είδος των προς ομαδοποίηση δεδομένων και το πώς ορίζεται η έννοια της απόστασης - ομοιότητας σε αυτά.

[Βήμα 1]:

(Στο βήμα αυτό αρχικοποιείται ο γράφος και σχηματίζονται οι αρχικές ομάδες)

Αρχικώς χρησιμοποιούνται όσα δεδομένα υπάρχουν προκειμένου να κατασκευαστεί ο πλήρης γράφος.

Κατασκευάζεται το ελάχιστο ζευγνύον δένδρο του γράφου αυτού και εφαρμόζεται ο αλγόριθμος του πίνακα 4.1, προκειμένου να δημιουργηθούν οι αρχικές ομάδες.

Εάν υπάρχουν ήδη ομαδοποιημένα δεδομένα, π.χ. με τη χρήση των αλγόριθμων ομαδικής επεξεργασίας του CLUE (HALC, K-MEANS ή BOND ENERGY ALGORITHM [MSW72]), τότε για κάθε ομάδα υπολογίζεται ο υπογράφος που σχηματίζει αυτή, καθώς και το ελάχιστο ζευγνύον δένδρο της. Κατόπιν, για κάθε ζεύγος ομάδων-δένδρων, βρίσκεται η μικρότερη ακμή που συνδέει δύο στοιχεία τους και αυτή προστίθεται στον συνολικό γράφο, χαρακτηριζόμενη ως συνδετική.

Κατ' αυτόν τον τρόπο επιτυγχάνεται η χρήση της πρότερης γνώσης που έχουν σχηματίσει οι αλγόριθμοι ομαδικής επεξεργασίας εφαρμοζόμενοι σε **ίχνη** (*traces*) των χαρακτηριστικών των μονάδων φόρτου εργασίας. Η χρήση αυτή προσφέρει και στον χρόνο λειτουργίας του αλγόριθμου, καθώς τώρα δεν χρειάζεται να κατασκευαστεί ολόκληρος ο γράφος αλλά μόνον οι υπογράφοι των ήδη αναγνωρισθεισών ομάδων.

Ατέρμονας βρόχος (Κυρίως τμήμα του αλγόριθμου)

[Βήμα 2]:

(Στο βήμα αυτό ενσωματώνεται ένα νέο δεδομένο στο γράφο)

Για κάθε δεδομένο που εμφανίζεται, βρίσκεται ο κόμβος - δεδομένο εκείνος του γράφου, έστω ο κόμβος κ, που βρίσκεται πιο κοντά του συμφώνως προς την Ευκλείδιο απόσταση.

Δημιουργείται ένας νέος κόμβος, ν, στο γράφο που αντιστοιχεί στο νέο δεδομένο και προστίθεται η ακμή που συνδέει αυτόν με τον κόμβο κ. Καθώς ο κ ήταν ο πλησιέστερος κόμβος προς τον ν, η μεταξύ τους ακμή θα ανήκει στο ελάχιστο ζευγνύον δένδρο του νέου γράφου. Έτσι, δεν χρειάζεται να κατασκευαστεί αυτό από την αρχή.

Ελέγχονται οι γειτονικοί προς τον κ κόμβοι και αντικαθίστανται όσες ακμές χρειάζονται, προκειμένου το προκύπτον ζευγνύον δένδρο να είναι το ελάχιστο. Αν λ ένας από αυτούς, τότε η ακμή (κ, λ) αντικαθίσταται από την (ν, λ) εφόσον ισχύει $w((\nu, \lambda)) < w((\kappa, \lambda))$.

Εάν κάποια από τις ακμές που αντικαταστάθηκαν, (κ, λ), ήταν συνδετική, τότε θα πρέπει οι ομάδες που αυτή διαχώριζε να ενοποιηθούν σε μία μεγαλύτερη.

[Βήμα 3]:

(Στο βήμα αυτό ελέγχεται ο γράφος που προέκυψε από την ενσωμάτωση του νέου δεδομένου για τυχόν αλλαγές που χρειάζεται να γίνουν)

Ξεκινώντας από τον νέο κόμβο ν, ελέγχονται όλες οι ακμές προκειμένου να εντοπιστούν τυχόν συνδετικές. Ο έλεγχος αυτός συνεχίζεται προς κάποια κατεύθυνση μόνον όταν στην κατεύθυνση αυτή δεν έχει ήδη συναντηθεί κάποια προϋπάρχουσα συνδετική ακμή.

Κατά τη διάρκεια της διάσχισης αυτής του γράφου σημειώνονται οι συνδετικές εκείνες ακμές που καθορίζουν τα όρια της ομάδας που περιέχει τον κόμβο ν.

(Συνεχίζεται στον πίνακα 4.3)

Πίνακας 4.2: Γραφοθεωρητικός αλγόριθμος ομαδοποίησης δεδομένων ''εν πτήσει'' (Μέρος πρώτο)

(Συνέχεια από τον πίνακα 4.2)

[Βήμα 4]:

(Στο βήμα αυτό ελέγχονται οι προηγουμένως χαρακτηρισθείσες ως συνδετικές ακμές προκειμένου να διαπιστωθεί κατά πόσο συνεχίζουν να είναι τέτοιες)

Για κάθε συνδετική ακμή που είχε προσεγγισθεί κατά το βήμα 3, επανεξετάζεται η γειτονιά της για να διαπιστωθεί πιθανή μεταβολή της συγκεκριμένης περιοχής που θα οδηγούσε στον αποχαρακτηρισμό της.

Εάν πράγματι διαπιστωθεί κάτι τέτοιο, τότε αποχαρακτηρίζεται η ακμή οπότε και οι δύο ομάδες που συνέδεε προηγουμένως ενοποιούνται σε μία.

Κατόπιν, *αναδρομικώς* εφαρμόζονται τα βήματα 3 και 4 στην νέα αυτή ομάδα, προκειμένου να βρεθεί το νέο όριο των ομάδων, αν υπάρχει.

Επανάληψη

Πίνακας 4.3: Γραφοθεωρητικός αλγόριθμος ομαδοποίησης δεδομένων ''εν πτήσει'' (Μέρος δεύτερο)

Κεφάλαιο 5

Αποτελέσματα Αρχικών Πειραμάτων

Το κεφάλαιο αυτό παρουσιάζει τα αποτελέσματα που ελήφθησαν από μία σειρά αρχικών πειραμάτων που έγιναν προκειμένου να μελετηθεί η επίδοση των αλγόριθμων ομαδοποίησης ‘‘εν πτήσει’’. Από τα αποτελέσματα αυτά έγινε μία αρχική εκτίμηση των δυνατοτήτων, καθώς και των περιορισμών που εμφανίζουν οι αλγόριθμοι, πάνω στην οποία στηρίχθηκε η μετέπειτα ανάπτυξή τους.

Το κεφάλαιο χωρίζεται ως εξής: στο υποκεφάλαιο 5.1 περιγράφεται πώς σχεδιάστηκαν τα πειράματα αυτά και ποιά μορφή είχαν τα δεδομένα που χρησιμοποιήθησαν. Κατόπιν, στο υποκεφάλαιο 5.2 περιγράφονται τα αποτελέσματα που ελήφθησαν από τον αναπροσαρμοζόμενο K-MEANS όταν αυτός προσπαθεί να ελαχιστοποιήσει τη συνάρτηση κόστους VWMSE (βλέπε το υποκεφάλαιο 3.4), ενώ στο υποκεφάλαιο 5.3 παρουσιάζονται τα αποτελέσματα που ελήφθησαν από τον αναπροσαρμοζόμενο K-MEANS όταν αυτός προσπαθεί να ελαχιστοποιήσει τη συνάρτηση κόστους MSE (βλέπε το υποκεφάλαιο 3.5). Τέλος, στο υποκεφάλαιο 5.4 παρουσιάζονται τα αποτελέσματα που ελήφθησαν από το γραφοθεωρητικό αλγόριθμο ομαδοποίησης ομαδικής επεξεργασίας (βλέπε τον πίνακα 4.1 του υποκεφαλαίου 4.2).

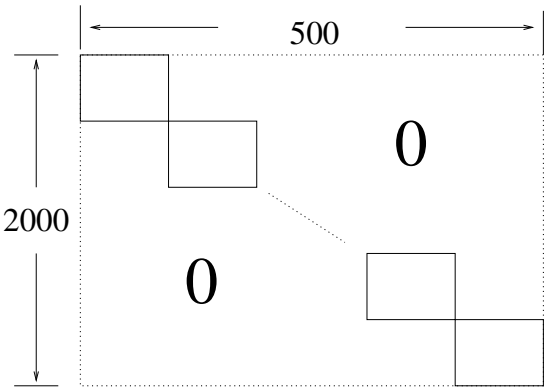
5.1 Σχεδιασμός Αρχικών Πειραμάτων

Για τον αρχικό έλεγχο και μελέτη των διαφορετικών υλοποιήσεων των νευρωνικών δικτύων καθώς και της γραφοθεωρητικής μεθόδου, χρησιμοποιήθησαν κάποια τεχνητώς κατασκευασμένα δεδομένα.

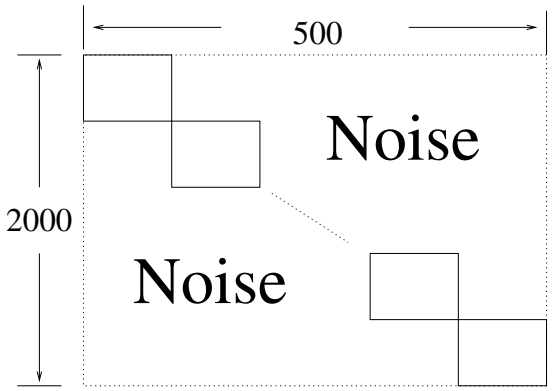
Τα δεδομένα αυτά αποτελούνται από διανύσματα ακεραίων αριθμών. Η διάστασή τους μπορεί να θεωρηθεί ότι αντιπροσωπεύει το πλήθος των σελίδων που υπάρχουν στην κατανεμημένη βάση δεδομένων, ενώ οι διάφορες τιμές τις αναφορές σε κάποια συγκεκριμένη σελίδα που κάνει η μονάδα φορτίου.

Συνολικά χρησιμοποιήθησαν τέσσερα διαφορετικά σύνολα δεδομένων, όπου η διάστασή τους ήταν ίση με 500, και το κάθε σύνολο περιείχε 2000 στοιχεία μοιρασμένα σε 10 ομάδες. Τα τέσσερα αυτά σύνολα παραδειγμάτων ήταν:

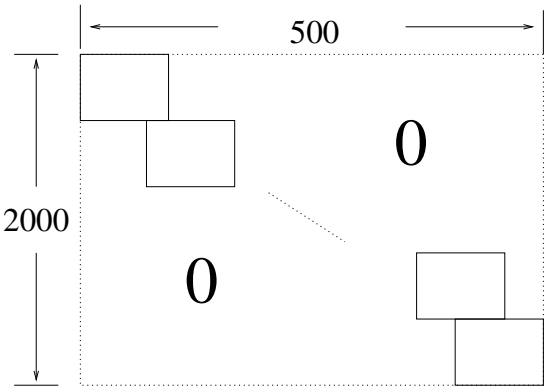
- Μη επικαλυπτόμενα τετράγωνα.
- Μη επικαλυπτόμενα τετράγωνα με θόρυβο.
- Μερικώς επικαλυπτόμενα τετράγωνα.
- Μερικώς επικαλυπτόμενα τετράγωνα με θόρυβο.



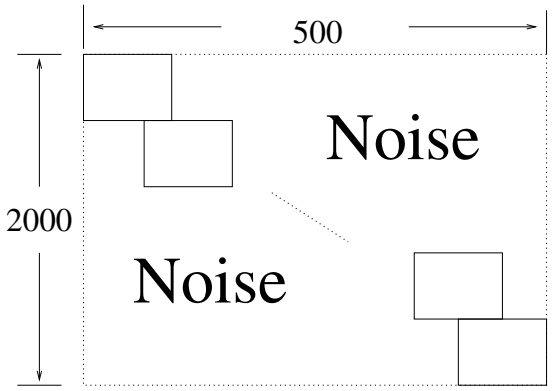
(α) Μη επικαλυπτόμενα τετράγωνα



(β) Μη επικαλυπτόμενα τετράγωνα με θόρυβο



(γ) Μερικώς επικαλυπτόμενα τετράγωνα



(δ) Μερικώς επικαλυπτόμενα τετράγωνα με θόρυβο

Σχήμα 5.1: Γραφική παράσταση των διαφορετικών τύπων δεδομένων
Οι διαστάσεις του κάθε τετραγώνου επιλέχθηκαν τυχαία. Ως θόρυβο θεωρούμε τυχαίες μη μηδενικές τιμές που βρίσκονται εκτός του τετραγώνου που ορίζει τη συγκεκριμένη ομάδα.

Το σχήμα 5.1 παρουσιάζει μία γραφική παράσταση αυτών.

Η επιλογή των ανωτέρω τύπων δεδομένων έγινε προκειμένου να δοκιμασθεί η επίδοση των αλγόριθμων ομαδοποίησης ‘‘εν πτήσει’’ σε διαδοχικώς δυσκολότερα σύνολα δεδομένων.

Είναι εύκολο να δει κανείς ότι το σύνολο των μη επικαλυπτόμενων τετραγώνων είναι το απλούστερο, αφού δεδομένα που ανήκουν στην ίδια ομάδα θα έχουν πάντοτε μικρότερη Ευκλίδεια απόσταση από δεδομένα που ανήκουν σε δύο διαφορετικές ομάδες. Με την προσθήκη ενός σχετικού θορύβου εκτός των τετραγώνων που ορίζουν την κάθε ομάδα, το πρόβλημα δυσκολεύει καθώς τώρα οι αλγόριθμοι θα πρέπει να καταφέρουν να αναγνωρίσουν το θόρυβο και να ξεχωρίσουν πότε κάποιες αναφορές οφείλονται σε αυτόν και πότε οφείλονται στα χαρακτηριστικά της ομάδας του φορτίου εργασίας.

Το σύνολο των μερικώς επικαλυπτόμενων τετραγώνων είναι ακόμα δυσκολότερο, αφού σε αυτό αναφορές σε συγκεκριμένες σελίδες της βάσης δεδομένων αποτελούν κοινά χαρακτηριστικά τουλάχιστον δύο ομάδων. Έτσι οι αλγόριθμοι αναγκάζονται να διαχωρίζουν τα δεδομένα βασιζόμενοι σε ένα μικρότερο σύνολο χαρακτηριστικών αυτών.

Τέλος, το σύνολο των μερικώς επικαλυπτόμενων τετραγώνων είναι το πλέον δύσκολο,

αφού συνδυάζει την εμφάνιση κοινών χαρακτηριστικών σε περισσότερες της μίας ομάδας, καθώς και την ύπαρξη θορύβου στα δεδομένα.

5.2 Αποτελέσματα του Αναπροσαρμοζόμενου K-MEANS (VWMSE)

Στα κάτωθι σχήματα παρατίθενται τα αποτελέσματα των πειραμάτων που έγιναν με χρήση του αναπροσαρμοζόμενου K-MEANS και συνάρτηση ελαχιστοποίησης την VWMSE. Τα ίδια αποτελέσματα παρουσιάζονται και με τη χρήση απλών πινάκων στο παράρτημα Δ.

Για κάθε ένα από τα σύνολα των παραδειγμάτων, έγιναν δύο πειράματα. Στο πρώτο, τα κέντρα των ομάδων αρχικοποιήθησαν με διανύσματα που είχαν μικρές τυχαίες τιμές (της τάξης του 0.005), ενώ στο δεύτερο με ένα διάνυσμα από την κάθε ομάδα.

Σε όλα τα πειράματα ο αριθμός των πραγματικών ομάδων τέθηκε ίσος με 10, ενώ ο αριθμός των εποχών[1] προκειμένου να εκπαιδευθεί το δίκτυο τέθηκε ίσος με 1000. Ως δεδομένα προς εκπαίδευση των δικτύων χρησιμοποιήθηκαν το 10% των συνολικών δεδομένων κάθε φορά, καθώς τα πειράματα αυτά διεξήχθησαν πριν την ενσωμάτωση των αλγόριθμων στο CLUE, οπότε δεν υπήρχε η δυνατότητα χρήσης προηγούμενης γνώσης αποκτηθείσης από τους αλγόριθμους ομαδοποίησης ομαδικής επεξεργασίας αυτού (HALC, K-MEANS, BOND ENERGY ALGORITHM [MSW72]).

Το σχήμα 5.2 περιέχει τα αποτελέσματα της ομαδοποίησης των μη επικαλυπτόμενων τετραγώνων, όταν τα κέντρα έχουν αρχικοποιηθεί με τυχαίες τιμές.

Τα αντίστοιχα αποτελέσματα για μη επικαλυπτόμενα τετράγωνα, όταν έχουν χρησιμοποιηθεί παραδείγματα ως αρχικές τιμές των κέντρων, φαίνονται στο σχήμα 5.3.

Όταν εισαγάγουμε και θόρυβο (μη μηδενικές τιμές εκτός των τετραγώνων), τότε, για τυχαία αρχικοποίηση των κέντρων λαμβάνουμε τα αποτελέσματα του σχήματος 5.4, ενώ για αρχικοποίηση από τα παραδείγματα, αυτά του 5.5.

Όταν έχουμε μερικώς επικαλυπτόμενα τετράγωνα και τυχαία αρχικοποίηση, τότε λαμβάνουμε τα αποτελέσματα του σχήματος 5.6, ενώ όταν η αρχικοποίηση γίνεται με κάποια εκ των παραδειγμάτων, τότε αυτά του 5.7.

Εάν επιπλέον, εισαγάγουμε και θόρυβο στα μερικώς επικαλυπτόμενα τετράγωνα, τότε τα αποτελέσματα μεταβάλλονται όπως φαίνεται στο σχήμα 5.8, καθώς και στο 5.9.

Η κατακόρυφη διάσταση των προαναφερθέντων σχημάτων δείχνει τις πραγματικές ομάδες των παραδειγμάτων (πραγματικά δεδομένα), ενώ η οριζόντια τις ομάδες που εξήγαγε τελικώς ο αλγόριθμος (εκτιμούμενα αποτελέσματα). Έτσι, π.χ. η θέση (1,7) ενός σχήματος αναφέρει πόσα παραδείγματα της πρώτης ομάδας τοποθετήθηκαν τελικώς στην έβδομη.

Παρατηρώντας τα αποτελέσματα, βλέπουμε ότι η τυχαία αρχικοποίηση δίνει καλύτερα αποτελέσματα, ιδίως στις περιπτώσεις που υπάρχει θόρυβος ή τα τετράγωνα είναι μερικώς επικαλυπτόμενα.

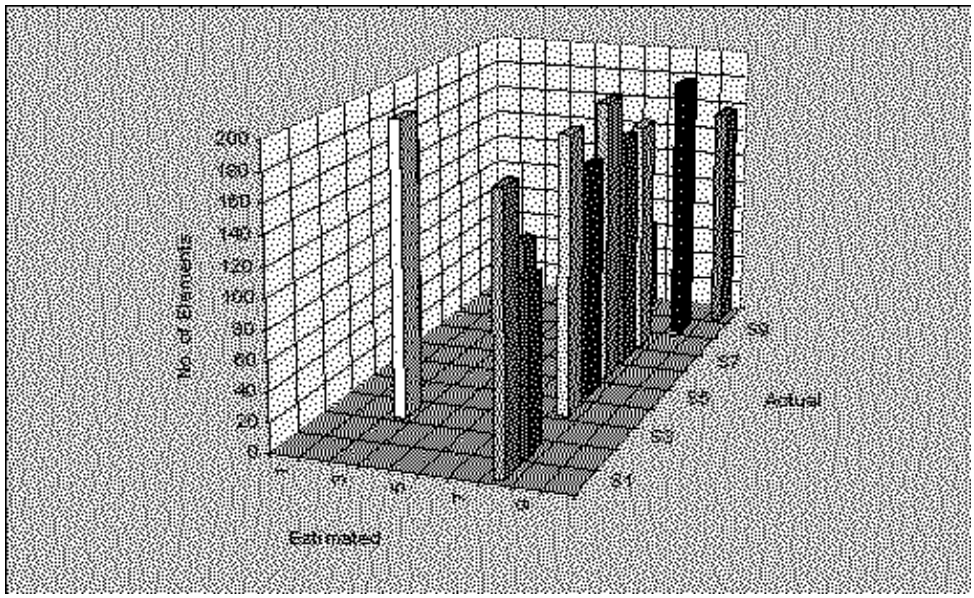
Επίσης, παρατηρούμε ότι στην περίπτωση των μη επικαλυπτόμενων τετραγώνων, η τελική κατάταξη δεν είναι ικανοποιητική, καθώς όλες οι ομάδες εκτός τριών (τρίτη, ένατη και δέκατη) αναγνωρίζονται ώς η δέκατη κλάση και εκτός αυτού η δέκατη έχει διασπασθεί σε άλλες μικρότερες.

Αυτή την κακή συμπεριφορά δεν την έχει ο αλγόριθμος στα μερικώς επικαλυπτόμενα τετράγωνα (ακόμα και παρουσία θορύβου στα δεδομένα).

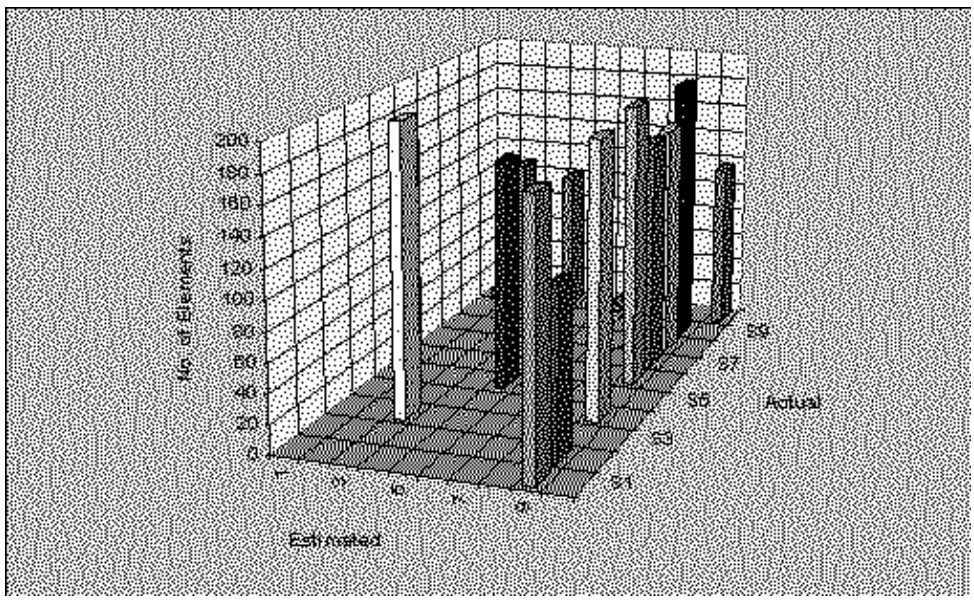
Μία πιθανή εξήγηση για το φαινόμενο αυτό, είναι πως τα παραδείγματα που έχουν μερικώς επικαλυπτόμενα τετράγωνα, είναι πιο ομαλά κατανεμημένα στις διάφορες ομάδες από αυτά που δεν έχουν επικαλυπτόμενα τετράγωνα και έτσι, η διασπορά v_k της κάθε

[1] Η **εποχή** (*epoch*) ορίζεται ως μία παρουσίαση όλων των δεδομένων, που διαθέτουμε προς εκπαίδευση του δικτύου, σε αυτό.

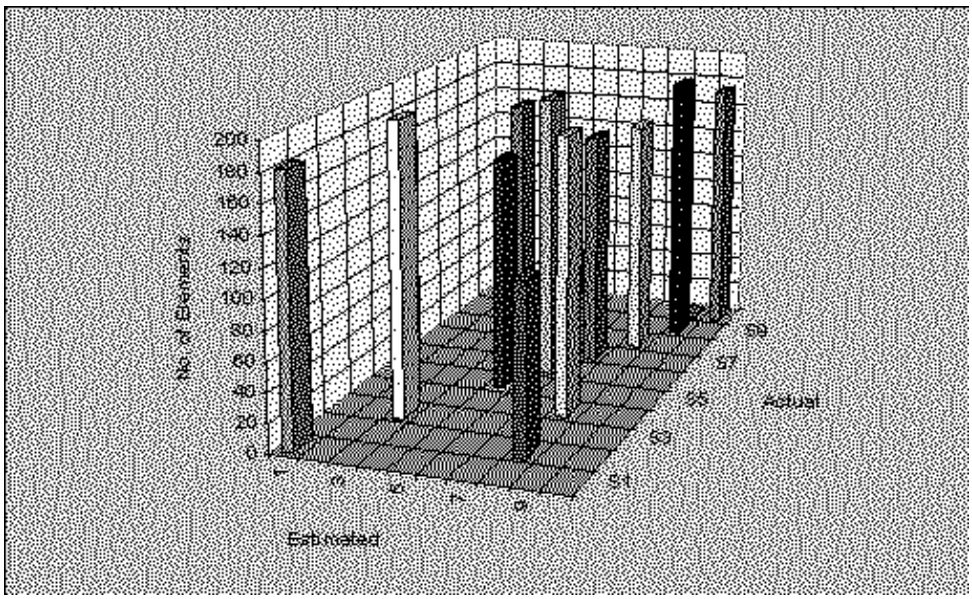
ομάδας είναι περίπου η ίδια. Επομένως, η απόσταση τείνει να προσεγγίσει την Ευκλείδια, που στην περίπτωσή μας διαχωρίζει καλύτερα τις ομάδες.



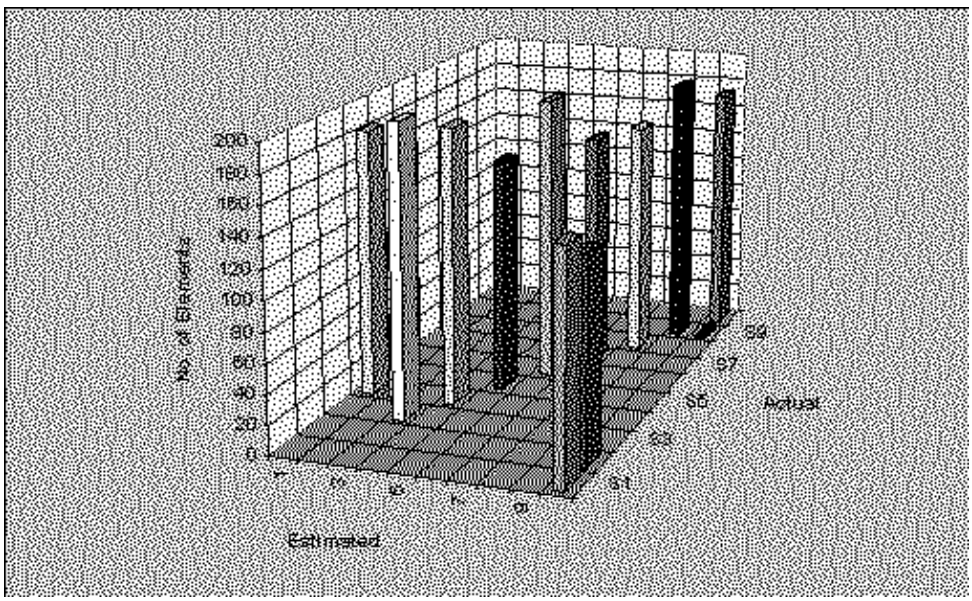
Σχήμα 5.2: NN-VWMSE: Μη επικαλυπτόμενα τετράγωνα δίχως θόρυβο (Αρχικοποίηση τυχαία)
Εκτιμηθείσες ομάδες έναντι των πραγματικών.



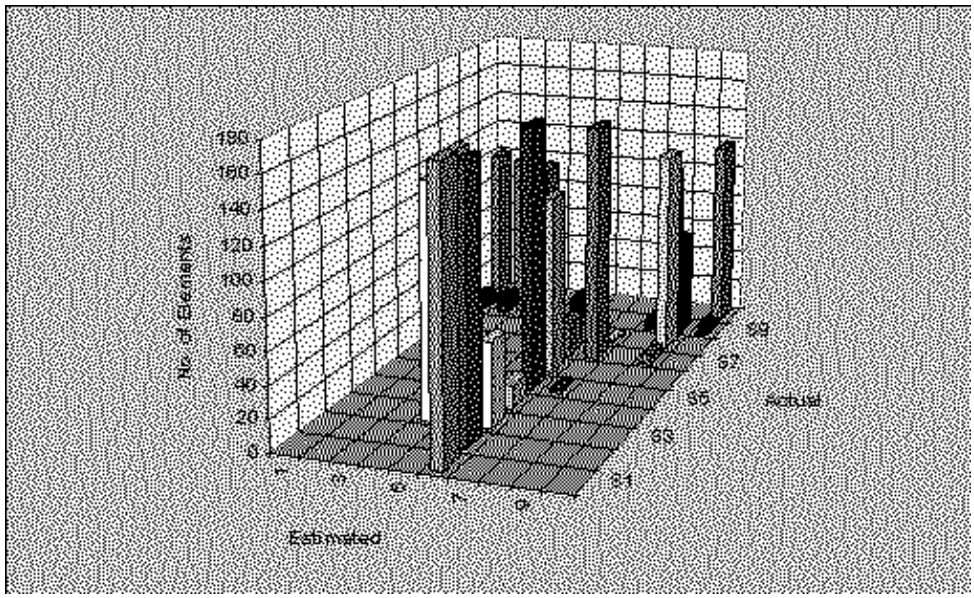
Σχήμα 5.3: NN-VWMSE: Μη επικαλυπτόμενα τετράγωνα δίχως θόρυβο
Εκτιμηθείσες ομάδες έναντι των πραγματικών.



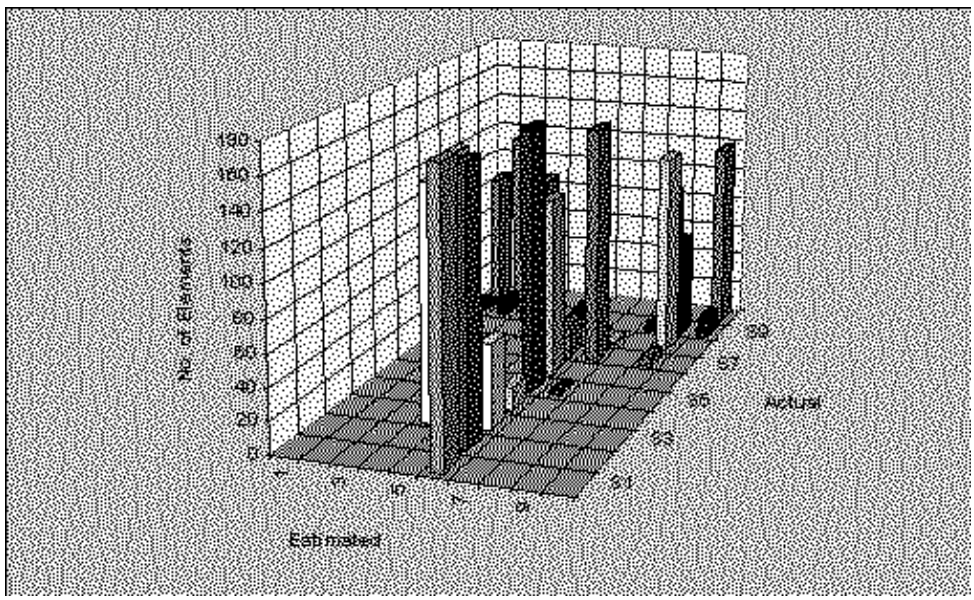
Σχήμα 5.4: NN-VWMSE: Μη επικαλυπτόμενα τετράγωνα με θόρυβο (Αρχικοποίηση τυχαία) Εκτιμηθείσες ομάδες έναντι των πραγματικών.



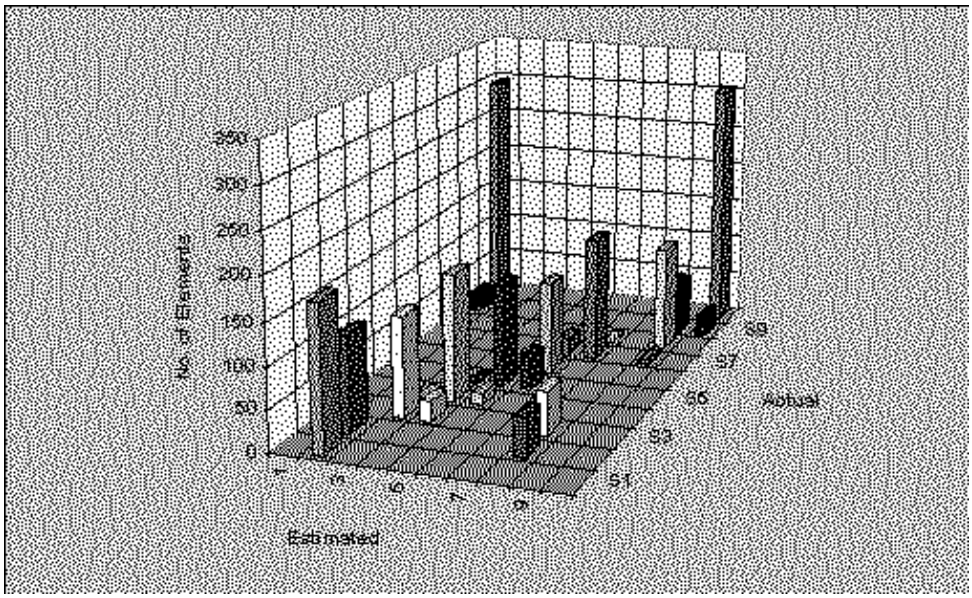
Σχήμα 5.5: NN-VWMSE: Μη επικαλυπτόμενα τετράγωνα με θόρυβο Εκτιμηθείσες ομάδες έναντι των πραγματικών.



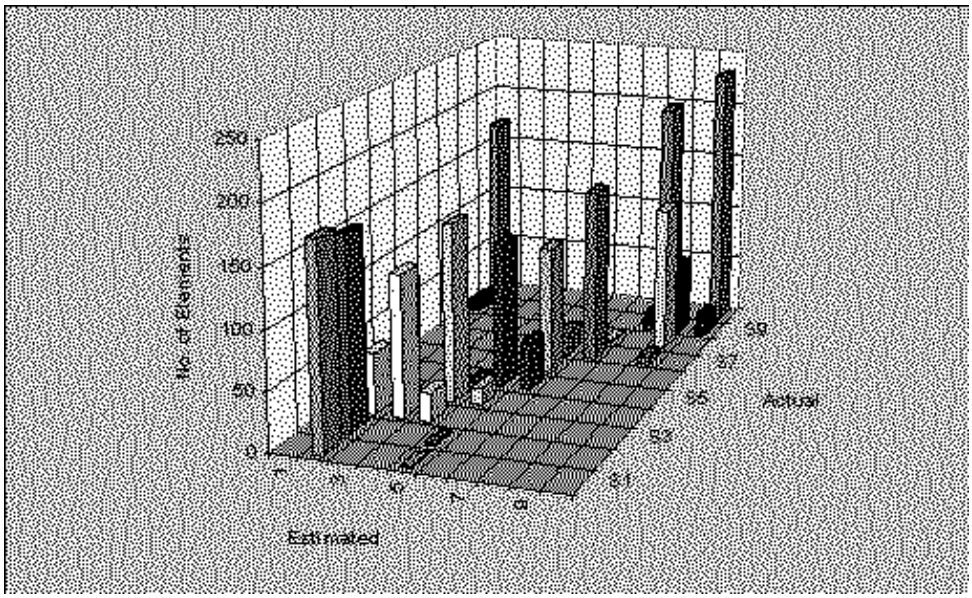
Σχήμα 5.6: NN-VWMSE: Μερικώς επικαλυπτόμενα τετράγωνα δίχως θόρυβο (Αρχικοποίηση τυχαία)
Εκτιμηθείσες ομάδες έναντι των πραγματικών.



Σχήμα 5.7: NN-VWMSE: Μερικώς επικαλυπτόμενα τετράγωνα δίχως θόρυβο
Εκτιμηθείσες ομάδες έναντι των πραγματικών.



Σχήμα 5.8: NN-VWMSE: Μερικώς επικαλυπτόμενα τετράγωνα με θόρυβο (Αρχικοποίηση τυχαία)
Εκτιμηθείσες ομάδες έναντι των πραγματικών.

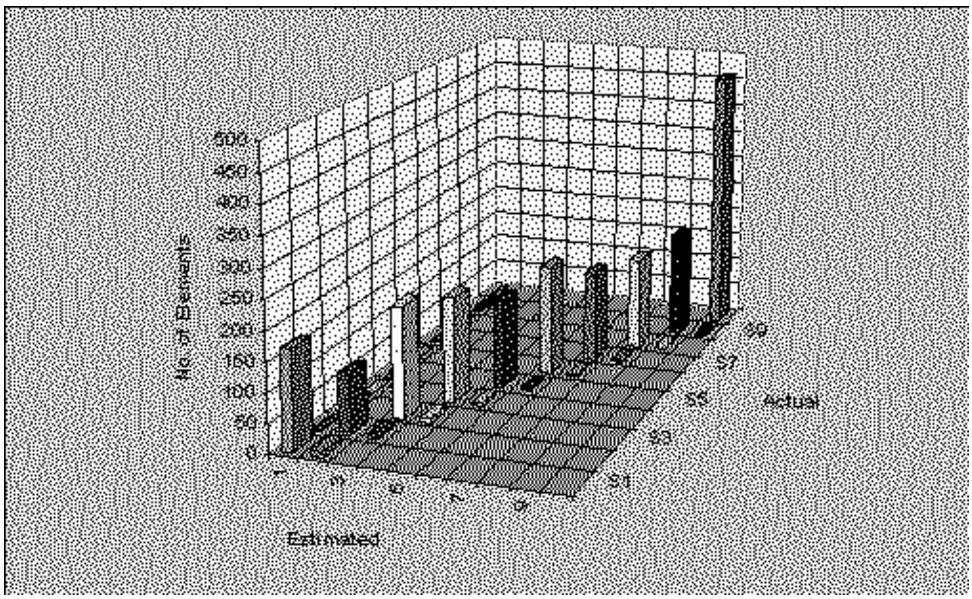


Σχήμα 5.9: NN-VWMSE: Μερικώς επικαλυπτόμενα τετράγωνα με θόρυβο
Εκτιμηθείσες ομάδες έναντι των πραγματικών.

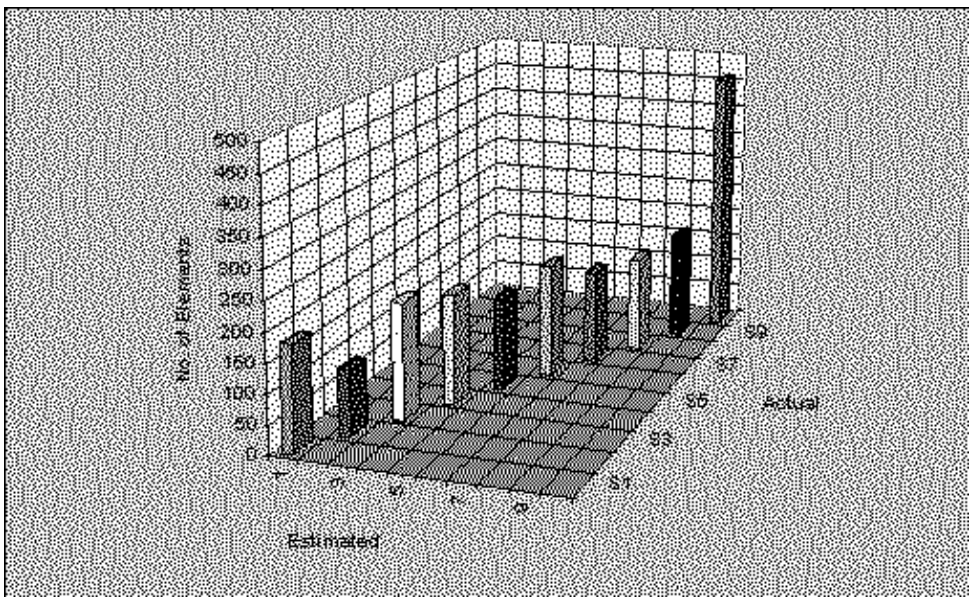
5.3 Πειράματα με συνάρτηση κόστους τη MSE

Στο υποκεφάλαιο αυτό παρουσιάζονται τα αποτελέσματα των πειραμάτων που έγιναν με χρήση του αναπροσαρμοζόμενου K-MEANS και συνάρτηση ελαχιστοποίησης την MSE. Τα ίδια αποτελέσματα παρουσιάζονται και με τη χρήση απλών πινάκων στο παράρτημα Δ.

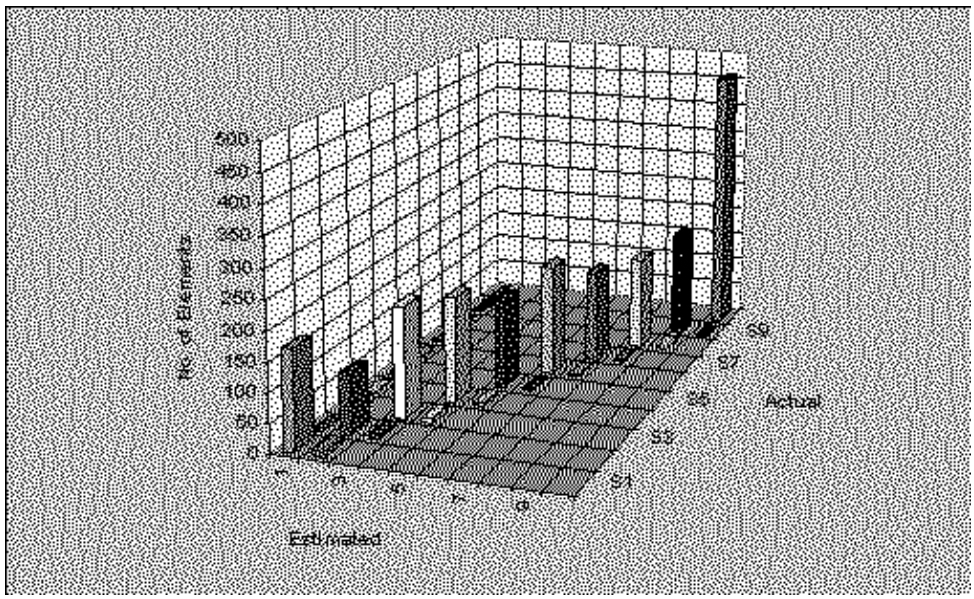
Μελετώντας τα σχήματα 5.10, 5.11, 5.12, 5.13, 5.14, 5.15, 5.16 και 5.17 είναι φανερό ότι το νευρωνικό δίκτυο επιδεικνύει πολύ καλύτερη συμπεριφορά και μεγαλύτερη ακρίβεια ομαδοποίησης όταν χρησιμοποιείται ως συνάρτηση ελαχιστοποίησης η MSE, δηλαδή όταν χρησιμοποιείται η κλασσική Ευκλείδια απόσταση αντί της βεβαρημένης Ευκλείδιας συνάρτησης απόστασης που προτείνεται στο [CS95].



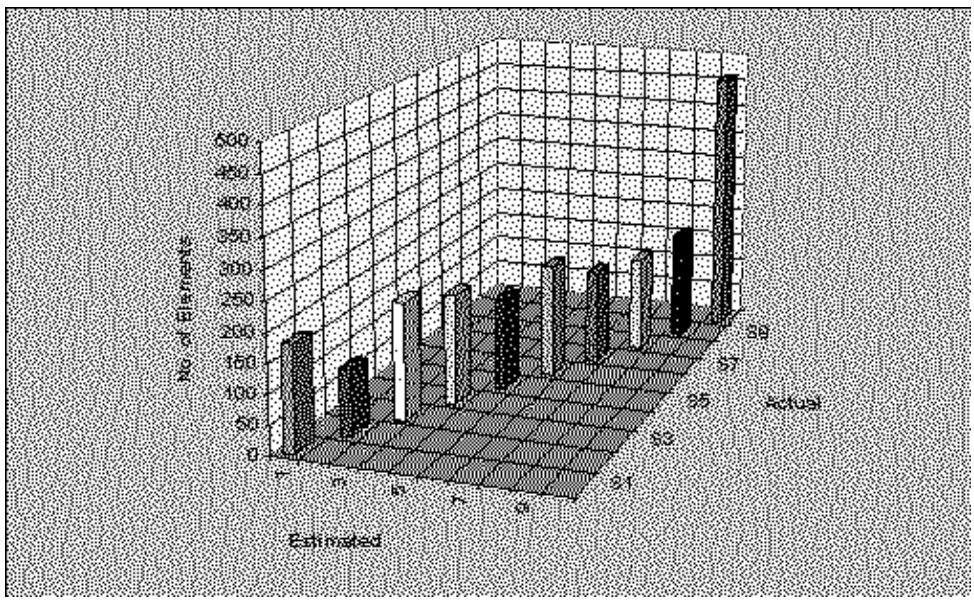
Σχήμα 5.10: NN-MSE: Μη επικαλυπτόμενα τετράγωνα δίχως θόρυβο (Αρχικοποίηση τυχαία) Εκτιμηθείσες ομάδες έναντι των πραγματικών.



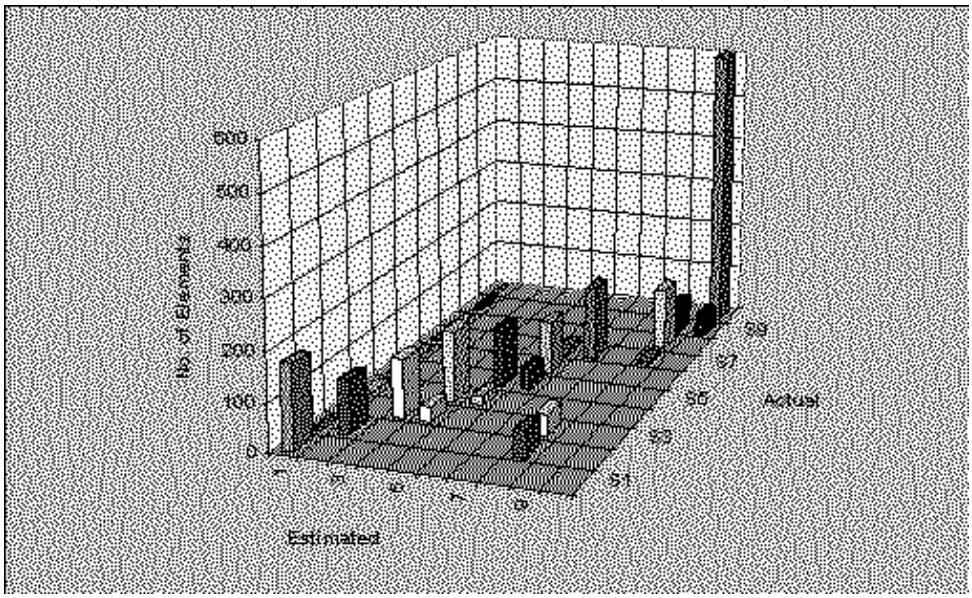
Σχήμα 5.11: NN-MSE: Μη επικαλυπτόμενα τετράγωνα δίχως θόρυβο Εκτιμηθείσες ομάδες έναντι των πραγματικών.



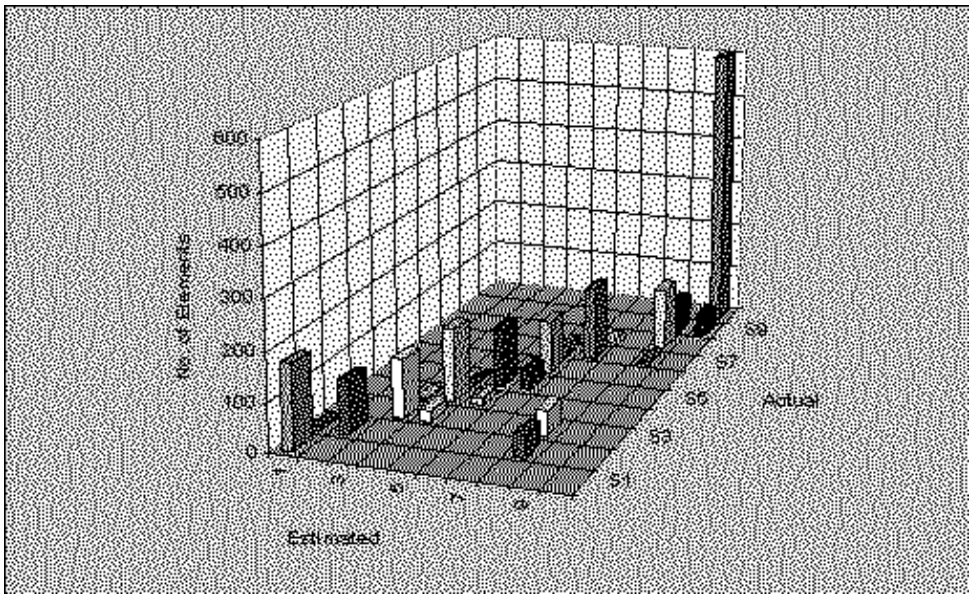
Σχήμα 5.12: NN-MSE: Μη επικαλυπτόμενα τετράγωνα με θόρυβο (Αρχικοποίηση τυχαία) Εκτιμηθείσες ομάδες έναντι των πραγματικών.



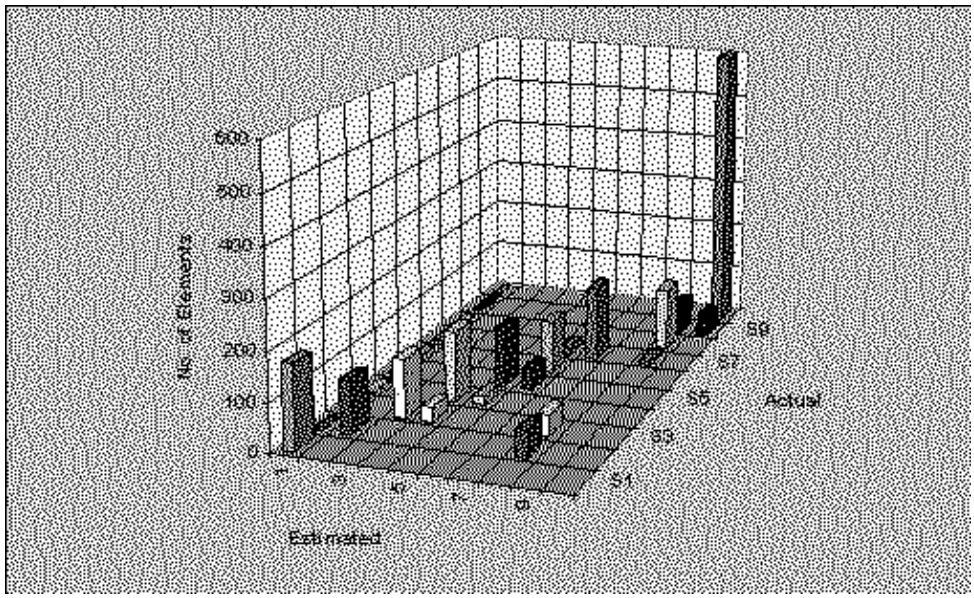
Σχήμα 5.13: NN-MSE: Μη επικαλυπτόμενα τετράγωνα με θόρυβο Εκτιμηθείσες ομάδες έναντι των πραγματικών.



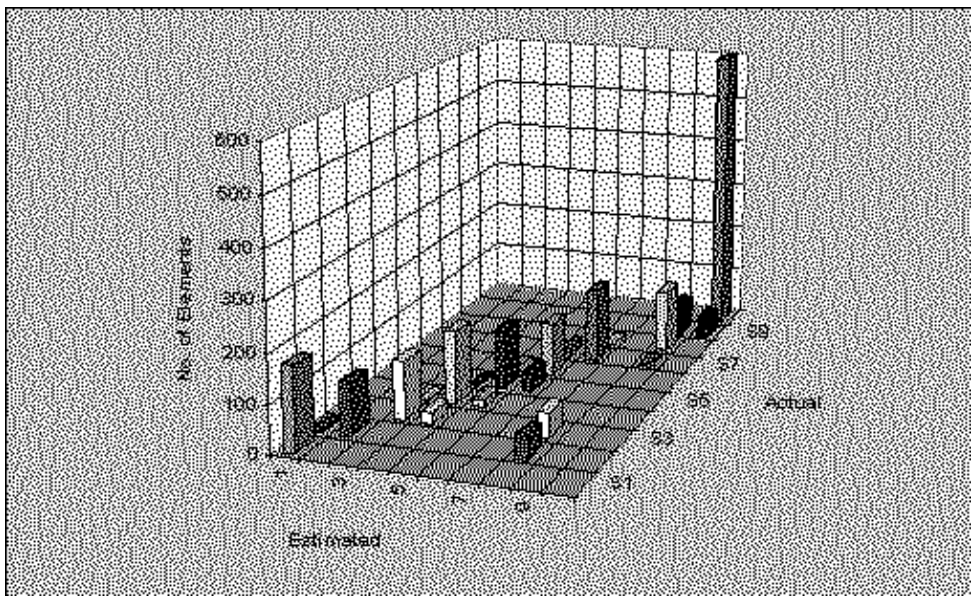
Σχήμα 5.14: NN-MSE: Μερικώς επικαλυπτόμενα τετράγωνα δίχως θόρυβο (Αρχικοποίηση τυχαία)
Εκτιμηθείσες ομάδες έναντι των πραγματικών.



Σχήμα 5.15: NN-MSE: Μερικώς επικαλυπτόμενα τετράγωνα δίχως θόρυβο
Εκτιμηθείσες ομάδες έναντι των πραγματικών.



Σχήμα 5.16: NN-MSE: Μερικώς επικαλυπτόμενα τετράγωνα με θόρυβο (Αρχικοποίηση τυχαία)
Εκτιμηθείσες ομάδες έναντι των πραγματικών.



Σχήμα 5.17: NN-MSE: Μερικώς επικαλυπτόμενα τετράγωνα με θόρυβο
Εκτιμηθείσες ομάδες έναντι των πραγματικών.

5.4 Αποτελέσματα με τη Γραφοθεωρητική Μέθοδο Ομαδοποίησης Ομαδικής Επεξεργασίας

Στα σχήματα που ακολουθούν 5.18, 5.19, 5.20, και 5.21, παρουσιάζονται τα αποτελέσματα των πειραμάτων με τη γραφοθεωρητική μέθοδο ομαδοποίησης ομαδικής επεξεργασίας (βλέπε τον πίνακα 4.1 του υποκεφαλαίου 4.2).

Τα ίδια αποτελέσματα παρουσιάζονται και με τη χρήση απλών πινάκων στο παράρτημα Ε.

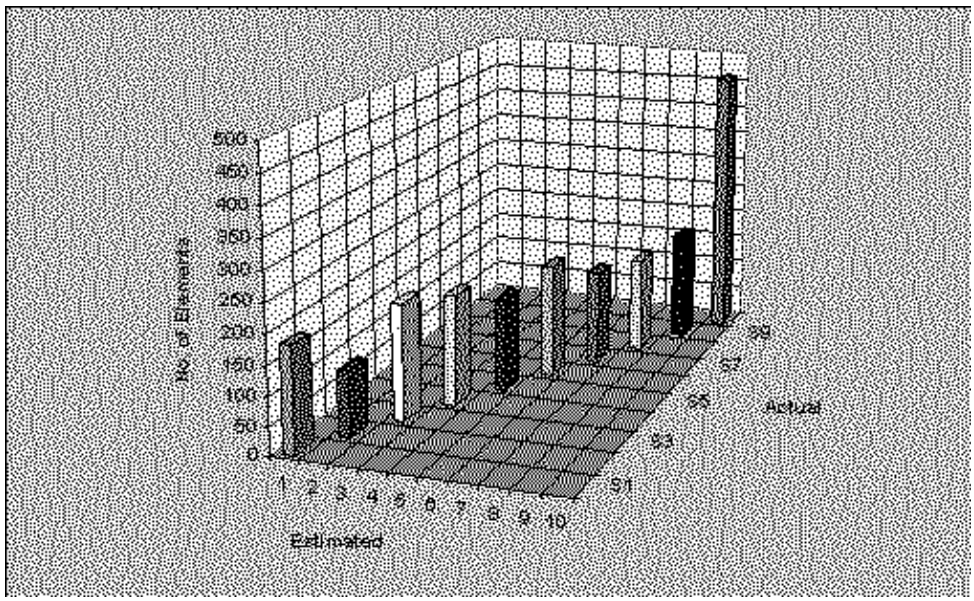
Οι τιμές που χρησιμοποιήθησαν ήταν:

$\mathcal{K}$ Αριθμός 'πραγματικών' ομάδων των δεδομένων = 10

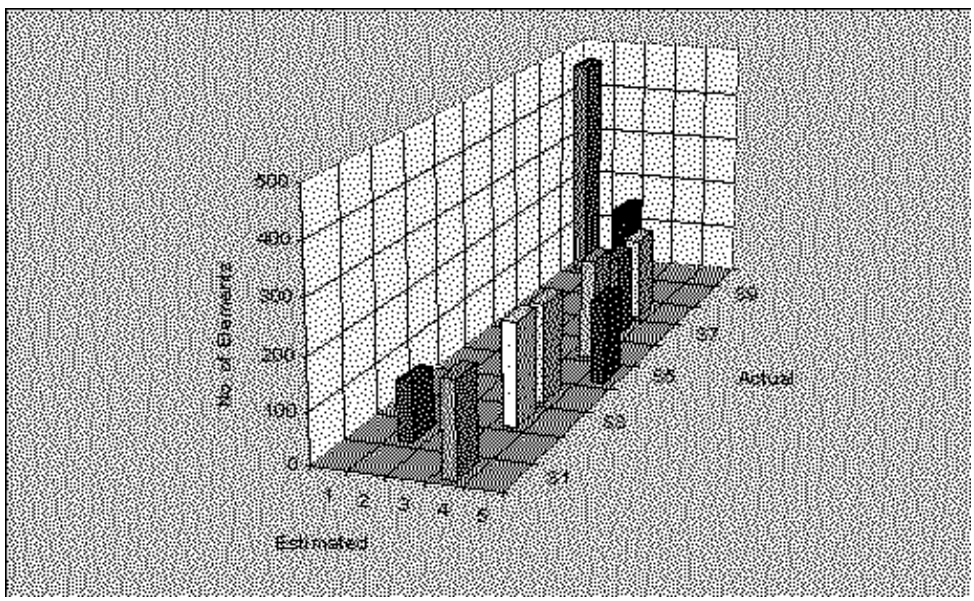
$\mathcal{L}$ Μήκος μονοπατιού = 10

$\mathcal{P}$ Τιμή που πρέπει να υπερβαίνει το πηλίκο του βάρους μίας ακμής προς τον μέσο όρο των βαρών των γειτονικών της ακμών, προκειμένου να θεωρηθεί αυτή ως συνδετική μεταξύ δύο διαφορετικών ομάδων = 1.5

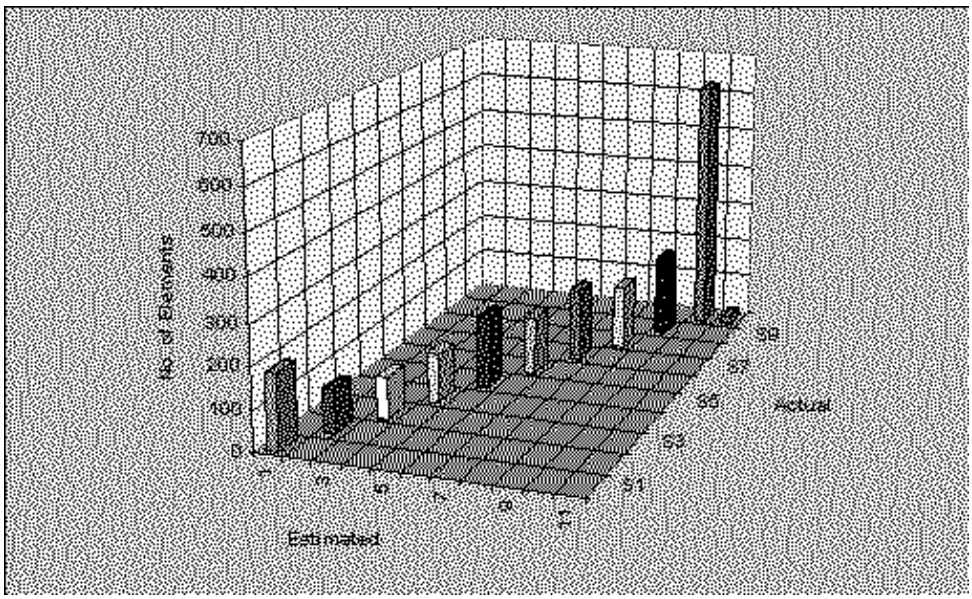
Από τα αποτελέσματα φαίνεται ότι η γραφοθεωρητική μέθοδος ομαδοποίησης εμφανίζει μία αρκετά μεγάλη ευαισθησία στο θόρυβο. Η ευαισθησία της αυτή δεν έγινε δυνατό να προσδιορισθεί αν οφείλεται στην επιλογή των παραμέτρων του αλγόριθμου ($\mathcal{L}$ και $\mathcal{P}$) ή σε εγγενείς αδυναμίες αυτού.



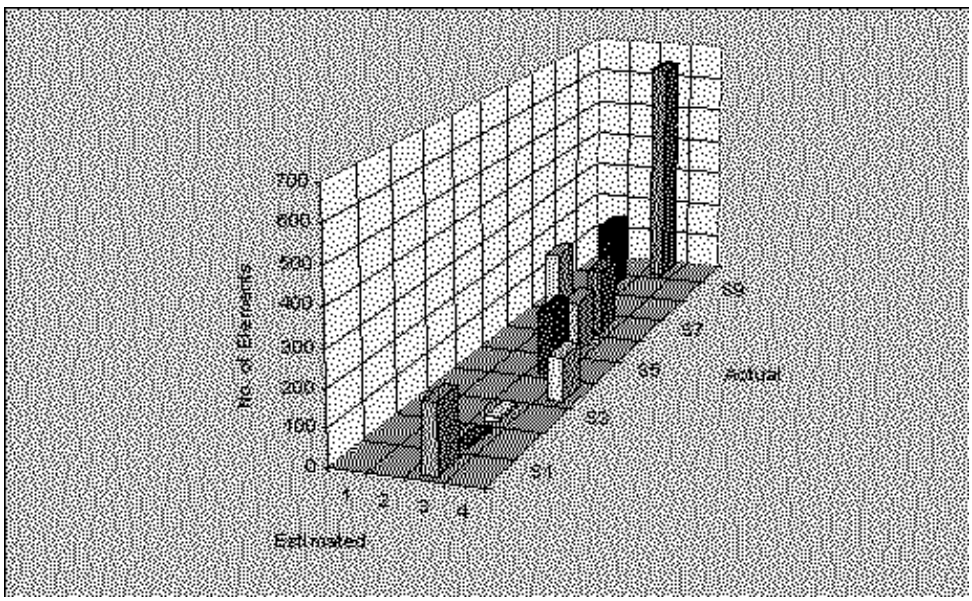
Σχήμα 5.18: Γραφοθεωρητική μέθοδος: Μη επικαλυπτόμενα τετράγωνα δίχως θόρυβο. Εκτιμηθείσες ομάδες έναντι των πραγματικών.



Σχήμα 5.19: Γραφοθεωρητική μέθοδος: Μη επικαλυπτόμενα τετράγωνα με θόρυβο. Εκτιμηθείσες ομάδες έναντι των πραγματικών.



Σχήμα 5.20: Γραφοθεωρητική μέθοδος: Μερικώς επικαλυπτόμενα τετράγωνα δίχως θόρυβο. Εκτιμηθείσες ομάδες έναντι των πραγματικών.



Σχήμα 5.21: Γραφοθεωρητική μέθοδος: Μερικώς επικαλυπτόμενα τετράγωνα με θόρυβο. Εκτιμηθείσες ομάδες έναντι των πραγματικών.

Κεφάλαιο 6

Αποτελέσματα Τελικών Πειραμάτων

Στο κεφάλαιο αυτό περιγράφονται τα αποτελέσματα των τελικών πειραμάτων που έγιναν με τους διάφορους αλγόριθμους ομαδοποίησης ‘‘εν πτήσει’’, καθώς και με τους δύο αλγόριθμους ομαδικής επεξεργασίας, τον K-MEANS και τον HALC.

Τα πειράματα έγιναν με δύο είδη δεδομένων. Αρχικώς χρησιμοποιήθησαν συνθετικά δεδομένα που είχαν κατασκευαστεί από τον Αλέξανδρο Λαμπρινίδη και κατόπιν πραγματικά δεδομένα που προήλθαν από ίχνη που ελήφθησαν από πραγματικές εφαρμογές επεξεργασίας δοσοληψιών. Τα τελευταία παρασχέθησαν από τη Siemens Nixdorf Informationssysteme AG [SNI].

6.1 Σενάριο πειραμάτων

Το σενάριο που ακολουθήθηκε στα πειράματα έχει ως εξής:

Θεωρώντας ότι σε ένα πραγματικό σύστημα επεξεργασίας δοσοληψιών θα λαμβάνεται αρχικώς ένα ίχνος της λειτουργίας του, το οποίο θα χρησιμοποιείται για την αρχική μελέτη της συμπεριφοράς του συστήματος, κάθε σύνολο των δεδομένων χωρίσθηκε σε δύο τμήματα, ένα που θεωρείται ότι αποτελεί το προαναφερθέν ίχνος και ένα που αποτελεί τις τριάδες που εμφανίζονται στο σύστημα όταν πλέον έχουν αρχίσει να τρέχουν οι αλγόριθμοι ομαδοποίησης ‘‘εν πτήσει’’, κατά τη διάρκεια της κανονικής λειτουργίας του όλου συστήματος.

Το πρώτο υποσύνολο περιέχει περίπου το 10% των συνολικών δεδομένων, ενώ το δεύτερο το υπόλοιπο 90%. Προκειμένου τα αποτελέσματα να είναι όσο το δυνατόν ανεξάρτητα του συγκεκριμένου τρόπου με τον οποίο έγινε ο χωρισμός των δεδομένων στα δύο σύνολα, κάθε πείραμα πραγματοποιήθηκε δέκα φορές και κατόπιν χρησιμοποιήθηκε η μέση τιμή των αποτελεσμάτων στις διάφορες γραφικές παραστάσεις που ακολουθούν. Ο διαχωρισμός των δεδομένων έγινε με τη χρήση του προγράμματος TTS, το οποίο παράγεται ταυτοχρόνως με το CLUE. Στο παράρτημα ΣΤ υπάρχει μία περιγραφή του προγράμματος αυτού, καθώς και των ορισμάτων που χρησιμοποιήθησαν στα πειράματα.

Σε κάθε μία από αυτές τις επαναλήψεις, ζητείται από το CLUE να παράγει ένα **στιγμιότυπο** (*snapshot*) της τρέχουσας κατάστασής του σε τακτά χρονικά διαστήματα. Τα στιγμιότυπα αυτά περιείχαν τόσο το σύνολο όλων των δεδομένων που είχαν παρουσιασθεί μέχρι στιγμής στον εκάστοτε αλγόριθμο ομαδοποίησης ‘‘εν πτήσει’’, όσο και την αντίστοιχη ομαδοποίηση που αυτός είχε παραγάγει. Κατόπιν, χρησιμοποιήθησαν οι αλγόριθμοι ομαδοποίησης ομαδικής επεξεργασίας πάνω στα σύνολα δεδομένων των στιγμιοτύπων, ώστε να είναι δυνατή η σύγκριση των ενδιάμεσων αυτών αποτελεσμάτων.

Η ποσότητα που υπολογίζεται σε κάθε στιγμιότυπο, προκειμένου να συγκριθεί η επίδοση

των αλγόριθμων, είναι το **τετραγωνικό σφάλμα** (*square error*), SE, που ορίζεται στην εξίσωση 6.1 (όπου S_i το πλήθος των στοιχείων της i-οστής ομάδας, $\vec{c}_i$ το κέντρο αυτής και $\vec{x}_{i_j}$ το j-οστό στοιχείο της i-οστής ομάδας).

$$SE = \sum_{i=1}^{K} \sum_{j=1}^{S_i} ||\vec{x}_{i_j} - \vec{c}_i||^2 \tag{6.1}$$

6.2 Συνθετικά δεδομένα

Τα συνθετικά δεδομένα δημιουργήθησαν με το εργαλείο TSG που αναπτύχθηκε από τον Αλέξανδρο Λαμπρινίδη στα πλαίσια της μεταπτυχιακής του εργασίας [Lab95]. Αποτελούνται όλα από εκατό ομάδες δεδομένων, η κάθε μία εκ των οποίων περιέχει δεδομένα που κάνουν αναφορές μόνο σε συγκεκριμένα σύνολα σελίδων της υποκείμενης βάσης δεδομένων. Τα σύνολα αυτά των σελίδων είναι ξένα μεταξύ τους, απαρτίζοντας έτσι έναν διαμερισμό των σελίδων της βάσης. Οι αναφορές που κάνει κάθε δεδομένο στις διάφορες σελίδες είναι ένας τυχαίος αριθμός. Τέλος, θα πρέπει να σημειωθεί κάτι αρκετά σημαντικό για τα δεδομένα' παράγονται ανά ομάδα, δηλαδή στο αρχείο που γράφονται, πρώτα υπάρχουν τα δεδομένα της πρώτης ομάδας, μετά αυτά της δεύτερης και ούτω καθεξής. Αυτό έχει σημασία καθότι η σειρά αυτή δεν αναμένεται να συναντάται σε πραγματικά δεδομένα, αφού σε αυτά είναι πιο λογικό να υπάρχει μία πιο τυχαία κατάταξη των δεδομένων ως προς τις ομάδες τους.

Συνοπτικώς, τα σύνολα συνθετικών δεδομένων είναι τα κάτωθι:

1000x1000=100:diagsR 1000 τριάδες που κάνουν αναφορές σε μία βάση 1000 σελίδων. Υπάρχουν 100 ομάδες, τα δε δεδομένα έχουν μορφή παραλληλογράμμων επί τη διαγώνιο, στο χώρο (τριάδες × σελίδες βάσης × πλήθος αναφορών). Οι τιμές των αναφορών είναι τυχαίες.

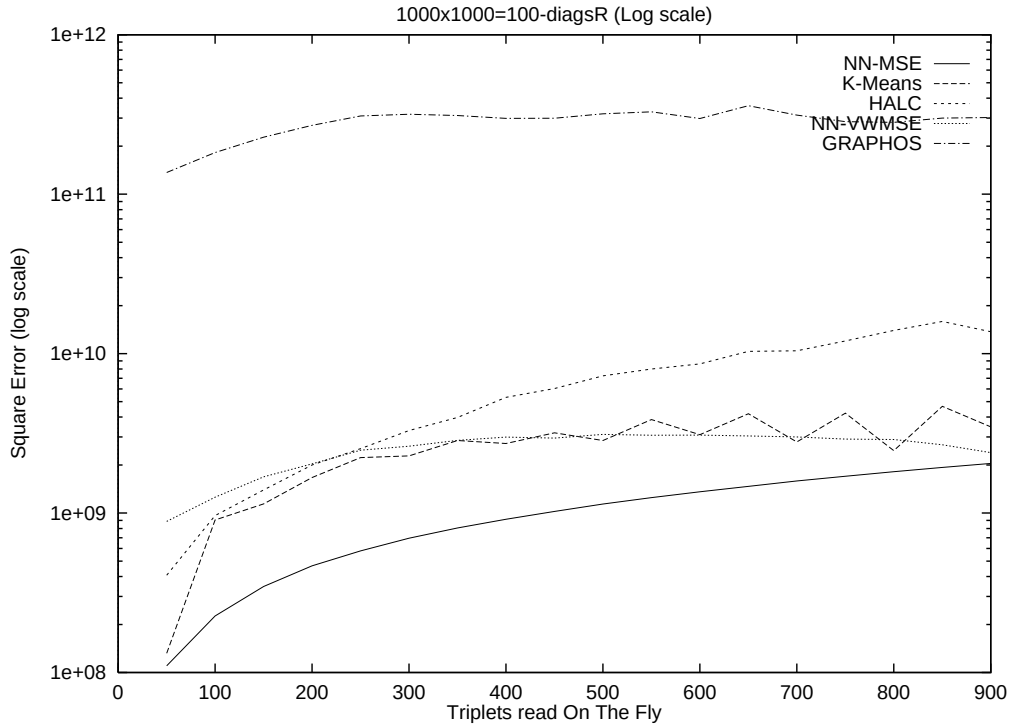
10000x1000=100:diagsR 10000 τριάδες που κάνουν αναφορές σε μία βάση 1000 σελίδων. Υπάρχουν 100 ομάδες, τα δε δεδομένα έχουν μορφή παραλληλογράμμων επί τη διαγώνιο, στο χώρο (τριάδες × σελίδες βάσης × πλήθος αναφορών). Οι τιμές των αναφορών είναι τυχαίες.

30000x1000=100:diagsR 30000 τριάδες που κάνουν αναφορές σε μία βάση 1000 σελίδων. Υπάρχουν 100 ομάδες, τα δε δεδομένα έχουν μορφή παραλληλογράμμων επί τη διαγώνιο, στο χώρο (τριάδες × σελίδες βάσης × πλήθος αναφορών). Οι τιμές των αναφορών είναι τυχαίες.

Στο σχήμα 6.1 παρουσιάζονται τα αποτελέσματα των πειραμάτων που έγιναν με το σύνολο δεδομένων 1000x1000=100:diagsR. Συγκεκριμένα, παρουσιάζονται οι επιδόσεις (τετραγωνικό σφάλμα) όλων των αλγόριθμων σε λογαριθμική κλίμακα. Η επιλογή της λογαριθμικής κλίμακας έγινε διότι η επίδοση του γραφοθεωρητικού αλγόριθμου ήταν πολύ χειρότερη από αυτές των υπολοίπων αλγόριθμων, με αποτέλεσμα οι καμπύλες επίδοσης αυτών να συμπιέζονται στο κάτω μέρος της γραφικής παράστασης, καθιστώντας αδύνατη την μελέτη τους.

Παρατηρώντας το σχήμα 6.1, βλέπουμε ότι το τετραγωνικό σφάλμα αυξάνεται εν γένει συνεχώς. Αυτό οφείλεται στο γεγονός ότι εμφανίζονται συνεχώς νέες τριάδες που προστίθενται στο προς ομαδοποίηση σύνολο.

Ένα άλλο γεγονός, που παρατηρήθηκε και στα υπόλοιπα πειράματα, είναι ότι οι αλγόριθμοι ομαδικής επεξεργασίας, δηλαδή ο K-MEANS και ο HALC, εμφανίζουν χειρότερη

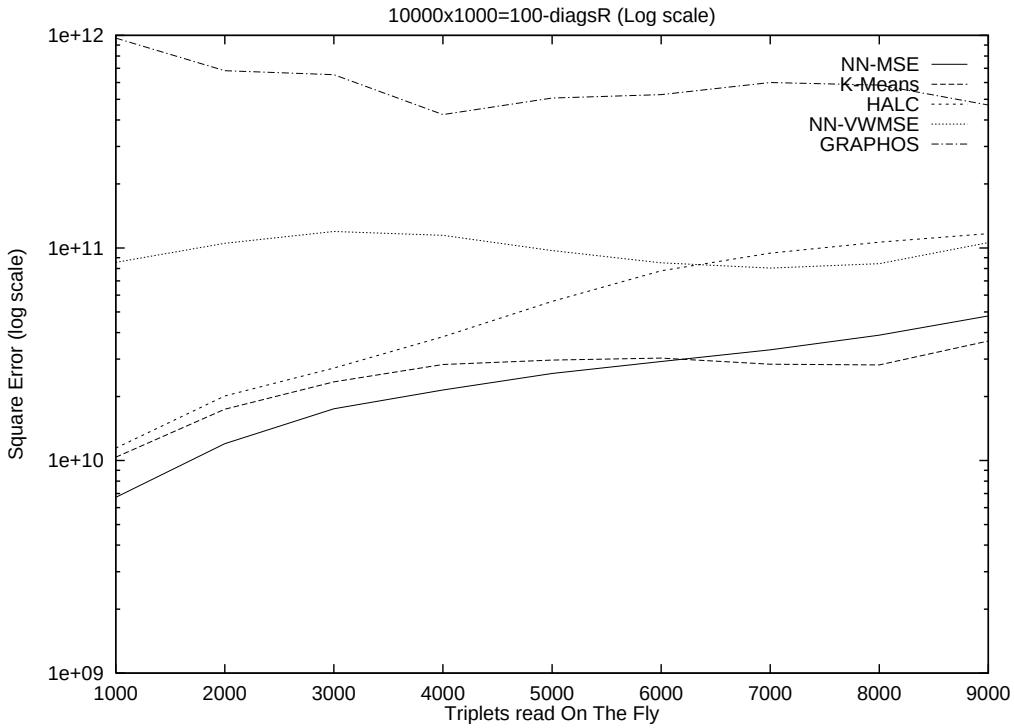


Σχήμα 6.1: Διακύμανση του Τετραγωνικού Σφάλματος για το σύνολο συνθετικών δεδομένων 1000x1000=100:diagsR. Κλίμακα Λογαριθμική.

επίδοση από τους δύο αλγόριθμους ομαδοποίησης ‘‘εν πτήσει’’, τον NN-VWMSE και τον NN-MSE. Αυτό οφείλεται στο ότι οι αλγόριθμοι ομαδικής επεξεργασίας έχουν να αντιμετωπίσουν ένα τεράστιο πλήθος πιθανών διαφορετικών ομαδοποιήσεων, εκ των οποίων αναγκάζονται εκ των πραγμάτων να ελέγχουν κάθε φορά μόνο ένα πολύ μικρό ποσοστό. Αντιθέτως, οι αλγόριθμοι ομαδοποίησης ‘‘εν πτήσει’’, είναι σχεδιασμένοι ώστε να ακολουθούν όσο το δυνατόν περισσότερο την αρχική ομαδοποίηση που είχε χρησιμοποιηθεί για να τους αρχικοποιήσουν. Καθώς στην ομαδοποίηση αυτή το πλήθος των προς ομαδοποίηση δεδομένων ήταν αρκετά μικρό, οι αλγόριθμοι ομαδικής επεξεργασίας που την παρήγαγαν μπόρεσαν να προσεγγίσουν τη βέλτιστη ομαδοποίηση, δίνοντας έτσι ένα πολύ καλό σημείο εκκίνησης στον NN-VWMSE και τον NN-MSE. Οι ίδιοι όμως προσπαθούν κάθε φορά να εξετάζουν όσο το δυνατόν περισσότερες εναλλακτικές λύσεις, δίχως ταυτόχρονα να λαμβάνουν υπόψη τους τις ομαδοποιήσεις στις οποίες είχαν καταλήξει στις προηγούμενες περιπτώσεις, όπου τα δεδομένα ήταν λιγότερα και άρα το πρόβλημα πιο εύκολο. Έτσι, πέφτουν θύματα του μεγέθους του προβλήματος που αυξάνει εκθετικώς και καταλήγουν να παραγάγουν λύσεις που είναι μόνον τοπικώς βέλτιστες.

Τέλος, είναι φανερό ότι η επίδοση του γραφοθεωρητικού αλγόριθμου ομαδοποίησης ‘‘εν πτήσει’’ είναι πολύ χειρότερη όλων των υπολοίπων. Υπάρχουν αρκετοί λόγοι που προκαλούν το φαινόμενο αυτό. Πρώτα πρέπει να αναφερθεί ότι ο αλγόριθμος μελετήθηκε πολύ λιγότερο εν σχέσει προς τους υπόλοιπους και ως εκ τούτου είναι πιθανόν οι τιμές των παραμέτρων που χρησιμοποιήθησαν κατά την εκτέλεσή του να απέχουν από τις βέλτιστες. Επίσης, ο αλγόριθμος αυτός παράγει συνήθως ένα πολύ μικρότερο πλήθος ομάδων (οι ακριβής αριθμοί παρουσιάζονται στο παράρτημα ΣΤ). Έτσι, είναι δύσκολο να εμφανίσει καλή επίδοση, καθώς προφανώς ομαδοποιεί τριάδες που ανήκουν σε διαφορετικές ομάδες. Ο πιο σημαντικός, όμως, λόγος της άσχημης επίδοσής του είναι το γεγονός ότι

η ποσότητα που προσπαθεί να ελαχιστοποιήσει δεν σχετίζεται με το τετραγωνικό σφάλμα, όπως συμβαίνει με τους υπόλοιπους, οπότε είναι τελικώς λογική η κακή του επίδοση συμφώνως προς αυτό το μέτρο.

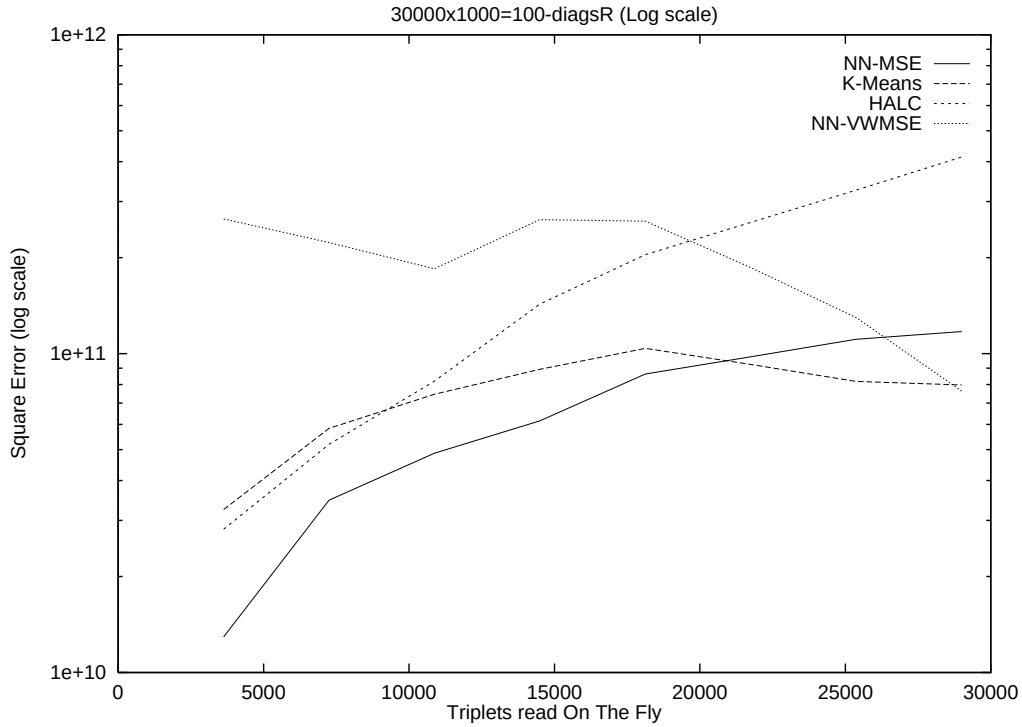


Σχήμα 6.2: Διακύμανση του Τετραγωνικού Σφάλματος για το σύνολο συνθετικών δεδομένων 10000x1000=100:diagsR. Κλίμακα Λογαριθμική.

Εξετάζοντας τώρα τα αποτελέσματα του δεύτερου πειράματος με το σύνολο δεδομένων 10000x1000=100:diagsR που παρουσιάζονται στο σχήμα 6.2, είναι προφανές ότι οι παρατηρήσεις που έγιναν για τα αποτελέσματα του προηγούμενου πειράματος συνεχίζουν να ισχύουν. Επιπλέον, εδώ φαίνεται πιο καθαρά η τάση, του μεν NN-MSE να οδηγεί προς μία ομαδοποίηση με μεγαλύτερο συνεχώς τετραγωνικό σφάλμα, του δε NN-VWMSE προς μία με συνεχώς μικρότερο. Εξαιτίας του φαινομένου αυτού, αποφασίσθηκε τελικώς και η διεξαγωγή του τρίτου και τελευταίου πειράματος με συνθετικά δεδομένα, δηλαδή με το σύνολο δεδομένων 30000x1000=100:diagsR, προκειμένου να φανεί εάν με την πάροδο του χρόνου και με την εμφάνιση περισσότερων ακόμα τριάδων, οι δύο αυτοί αλγόριθμοι θα αλλάξουν τελικώς συμπεριφορά.

Στο σχήμα 6.3 παρουσιάζονται τα αποτελέσματα των πειραμάτων που έγιναν με το σύνολο δεδομένων 30000x1000=100:diagsR. Στο σχήμα αυτό, δεν περιλαμβάνεται ο γραφοθεωρητικός αλγόριθμος, καθώς ο χρόνος εκτέλεσής του ήταν απαγορευτικός για το συγκεκριμένο πλήθος δεδομένων. Επίσης, ήδη από τα προηγούμενα πειράματα, είχε φανεί ότι η επίδοσή του ήταν κατά πολύ χειρότερη των υπολοίπων αλγόριθμων, οπότε δεν υπήρχε λόγος περαιτέρω μελέτης του.

Στο πείραμα αυτό φαίνεται ότι τελικώς ο αλγόριθμος NN-VWMSE καταφέρνει να μειώσει αισθητά το τετραγωνικό σφάλμα στην ομαδοποίηση που παράγει, ξεπερνώντας στο τέλος όλους τους άλλους αλγόριθμους. Αντιθέτως, ο NN-MSE παράγει ομαδοποιήσεις με ολοένα μεγαλύτερο τετραγωνικό σφάλμα και υπερβαίνει στο τέλος τόσο τον NN-VWMSE όσο και τον K-MEANS.



Σχήμα 6.3: Διακύμανση του Τετραγωνικού Σφάλματος για το σύνολο συνθετικών δεδομένων 30000x1000=100:diagsR. Κλίμακα Λογαριθμική.

Η συμπεριφορά του NN-VWMSE, που συνεχώς αυξάνει την επίδοσή του από ένα σημείο και μετά, φαίνεται εκ πρώτης όψεως να έρχεται σε αντίθεση με την παρατήρηση που είχε γίνει ανωτέρω, ότι καθώς ο αριθμός των προς ομαδοποίηση τριάδων αυξάνει, θα αυξάνει και το τετραγωνικό σφάλμα της εκάστοτε ομαδοποίησης. Η εξήγηση όμως του φαινομένου αυτού είναι απλή: ο NN-VWMSE προσπαθεί να δημιουργήσει ομάδες με σχετικώς ίσα πλήθη στοιχείων. Καθώς όμως οι τριάδες έχουν αποθηκευτεί ταξινομημένες ως προς την ομάδα τους, για ένα μεγάλο χρονικό διάστημα αναγκάζεται να αγνοεί τις πραγματικές τους ομάδες και να τις τοποθετεί σε άλλες, με μικρό ως εκείνη τη στιγμή πλήθος στοιχείων, ώστε να τις ισοκατανείμει. Έτσι, σε όλα τα μέχρις στιγμής πειράματα, παρουσιάζει αρχικώς μία πολύ χειρότερη επίδοση από τους υπόλοιπους αλγόριθμους, την οποία βελτιώνει, καθώς περνά ο χρόνος και εμφανίζονται στο σύστημα τριάδες από όλες τις ομάδες. Αξίζει εδώ να σημειωθεί ότι αυτού του είδους η ταξινόμηση των δεδομένων είναι η χειρότερη για τον συγκεκριμένο αλγόριθμο. Παρόλα ταύτα, τελικώς καταφέρνει να επιτύχει την καλύτερη επίδοση, όπως φαίνεται και στο σχήμα 6.3.

6.3 Πραγματικά δεδομένα

Ακολούθως των πειραμάτων με τα σύνολα συνθετικών δεδομένων, χρησιμοποιήθησαν σύνολα πραγματικών δεδομένων που είχαν ληφθεί από την **ιχνοληψία** (*tracing*) μίας πραγματικής εφαρμογής βάσης δεδομένων σε ένα σύστημα επεξεργασίας δοσοληψιών της εταιρείας Siemens Nixdorf Informationssysteme AG [SNI]. Αυτά μας παρασχέθησαν από την εν λόγω εταιρεία στα πλαίσια του ερευνητικού προγράμματος LYDIA [ESP], στο οποίο συμμετέχει τόσο αυτή όσο και η ομάδα Παράλληλων και Κατανεμημένων Συστημάτων του Ινστιτούτου Πληροφορικής, του Ιδρύματος Τεχνολογίας και Έρευνας. Τα σύνολα αυτά είναι δύο τον

αριθμό' το σύνολο PULS και το σύνολο DOA.

Τα σύνολα αυτά μετατράπησαν στη μορφή εισόδου που λαμβάνει το CLUE (βλέπε το παράρτημα Γ για την περιγραφή της μορφής εισόδου του CLUE) μέσω ενός προγράμματος φίλτρου, το οποίο παραλλήλως με την μετατροπή αυτή καθ' αυτή, συγκέρασε τις προσπελάσεις στη βάση δεδομένων όλων των ξεχωριστών **περιπτώσεων** (*instances*) των τριάδων, προκειμένου να ληφθεί ένας συγκεντρωτικός πίνακας προσπελάσεων.

Θα πρέπει να σημειωθεί εδώ η παρατήρηση του Αλέξανδρου Λαμπρινίδη [Lab95], ότι τα δύο αυτά σύνολα έχουν πολύ χαμηλή πυκνότητα προσπελάσεων, όπου η πυκνότητα προσπελάσεων ορίζεται ως:

$$\text{πυκνότητα προσπελάσεων} = \frac{\text{πλήθος μη μηδενικών προσπελάσεων}}{\text{πλήθος τριπλετών} \times \text{πλήθος σελίδων βάσης δεδομένων}} \tag{6.2}$$

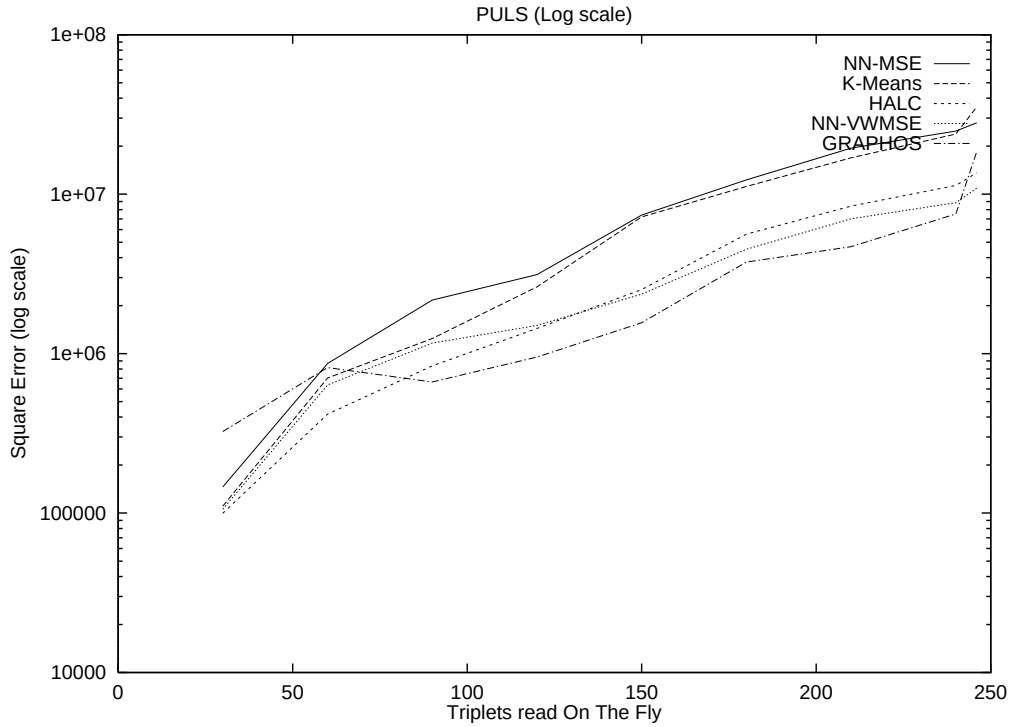
Συγκεκριμένα, το σύνολο PULS έχει τα κάτωθι χαρακτηριστικά:

- **53** διαφορετικά αναγνωριστικά προγραμμάτων
- **103** διαφορετικά αναγνωριστικά χρηστών
- **110** διαφορετικά αναγνωριστικά τερματικών
- **280** τριάδες
- **66846** σελίδες βάσης
- **178749** μη μηδενικές προσπελάσεις συνολικώς
- πυκνότητα προσπελάσεων: **0,95501%**

Αντιστοίχως, το σύνολο DOA έχει τα κάτωθι χαρακτηριστικά:

- **103** διαφορετικά αναγνωριστικά προγραμμάτων
- **474** διαφορετικά αναγνωριστικά χρηστών
- **474** διαφορετικά αναγνωριστικά τερματικών
- **1984** τριάδες
- **41003** σελίδες βάσης
- **180017** μη μηδενικές προσπελάσεις
- πυκνότητα προσπελάσεων: **0,22123%**

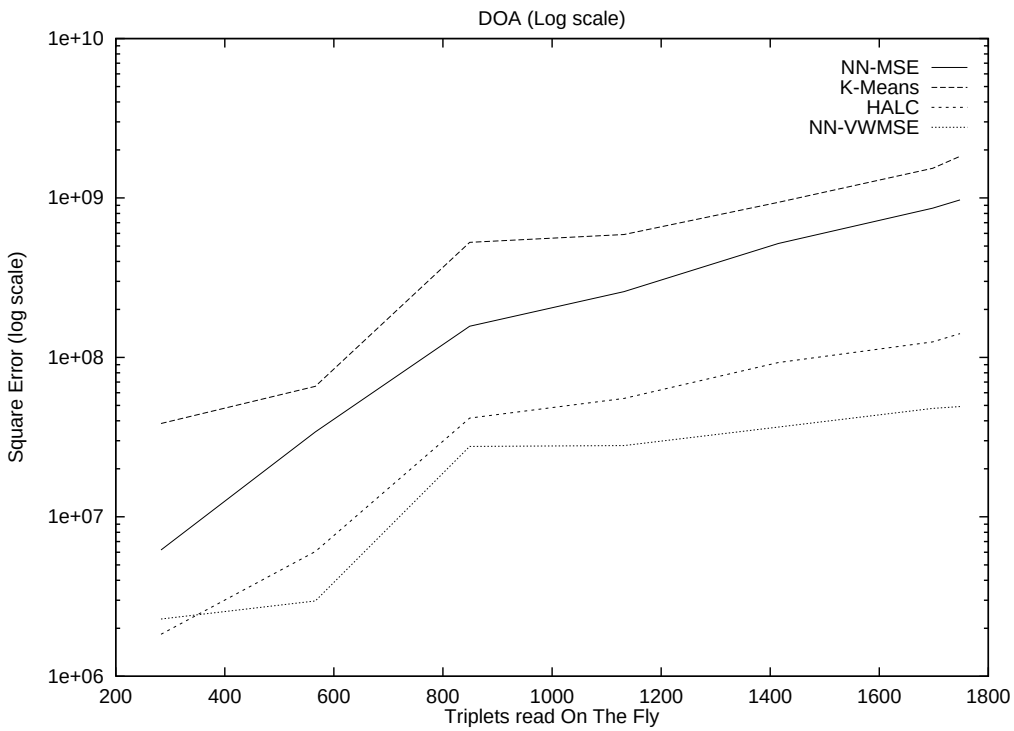
Στο σχήμα 6.4 παρουσιάζονται τα αποτελέσματα των πειραμάτων με το σύνολο πραγματικών δεδομένων PULS, σε λογαριθμική κλίμακα. Καθώς σε αυτό, οι τριάδες δεν εμφανίζονται στη σειρά με βάση την ομάδα τους όπως στα συνθετικά δεδομένα, οι αλγόριθμοι εμφανίζουν διαφορετική συμπεριφορά, ιδίως ο HALC και ο NN-VWMSE. Έτσι, φαίνεται ότι ο NN-MSE έχει σχεδόν την ίδια συμπεριφορά με τον K-MEANS, γεγονός που ήταν αναμενόμενο, δεδομένου ότι ο NN-MSE είναι στην πραγματικότητα η υλοποίηση του K-MEANS με ένα νευρωνικό δίκτυο. Επίσης ο NN-VWMSE καταφέρνει να πετύχει τη βέλτιστη επίδοση, καθώς τώρα η σειρά εμφάνισης των τριάδων προσεγγίζει καλύτερα το μοντέλο που αυτός θεωρεί ότι ισχύει.



Σχήμα 6.4: Διακύμανση του Τετραγωνικού Σφάλματος για το σύνολο πραγματικών δεδομένων PULS. Κλίμακα Λογαριθμική.

Καθότι το πλήθος των δεδομένων ήταν σχετικώς μικρό, στάθηκε δυνατή και η αξιολόγηση του γραφοθεωρητικού αλγόριθμου ομαδοποίησης ‘‘εν πτήσει’’. Παρόλο που αυτός φαίνεται να επιτυγχάνει τη βέλτιστη επίδοση στο σχήμα 6.4, θα πρέπει να σημειωθεί ότι το πλήθος των ομάδων που αναγνωρίζει είναι κατά πολύ μεγαλύτερο από αυτό των υπολοίπων. Έτσι, ενώ οι υπόλοιποι αλγόριθμοι αναγνωρίζουν περί τις 30 ομάδες, ο γραφοθεωρητικός αλγόριθμος ομαδοποίησης ‘‘εν πτήσει’’ αναγνωρίζει έως και 167 ομάδες (το πλήθος των ομάδων παρουσιάζεται αναλυτικότερα στον πίνακα ΣΤ.4 του παραρτήματος ΣΤ). Επομένως, είναι προφανές ότι η υψηλή του επίδοση οφείλεται αποκλειστικώς στο μεγάλο πλήθος των ομάδων που έχει κατασκευάσει και όχι στον καλό διαχωρισμό των δεδομένων αυτών καθ’ αυτών. Ως εκ τούτου και δεδομένου του ότι είναι σχετικώς αργός, ο γραφοθεωρητικός αλγόριθμος δεν χρησιμοποιήθηκε στο τελευταίο πείραμα με το σύνολο πραγματικών δεδομένων DOA.

Μελετώντας τώρα και τα αποτελέσματα των πειραμάτων με το σύνολο πραγματικών δεδομένων DOA, που παρουσιάζονται στο σχήμα 6.5, ενισχύονται τα συμπεράσματα που εξίχθησαν από τα πειράματα με τα σύνολα συνθετικών δεδομένων, καθώς και με το σύνολο PULS. Δηλαδή, ο NN-VWMSE δίνει την καλύτερη ομαδοποίηση, καλύτερη ακόμα και από τους αλγόριθμους ομαδικής επεξεργασίας K-MEANS και HALC.



Σχήμα 6.5: Διακύμανση του Τετραγωνικού Σφάλματος για το σύνολο πραγματικών δεδομένων DOA. Κλίμακα Λογαριθμική.

Κεφάλαιο 7

Συμπεράσματα - Μελλοντικές Κατευθύνσεις

Στο κεφάλαιο αυτό παρατίθεται μία σύνοψη των συμπερασμάτων της παρούσας εργασίας. Παρουσιάζονται επίσης ορισμένα θέματα που θα είχε ενδιαφέρον να μελετηθούν στο μέλλον.

7.1 Χρήση των Αλγόριθμων σε Πραγματικά Συστήματα

Από τα αποτελέσματα των τελικών πειραμάτων (βλέπε το κεφάλαιο 6) είναι προφανές ότι στις περισσότερες, αν όχι σε όλες, τις περιπτώσεις που πρέπει να ομαδοποιηθούν πραγματικά δεδομένα, ο αλγόριθμος που θα πρέπει να χρησιμοποιηθεί είναι ο NN-VWMSE. Τα πειράματα έδειξαν ότι καταφέρνει πάντοτε να έχει την καλύτερη επίδοση. Εάν όμως σε κάποιο σύστημα είναι πιθανόν να εμφανίζονται όμοιες τριάδες για μακρό χρονικό διάστημα, όπως συνέβαινε στα τεχνητώς κατασκευασμένα δεδομένα, τότε θα πρέπει να ληφθεί υπόψη ότι ο NN-VWMSE θα χρειαστεί κάποια περίοδο προσαρμογής προτού αρχίσει να παράγει καλύτερες ομαδοποιήσεις από τον NN-MSE.

Δεδομένης επίσης της ανάγκης για πολλή γρήγορη επεξεργασία των δοσοληψιών που εμφανίζονται, θα πρέπει να εξετασθεί σοβαρά και η παραλληλοποίηση του αλγόριθμου που θα επιλεχθεί τελικώς. Το ακόλουθο υπο-υποκεφάλαιο αναφέρεται ειδικώς στο θέμα αυτό.

7.1.1 Παραλληλοποίηση των Αλγόριθμων

Οι αλγόριθμοι ομαδοποίησης ‘‘εν πτήσει’’ που υλοποιήθησαν με νευρωνικά δίκτυα, έχουν την πολύ ενδιαφέρουσα ιδιότητα ότι μπορούν να παραλληλοποιηθούν προκειμένου να γίνουν ακόμα πιο γρήγοροι. Έτσι, εάν χρησιμοποιείται ως **εξυπηρετητής** (*server*) κάποιο μηχάνημα με περισσότερες της μίας **κεντρικής μονάδας επεξεργασίας** (*CPU*), είναι δυνατόν να αντιστοιχηθεί ένας ή και περισσότεροι νευρώνες ανά μία κεντρική μονάδα επεξεργασίας. Καθώς κάθε νευρώνας δουλεύει ανεξάρτητα από τους υπόλοιπους, η παραλληλοποίηση αυτή θα έχει ως αποτέλεσμα μία αρκετά μεγάλη βελτίωση του συνολικού χρόνου που απαιτείται για την εύρεση του νευρώνα που αντιπροσωπεύει την ομάδα της εκάστοτε τριάδας που εμφανίζεται στο σύστημα.

Η επιτάχυνση αυτή οφείλεται στο ότι ο υπολογισμός της απόστασης της τριάδας από το κέντρο της ομάδας που αντιστοιχεί σε κάθε νευρώνα, μπορεί να γίνει ανεξάρτητα από τους υπόλοιπους υπολογισμούς και δίχως να μεταβάλλει κάποια δεδομένα που χρησιμοποιούνται σε άλλους υπολογισμούς. Έτσι, δεν χρειάζεται να γίνει χρήση **κλειδαριών** (*locks*) ή οποιουδήποτε άλλου είδους άμεσης ή έμμεσης επικοινωνίας. Η επικοινωνία μπορεί να

περιοριστεί στο τελικό στάδιο μόνο, όπου απαιτείται η εύρεση της ελάχιστης εκ των K συνολικώς αποστάσεων. Ήδη υπάρχουν πολλοί γνωστοί τρόποι με τους οποίους μπορεί να επιτευχθεί η γρήγορη υλοποίηση αυτού του τμήματος του παράλληλου αλγόριθμου.

Πιο συγκεκριμένα, έστω ότι K είναι ο αριθμός των ομάδων και D η διάσταση των τριάδων - δεδομένων. Τότε για τον υπολογισμό της απόστασης ενός δεδομένου από τους νευρώνες θα πρέπει να γίνουν για κάθε νευρώνα:

- D αφαιρέσεις
- D+1 πολλαπλασιασμοί (D υψώσεις στο τετράγωνο + 1 πολλαπλασιασμός με τη διασπορά της ομάδας)
- D-1 προσθέσεις

Θεωρώντας το κόστος της πρόσθεσης και της αφαίρεσης ίδιο και ίσο με C_{sum} και του πολλαπλασιασμού ίσο με C_{mul}, τότε το κόστος για τη σειριακή υλοποίηση θα είναι αυτό της εξίσωσης 7.1:

$$C_{Serial} = K * [(2 * D - 1) * C_{sum} + (D + 1) * C_{mul}] \overset{D>>1}{\approx} K * D * (2 * C_{sum} + C_{mul}) \quad (7.1)$$

Αντιστοίχως, για μία παράλληλη υλοποίηση που κάνει χρήση N επεξεργαστών ($N \leq K$), το κόστος θα είναι αυτό της εξίσωσης 7.2:

$$C_{Parallel} = \frac{K}{N} * [(2 * D - 1) * C_{sum} + (D + 1) * C_{mul}] \overset{D>>1}{\approx} \frac{K}{N} * D * (2 * C_{sum} + C_{mul}) \quad (7.2)$$

Επομένως, η **επιτάχυνση** (*speedup*) που θα επιτευχθεί θα είναι:

$$speedup = \frac{C_{Serial}}{C_{Parallel}} \approx N \quad (7.3)$$

Αξίζει εδώ να σημειωθεί ότι είναι πιθανόν να παρατηρηθεί ακόμα και το φαινόμενο της **υπέρ - γραμμικής επιτάχυνσης** (*overlinear speedup*), καθώς αν τα διανύσματα των βαρών των νευρώνων χωρούν στις **κρυφές μνήμες** (*caches*) των επεξεργαστών, τότε η παράλληλη υλοποίηση θα χρησιμοποιεί πολύ λιγότερο την κυρίως μνήμη του υπολογιστή, εν αντιθέσει προς την τρέχουσα σειριακή υλοποίηση.

7.2 Εύρεση Βέλτιστου Αλγόριθμου

Ένα πολύ ευαίσθητο σημείο, που παρόλα αυτά αξίζει να αναφερθεί, είναι το κατά πόσον αξίζει να μελετηθούν πιο πολύπλοκοι αλγόριθμοι στον τομέα αυτό της ομαδοποίησης ‘‘εν πτήσει’’.

Παρόλα τα πολύ καλά αποτελέσματα που ελήφθησαν από τους συγκεκριμένους αλγόριθμους, είναι αρκετά πιθανόν να υπάρχει κάποιος άλλος αλγόριθμος ο οποίος να έχει ακόμα καλύτερη επίδοση.

Ένας τέτοιος αλγόριθμος όμως, αναγκαστικώς θα είναι πιο πολύπλοκος από την υλοποίηση του K-MEANS με ένα νευρωνικό δίκτυο. Αυτό, αυτομάτως σημαίνει ότι θα εκτελεί περισσότερες εντολές προκειμένου να ομαδοποιήσει κάθε τριάδα που εμφανίζεται στο σύστημα. Επίσης είναι λογικό να υποθέσουμε ότι την αυξημένη επίδοσή του θα την οφείλει εν μέρει και στη χρησιμοποίηση στοιχείων από όλες τις ομάδες ή από κάποιο υποσύνολο

αυτών σε κάθε βήμα. Κάτι τέτοιο όμως σημαίνει ότι σε τυχόν παράλληλη υλοποίησή του θα είναι απαραίτητο να μεταφερθούν τα στοιχεία αυτά από τον ένα επεξεργαστή στον άλλο, μειώνοντας έτσι αρκετά την επιτάχυνση που δίνει η παραλληλοποίηση.

Τέλος, ένα αρκετά σημαντικό πλεονέκτημα του K-MEANS είναι ότι δύναται να χρησιμοποιήσει την ομαδοποίηση που παράγουν οι αλγόριθμοι ομαδικής επεξεργασίας προκειμένου να αρχικοποιήσει τους νευρώνες, αποφεύγοντας έτσι την συνήθως χρονοβόρα διαδικασία της εκπαίδευσης των νευρωνικών δικτύων. Αυτό είναι δυνατόν ακριβώς χάρη στην απλότητα τόσο του ίδιου όσο και των δομών που χρησιμοποιεί για να αποθηκεύσει την υπάρχουσα πληροφορία. Εάν πρόκειται να χρησιμοποιηθεί στη θέση του κάποιος πιο πολύπλοκος αλγόριθμος, το πλεονέκτημα αυτό είναι πολύ πιθανό να χαθεί. Στην περίπτωση αυτή, ο χρήστης ενός πραγματικού συστήματος θα αναγκάζεται να αγνοεί πλέον την ομαδοποίηση που παράγουν οι αλγόριθμοι ομαδικής επεξεργασίας για τα διαθέσιμα προς εκπαίδευση δεδομένα και να εκπαιδεύει το νευρωνικό δίκτυο από την αρχή.

Ως εκ τούτου, είναι άποψή μας ότι η μελέτη πολυπλοκοτέρων αλγόριθμων στο συγκεκριμένο θέμα θα είχε μόνον ερευνητικό ενδιαφέρον, πλην όμως ελάχιστα θα μπορούσε να προσφέρει επιπλέον στην πρακτική χρήση ενός συστήματος.

7.3 Μελλοντικές Κατευθύνσεις

Αντί να γίνει προσπάθεια εύρεσης καλύτερων αλγόριθμων, θα είχε ενδιαφέρον να παρθούν άλλες κατευθύνσεις, στη συνέχεια της εργασίας αυτής.

Αρχικώς, θα άξιζε να γίνει μία μελέτη των αποτελεσμάτων του CLUE, ενώ αυτό χρησιμοποιείται, είτε από κάποιο προσομοιωτή ενός συστήματος επεξεργασίας δοσοληψιών, όπως είναι το TPsim [Mar95], ή από ένα πραγματικό σύστημα. Δηλαδή, για κάθε δοσοληψία που εμφανίζεται στο σύστημα επεξεργασίας δοσοληψιών να ζητείται από το CLUE η ομάδα που ανήκει αυτή και η πληροφορία αυτή να χρησιμοποιείται για την εξισορρόπηση του φόρτου των διαφόρων κόμβων και τη δρομολόγηση της δοσοληψίας προς τον καταλληλότερο κόμβο. Έτσι, θα σταθεί δυνατό να μετρηθεί κατά πόσο η γνώση της ομάδας στην οποία ανήκουν οι διάφορες δοσοληψίες βοηθά πραγματικά το τελικό σύστημα.

Το θέμα αυτό έχει ήδη αρχίσει να μελετάται στην ερευνητική ομάδα των Πλειάδων. Συγκεκριμένα, η Μαρία Καραβασίλη και ο Μανώλης Μαραζάκης έχουν γράψει κώδικα που επιτρέπει στο CLUE να καταγράφει μία, όσο το δυνατόν πληρέστερη, εικόνα του συστήματος, όσον αφορά τις δοσοληψίες που εμφανίσθηκαν σε αυτό, καθώς και κάποια άλλα στοιχεία, τα οποία το TPsim τα διαβάζει και τα χρησιμοποιεί κατά την προσομοίωση της λειτουργίας του κατανεμημένου συστήματος επεξεργασίας δοσοληψιών.

Αυτή τη στιγμή δεν έχει γίνει ακόμα πλήρης διασύνδεση των δύο εργαλείων, δηλαδή το CLUE δεν μπορεί να επικοινωνεί δυναμικώς με το TPsim. Απλώς το δεύτερο διαβάζει την *τελική* ομαδοποίηση και χρησιμοποιεί αυτή. Έτσι, δεν είναι δυνατόν ακόμα να μελετηθεί η επίδραση των αλγόριθμων ομαδοποίησης ‘‘εν πτήσει’’, παρά μόνον αν θεωρηθούν ως αλγόριθμοι ομαδικής επεξεργασίας και ληφθεί υπόψη μόνο η τελική ομαδοποίηση που παράγουν.

Από τη στιγμή που θα έχει ολοκληρωθεί η διασύνδεση αυτή, θα είχε ενδιαφέρον να ξεκινήσει μία μελέτη διαφορετικών αλγόριθμων εξισορρόπησης του φόρτου εργασίας και δρομολόγησης των δοσοληψιών, οι οποίοι να λαμβάνουν υπόψη τους τις ομάδες των δοσοληψιών.

Παράρτημα Α

Ισοδυναμία των Συναρτήσεων VWMSE και MSE

Στο παράρτημα αυτό παρατίθεται η απόδειξη της πρότασης ότι, όταν ο αριθμός K των ομάδων που δημιουργούνται είναι μεγάλος και η κατανομή P των δεδομένων $\vec{x}$ είναι λεία, τότε η βέλτιστη διαμέριση και το σύνολο των βέλτιστων διανυσμάτων αναφοράς (κέντρων των ομάδων) που ελαχιστοποιούν τη συνάρτηση κόστους VMSE και αυτά που ελαχιστοποιούν τη συνάρτηση κόστους MSE είναι τα ίδια.

Η απόδειξη αυτή έχει αντιγραφεί από το [CS95].

Οι ορισμοί των συναρτήσεων VWMSE και MSE, όπως είδαμε και στο κεφάλαιο 3, στις εξισώσεις 3.8 και 3.2, είναι οι κάτωθι:

$$\text{VWMSE}(K) = \sum_{i=1}^{K} v_i^2 \tag{A.1}$$

$$\text{MSE}(K) = \sum_{i=1}^{K} v_i \tag{A.2}$$

Στις εξισώσεις Α.1 και Α.2, v_i είναι η διασπορά της i-οστής ομάδας (βλέπε την εξίσωση 3.1).

Έστω I^* η βέλτιστη διαμέριση των δεδομένων που ελαχιστοποιεί τη συνάρτηση κόστους MSE. Θα δείξουμε ότι η διαμέριση αυτή ελαχιστοποιεί και τη συνάρτηση VWMSE.

Αν $v_1^*, \ldots, v_K^*$ είναι οι διασπορές των ομάδων της βέλτιστης διαμέρισης I^*, τότε θα έχουμε:

$$\sum_{i=1}^{K} v_i \geq \sum_{i=1}^{K} v_i^* \tag{A.3}$$

Για οποιεσδήποτε τιμές των $v_1, \ldots, v_K$ είναι γνωστό ότι ισχύει:

$$\frac{1}{K} \sum_{i=1}^{K} v_i^2 \geq (\frac{1}{K} \sum_{i=1}^{K} v_i)^2 \tag{A.4}$$

,όπου η ισότητα ισχύει όταν και μόνον όταν $v_1 = \ldots = v_K$.

Αντικαθιστώντας την Α.3 στην Α.4, λαμβάνουμε:

$$\frac{1}{K} \sum_{i=1}^{K} v_i^2 \geq (\frac{1}{K} \sum_{i=1}^{K} v_i^*)^2 \tag{A.5}$$

Στην εργασία [Ger79] αποδείχθηκε ότι για ασυμπτωτικά μεγάλο K και για λεία κατανομή P των δεδομένων, οι διασπορές των ομάδων, $v_1^*, \ldots, v_K^*$, της βέλτιστης διαμέρισης I^* ικανοποιούν τη σχέση:

$$v_1^* = \ldots = v_K^* = v^* \tag{A.6}$$

Συνδυάζοντας τις σχέσεις A.5 και A.6, λαμβάνουμε:

$$\frac{1}{K}\sum_{i=1}^{K} v_i^2 \geq (v^*)^2 \tag{A.7}$$

Η τελευταία σχέση μπορεί να γραφτεί και ως:

$$\sum_{i=1}^{K} v_i^2 \geq K(v^*)^2 \tag{A.8}$$

Καθώς, η ποσότητα $K(v^*)^2$ είναι η τιμή της VWMSE για τη διαμέριση I^*, η σχέση A.8 αποδεικνύει ότι η διαμέριση I^* ελαχιστοποιεί και τη συνάρτηση VWMSE.

Έστω, τώρα, I^{**} η βέλτιστη διαμέριση που ελαχιστοποιεί τη συνάρτηση VWMSE. Για να είναι πλήρης η απόδειξη της ισοδυναμίας των VWMSE και MSE, θα πρέπει να δείξουμε ότι η διαμέριση I^{**} ελαχιστοποιεί και τη MSE.

Έστω ότι η διαμέριση I^* που ελαχιστοποιεί τη MSE είναι διαφορετική από την I^{**}. Τότε, θα έχουμε:

$$\text{MSE}(I^{**}) > \text{MSE}(I^*) \tag{A.9}$$

Καθώς έχουμε ήδη δείξει ότι η I^* θα ελαχιστοποιεί και τη συνάρτηση VWMSE εφόσον ελαχιστοποιεί τη MSE, η σχέση A.9 οδηγεί στο συμπέρασμα ότι:

$$\text{VWMSE}(I^{**}) > \text{VWMSE}(I^*) \tag{A.10}$$

Αυτή η τελευταία σχέση όμως, αντιτίθεται στην αρχική μας υπόθεση ότι η I^{**} ήταν η βέλτιστη διαμέριση για τη συνάρτηση VWMSE. Επομένως, οι διαμερίσεις I^* και I^{**} πρέπει να είναι ίδιες.

Παράρτημα Β

Αλγόριθμοι Κατασκευής του Ελάχιστου Ζευγνύοντος Δένδρου

Στο παράρτημα αυτό παρουσιάζονται δύο αλγόριθμοι κατασκευής του ελάχιστου ζευγνύοντος δένδρου ενός γράφου. Οι αλγόριθμοι αυτοί είναι ο αλγόριθμος του Kruskal [Kru56] και ο αλγόριθμος του Prim [Pri57].

Μία πολλή καλή παρουσίαση και μελέτη των αλγόριθμων αυτών γίνεται στο [Cor89], ενώ στο [Tar83a] γίνεται μία ανασκόπηση της βιβλιογραφίας και παρατίθεται ένα αρκετά προχωρημένο υλικό πάνω στο πρόβλημα. Τέλος, στο [GH85] γίνεται μία ιστορική αναδρομή του προβλήματος αυτού.

Το πρόβλημα ορίζεται ως εξής:

Έστω γράφος, Γ, που αποτελείται από ένα σύνολο κόμβων, K, και ένα σύνολο ακμών, A, που συνδέουν τους κόμβους μεταξύ τους. Θεωρώντας ότι κάθε ακμή φέρει ένα βάρος, που υποδηλώνει την απόσταση των κόμβων που συνδέει, ζητείται το σύνολο εκείνο των ακμών που αφενός μεν συνδέει όλους τους κόμβους, αφετέρου δε, έχει το ελάχιστο συνολικό άθροισμα βαρών των ακμών του.

Είναι εύκολο να αποδειχτεί ότι το ζητούμενο σύνολο των ακμών θα αποτελεί ένα δένδρο και θα έχει τόσες ακμές όσες ο αριθμός των κόμβων (ν) του γράφου μείον ένα.

Ο αλγόριθμος του Kruskal ξεκινά έχοντας ταξινομήσει όλες τις ακμές του γράφου κατά αύξουσα σειρά συμφώνως προς το βάρος τους. Κατόπιν επιλέγει ακμές στη σειρά προσέχοντας η ακμή που επιλέγει κάθε φορά να μη δημιουργεί κάποιο κύκλο στο γράφο. Σταματά όταν έχει επιλέξει $\nu - 1$ ακμές, όπου ν είναι ο αριθμός των κόμβων του γράφου.

Ο αλγόριθμος του Prim ξεκινά από έναν οποιοδήποτε κόμβο του γράφου και προσθέτει την ακμή εκείνη, από αυτές που συνδέονται στον κόμβο, η οποία έχει το ελάχιστο βάρος. Η ακμή αυτή και τα δύο άκρα της αποτελούν το δένδρο T_1. Το δένδρο T_μ παράγεται από το δένδρο $T_{\mu-1}$ προσθέτοντας σε αυτό την ακμή με το μικρότερο βάρος που συνδέει το $T_{\mu-1}$ με κάποιον κόμβο του γράφου που δεν συμπεριλαμβάνεται ακόμα στο $T_{\mu-1}$. Η διαδικασία αυτή συνεχίζεται έως ότου καταλήξει ο αλγόριθμος στο ζητούμενο δένδρο $T_{\nu-1}$.

Έχει αποδειχθεί ότι και οι δύο αλγόριθμοι έχουν πολυπλοκότητα $O(A * \ln(K))$, όμως ο αλγόριθμος του Prim μπορεί να αποκτήσει, με τη χρήση **σωρών** (*heaps*) του Fibonacci, πολυπλοκότητα $O(A + K\ln(K))$. Αν ο αριθμός $|K|$ των κόμβων είναι πολύ μικρότερος του αριθμού των ακμών $|A|$, τότε αυτή η παραλλαγή του αλγόριθμου του Prim καταλήγει να είναι ταχύτερη.

Παράρτημα Γ

Interacting with CLUE

This appendix describes how one can use CLUE. The information herein is derived from the information contained in Alexandros Labrinidis' Master thesis [Lab95]. Additions, enhancements, etc. are marked accordingly.

Γ.1 CLUE command line options

CLUE accepts the following command line options:

- `-in` *Mandatory*

  Must be followed by a filename argument, which should contain a description of transaction triplets in CLUE format.

- `-out` *Mandatory*

  Must be followed by a filename argument, where output will be stored.

- `-par` *Mandatory*

  Must be followed by a filename argument, which should contain settings of parameters and commands to be executed by CLUE.

- `-fly` *Optional*

  Must be followed by a filename argument, which should contain a description of transaction triplets in CLUE format. These transaction triplets will be used as input to one of the on--the--fly clustering algorithms.

- `-msg` *Optional*

  Must be followed by a filename argument, where messages and other additional information will be stored.

- `-TPsim` *Optional*

  Must be followed by a positive integer argument, which should be the number of nodes that will be used in the experiments with TPsim. CLUE needs to know this, since it uses it for distributing the Data Base pages to the system nodes.

- `-Verbose` *Optional*

  Used to instruct CLUE to enter verbose mode, in which it produces additional messages, informing the user on the particular command it is currently executing.

The following two sections describe the format of the file where the transaction triplets are stored (argument to the `-in` / `-fly` options) and the format of the script file that is used to set parameters and give commands to CLUE for execution (argument to the `-par` option) respectively.

Γ.2 CLUE Reference Input File

```
H CLOU_sample_input
# The header line has to be the first line. THIS is a comment.
# The header line string must not contain any white-space characters.
# The header line string's maximum length is 40 characters.
# Maximum line length is 80 characters.
# -------1---------2---------3---------4---------5---------6---------7---------8
#
# Note: The order of the following SECTIONS is mandatory. The order of the
# ----- records WITHIN the sections is immaterial.
#
#
# ****************************************************************************
# * Global Section (record_type = G):                                        *
# ****************************************************************************
# Record Format: "G <what_string> = <how_many_string> [comment]".
#
# The global section is the first section after the header line.
# Within this section, the order of the entries is immaterial.
# All global lines are mandatory.
#
G interval_length = 1000        Length of observation interval (seconds).
G no_user_ids = 3       Number of user-IDs (first component of WL class id).
G no_term_ids = 4    Number of terminal-IDs (second component of WL class id).
G no_tran_ids = 5   Number of transaction IDs (third component of WL class id).
G no_workload_class_elements = 12     Number of WL class elements (triplets).
G no_workload_classes = 3              Number of workload classes.
G no_data_class_elements = 10 Number of data class elem. (initial data buckets).
G no_data_classes = 10                 Number of data classes.
#     >>> Note: This list may be extended in the near future!. <<<
#
# ****************************************************************************
# * List of USER_IDs (contains no_user_ids entries, record_type = U)      *
# ****************************************************************************
# Record Format: "U <u_key> <u_id_string> [comment]".
#
# A USER_ID is a character string without white-spaces.
# The maximum length is 8 characters.
# Each USER-ID string is uniquely associated with an integer key.
# The integer key is used to identify "it's" USER_ID string in the
# workload class description (see below). The order of the USER-IDs or
# the integer keys is immaterial. The integer key range must not have gaps.
# Zeroes are not allowed as key values. The number of USER-IDs has to be
# equal to the value of no_user_ids in the global section (this condition
# is checked by the program! Note: ID-Strings are case sensitive!!!
#
# IMPORTANT NOTE: Actually, the key list must be dense and in increasing
# =============== order. This may be changed in future versions of the
```

```
#         modules which read and process this type of input file.
#         Though it is not really necessary to explicitly represent
#         the keys in this file (under this restriction), it makes
#         debugging easier because the file is much easier to read.
#
U 1 UALPHA
U 2 UBRAVO
U 3 UCHARLY
#
#
#*****************************************************************************
# * List of TERM_IDs (contains no_term_ids entries, record_type = T)     *
#*****************************************************************************
# Record Format: "T <t_key> <t_id_string> [comment]".
#
# Specifications and restrictions of user_id list applies (see above).
#
T 1 TDELTA
T 2 TECHO
T 3 TFOXTROT
T 4 TGOLF
#
#
#*****************************************************************************
# * List of TRAN_IDs (contains no_tran_ids entries, record_type = X (xaction"))*
#*****************************************************************************
# Record Format: "X <x_key> <x_id_string> [comment]".
#
# Specifications and restrictions of user_id list applies (see above).
#
X 1 XINDIA
X 2 XJULIET
X 3 XKILO
X 4 XLIMA
X 5 XMIKE
#
#
#
#*****************************************************************************
# * Description of Workload Class Elements (no_workload_class_elements, 'e') *
#*****************************************************************************
# Record Format: "e <element_key> <u_key> <t_key> <x_key> [comment]".
#
# One entry for each occurring Workload Class Member. The members are
# some kind of "refined transaction types" and are uniquely defined
# by the triplet (user_id, term_id, tran_id). The IDs are
# not represented as strings, but as their integer keys as described in the
# above lists (U, T, and X).
#
e 1 1 2 3
e 2 1 2 4
e 3 1 3 1
e 4 1 4 1
e 5 3 3 2
e 6 1 2 5
```

```
e 7 3 1 4
e 8 1 4 5
e 9 2 1 5
e 10 2 2 4
e 11 2 3 3
e 12 2 4 2
#
#
#*****************************************************************************
# * Workload Class List    (contains no_workload_classes entires, rect = W)*
#*****************************************************************************
# Record Formats: "W <wl_class_key> <number_of_elements> [comment]".
#        "w <element_key1> <... _5> [comment]".
#
# Each workload class consists of a set of workload class members. The class
# description is introduced with a workload class record (record type 'W'). This
# record is followed by a series of 'w' records. A 'w' record lists up to
# 5 class members. A class member is identified by the element_key from the
# above element list. In an initial input file, each workload class has exactly
# one member.
#
#
# Workload Class Description (Class_1, 3 Members)
W 1 3
w 1
w 2
w 3
# Workload Class Description (Class_2, 4 Members)
W 2 5
w 9
w 10
w 11
w 12
# Workload Class Description (Class_3, 5 Members)
W 3 5
w 4
w 6
w 5
w 7
w 8
#
#*****************************************************************************
# * Description of Data Class Elements (no_workload_class_elements, 'b')   *
#*****************************************************************************
# Record Format: "b <bucket_key> <min> <max> [comment]".
#
# A data class member is a continuous sequence ("bucket": min, ... max) of data
# objects accessed by workload class members/elements. A data class may be
# described by physical block numbers or lock names or whatever as long as
# the identifiers can be used to identify the physical location of data.
# Min and Max may have the same value. The data bucket is an atomic unit;
# Normally, two (or more) buckets are NOT "melted together". If two buckets are
# assumed to be similar, they are put into/assumed to be in the same data class!
#
b 1 0 9
```

```
b 2 10 19
b 3 20 29
b 4 30 39
b 5 40 49
b 6 50 59
b 7 60 69
b 8 70 79
b 9 80 89
b 10 90 99
#
#
#
#
#*************************************************************************
# * Data Class List    (contains no_data_classes entires, rect = D)    *
#*************************************************************************
# Record Formats: "D <d_class_key> <number_of_elements> [comment]".
#         "d <bucket_key1> <... _5> [comment]".
#
# Each data class consists of a set of data class members (buckets). The class
# description is introduced with a data class record (record type 'D'). This
# record is followed by a series of 'd' records. A 'd' record lists up to
# 5 class members. A class member is identified by the bucket_key from the
# above element list. In an initial input file, each data class has exactly one
# member.
#
# Data Class Description (Class_1, 1 member)
D 1 1
d 1
# Data Class Description (Class_2, 1 member)
D 2 1
d 2
# Data Class Description (Class_3, 1 member)
D 3 1
d 3
# Data Class Description (Class_4, 1 member)
D 4 1
d 4
# Data Class Description (Class_5, 1 member)
D 5 1
d 5
# Data Class Description (Class_6, 1 member)
D 6 1
d 6
# Data Class Description (Class_7, 1 member)
D 7 1
d 7
# Data Class Description (Class_8, 1 member)
D 8 1
d 8
# Data Class Description (Class_9, 1 member)
D 9 1
d 9
# Data Class Description (Class_10, 1 member)
D 10 1
```

```
d 10
#
#
#
#
#*****************************************************************************
# * Reference Matrix                                *
#*****************************************************************************
# Record Formats: "r <workload_class_key> <data_class_key> <#accesses> [comnt]".
#        "u <workload_class_key> <data_class_key> <#accesses> [comnt]".
#
# This section contains all non_zero entries of the reference matrix. This
# matrix is organized as follows:
#
#  - a row for each workload class
#  - a column for each data class
#  - each entry consists of two sub-entries:
#   . number of retrieval/read accesses (record type 'r')
#   . number of update/write accesses (record type 'u')
#
# A line in this file describes exactly one sub_entry of the reference matrix.
#
# Though zero_entries may be stored here, this is not necessary and therefore
# a waste of disk space.
#
r 1 1 1.000000e+02
u 1 1 101
r 2 2 1000
u 2 2 1001
r 3 3 10000
u 3 3 10001
r 1 4 200
r 2 5 2000
r 3 6 20000
u 1 7 301
u 2 8 3001
u 3 9 30001
u 3 10 300001
#
#
#*****************************************************************************
# ********** This is the last line of the workload description file ******
```

Γ.3 SCRIPT Reference Input File

```
# This is a comment
# Maximum line length is 80 characters.
#-------1---------2---------3---------4---------5---------6---------7---------8

#   FILENAME: reference-input.pro
#  DESCRIPTION: Contains a sample protocol script file, and shows all the
#       commands / settings of the script grammar.
# CREATION DATE: July 13rd, 1994
```

```
#

# Note: The order of the following SECTIONS and the order of each command
# ----- WITHIN the sections is immaterial.
#

#*****************************************************************************
# * Settings Section                                *
#*****************************************************************************
#
set maximum number of utilization classes = 15
# Sets the maximum acceptable number of utilization classes
# (i.e. the goal of the clustering algorithm)

set automatic clustering = ON
# Sets whether the clustering performed will re-compute the clustering
# distance automatically or not. Valid choices are on/true OR off/false.

set metric = PLDM
# Sets the distance metric to be used by the clustering algorithm.
# For the time being valid choices are:
# PLDM = Pseudo Linear Dependency Metric
# VDM = Vector Dependency Metric

set clustering distance = 0.2
# OR: set clustering distance = sqrt(2.0)
# OR: set clustering distance = sqrt(2.0) / 15
# Sets the (initial or not) clustering distance for the clustering algorithm.
# Valid parameters are of type double or sqrt(double) or sqrt(double) / integer.

set clustering distance increase ratio = 10%
set clustering distance decrease ratio = 10%
# Sets the ratio that CLOU will use while automatically re-adjusting the
# clustering distance (increase or decrease)

set clustering distance upper bound = 0.0001
set clustering distance lower bound = 1.0
# Sets the range of allowable values for the clustering distance
# Upper bound should definitely be 1 (or less) for PLDM

set maximum clustering distance increase ratio = 1000%
set maximum clustering distance decrease ratio = 90%
# Sets the maximum allowable ratio that CLUE can use while automatically
# re-adjusting the clustering distance (increase or decrease).
# Going off bounds equals to the termination of the algorithm.

set maximum number of ping-pongs = 3
# Sets the maximum number, the algorithm is allowed to "ping-pong", i.e.
# to change its mind about decreasing / increasing the clustering distance.

set max compression rate = 10%
# Sets the maximum acceptable compression rate of the number of PUCs.
# When the compression rates becomes more than that value
# the clustering distance has to be re-adjusted (decreased).
# Valid parameters are of type double, or percentage.
```

```
set k factor = 15%
# Sets the minimum acceptable compression of the compression rate of PUCs.
# When the compression rates becomes less than some k% of its original value,
# the clustering distance has to be re-adjusted (increased).
# Valid parameters are of type double, or percentage.

set goal almost reached at 5%
# Sets the "proximity" to the original goal criterion which is used
# to relax the k_factor criterion, when goal is almost reached.

set unify style = avg
# Sets the behavior of the unify function regarding the update of the
# reference matrix. For the time being valid choices are :
# ADD: Simply adds up the vector rows of the workload classes to be unified.
# AVG: Calculates the average of the vector rows of the workload classes
#    to be unified.

set stop at no change = 10
# Sets whether to stop the clustering when a modification (i.e. increase) of
# the clustering distance does not lead to a reduction of the number of PUCs
# during a ECS for a certain number of times (10 in the example).
# Valid parameters are of type integer.

set write weight = 1
set read weight = 1
# Set the weight to use when adding up the number of read and the number of
# write accesses. Valid parameters are of type integer.

set maximum intervals coefficient = 299%
set minimum intervals coefficient = 43%
# Set the two coefficients that are used to define the range in which the
# number of rows for each intervals must fall in.


set snapshots at every 1000 triplets
set snapshots when learning rate change by 20%
# Sets the conditions that must be met for the on the fly clustering
# algorithms to take a snapshot of their current state.


# ******************************************************************************
# * Commands Section                                                           *
# ******************************************************************************
#

store zeroes in column0
# OR: initialize column0
# Fills column 0 entries in the reference matrix with zeroes.

store sums in column0
# Fills column 0 entries in the reference matrix with the sum of
# the elements for each row.

wipeout zero rows
# Completely eliminates rows with all elements=0
```

```
sort
# The 1st preprocessing step: sorting of rows in descending order of references.

store sum squares in column0
# Fills column 0 entries in the reference matrix with the sum of
# the squares of the elements for each row.

downscale
# The 2nd preprocessing step: downscaling of all the reference matrix entries.
# (All matrix entries are divided by the maximum number of accesses. )

normalize
# The 3rd preprocessing step: normalization of the row vectors using the
# maximum vector norm.
# (All matrix entries are divided by the maximum of the maximum vector norms. )

do clustering
# Perform the clustering (used when automatic clustering is on).

do clustering with HALC
# Perform the clustering with HALC (= the default)
# (other possible candidates: BEA, KMEANS, ISODATA).


do clustering step
do 6 clustering steps
# Perform one or more Elementary Clustering Steps.


do clustering on the fly using KMeans minimizing MSE [with KMeans] [@ 8890]
do clustering on the fly using KMeans minimizing VWMSE [with KMeans] [@ 8891]
do clustering on the fly using graph 5 180% [with KMeans] [@ 8892]
# Start on the fly clustering with either one of the two KMeans Neural-Networks
# or the graph theoretical algorithm.
# The last one takes two arguments: the depth (an integer) of the path it will search
# for inconsistent edges and the percentage which will be used as a threshold
# value.
# If `with KMeans' is specified, then batch Kmeans will be run at each snapshot
# in order to compare its clustering with the one of the on-the-fly algorithm.
# Input to the on-the-fly algorithm can be given in two ways:
# i) Through the -fly option at the command line, in which case CLUE will read
#  the contents of the specified file, or
# ii) Through a client program which establishes a socket connection with CLUE
#  and uses that to pass to CLUE the triplets to be clustered. The socket port
#  that will be used in this connection is the number given after the `@' character.


# ********************************************************************************
# ********** This is the last line of the protocol description file ******
```

Παράρτημα Δ

Αποτελέσματα Αρχικών Πειραμάτων με τον Αναπροσαρμοζόμενο K-MEANS

Στα κάτωθι τμήματα παρατίθενται τα αποτελέσματα των πειραμάτων που έγιναν με τα δεδομένα που περιγράφονται στο κεφάλαιο 5 με χρήση στο μεν τμήμα Δ.1 του αναπροσαρμοζόμενου K-MEANS που ελαχιστοποιεί τη συνάρτηση VWMSE, στο δε τμήμα Δ.2 του αναπροσαρμοζόμενου K-MEANS που ελαχιστοποιεί τη συνάρτηση MSE.

Τα μηδενικά στοιχεία των πινάκων που ακολουθούν έχουν παραλειφθεί ώστε να είναι ευκολότερη η κατανόησή τους.

Δ.1 Πειράματα με συνάρτηση κόστους τη VWMSE

Ο πίνακας Δ.1 περιέχει τα αποτελέσματα της ομαδοποίησης των μη επικαλυπτόμενων τετραγώνων, όταν τα κέντρα έχουν αρχικοποιηθεί με τυχαίες τιμές.

Τα αντίστοιχα αποτελέσματα για μη επικαλυπτόμενα τετράγωνα, όταν έχουν χρησιμοποιηθεί παραδείγματα ως αρχικές τιμές των κέντρων, φαίνονται στον πίνακα Δ.2.

Όταν εισαγάγουμε και θόρυβο (μη μηδενικές τιμές εκτός των τετραγώνων), τότε, για τυχαία αρχικοποίηση των κέντρων λαμβάνουμε τα αποτελέσματα του πίνακα Δ.3, ενώ για αρχικοποίηση από τα παραδείγματα, αυτά του Δ.4.

Όταν έχουμε μερικώς επικαλυπτόμενα τετράγωνα και τυχαία αρχικοποίηση, τότε λαμβάνουμε τα αποτελέσματα του πίνακα Δ.5, ενώ όταν η αρχικοποίηση γίνεται με κάποια εκ των παραδειγμάτων, τότε αυτά του Δ.6.

Εάν επιπλέον, εισαγάγουμε και θόρυβο στα μερικώς επικαλυπτόμενα τετράγωνα, τότε τα αποτελέσματα μεταβάλλονται όπως φαίνεται στον πίνακα Δ.7, καθώς και στον Δ.8.

Η κατακόρυφη διάσταση των προαναφερθέντων πινάκων δείχνει τις πραγματικές ομάδες των παραδειγμάτων (πραγματικά δεδομένα), ενώ η οριζόντια τις ομάδες που εξήγαγε τελικώς ο αλγόριθμος (εκτιμούμενα αποτελέσματα). Έτσι, π.χ. η θέση (1,7) ενός πίνακα αναφέρει πόσα παραδείγματα της πρώτης ομάδας τοποθετήθηκαν τελικώς στην έβδομη.

Παρατηρώντας τα αποτελέσματα, βλέπουμε ότι η τυχαία αρχικοποίηση δίνει καλύτερα αποτελέσματα, ιδίως στις περιπτώσεις που υπάρχει θόρυβος ή τα τετράγωνα είναι μερικώς επικαλυπτόμενα.

Επίσης, παρατηρούμε ότι στην περίπτωση των μη επικαλυπτόμενων τετραγώνων, η τελική κατάταξη δεν είναι ικανοποιητική, καθώς όλες οι ομάδες εκτός τριών (τρίτη, ένατη και δέκατη) αναγνωρίζονται ώς η δέκατη κλάση και εκτός αυτού η δέκατη έχει διασπασθεί σε άλλες μικρότερες.

Αυτή την κακή συμπεριφορά δεν την έχει ο αλγόριθμος στα μερικώς επικαλυπτόμενα τετράγωνα (ακόμα και παρουσία θορύβου στα δεδομένα).

Μία πιθανή εξήγηση για το φαινόμενο αυτό, είναι πως τα παραδείγματα που έχουν μερικώς επικαλυπτόμενα τετράγωνα, είναι πιο ομαλά κατανεμημένα στις διάφορες ομάδες από αυτά που δεν έχουν επικαλυπτόμενα τετράγωνα και έτσι, η διασπορά v_k της κάθε ομάδας είναι περίπου η ίδια. Επομένως, η απόσταση τείνει να προσεγγίσει την Ευκλείδια, που στην περίπτωσή μας διαχωρίζει καλύτερα τις ομάδες.

Οι πίνακες Δ.9, Δ.10, Δ.11, Δ.12, Δ.13, Δ.14, Δ.15 και Δ.16 παρουσιάζουν τα αποτελέσματα που προέκυψαν όταν χρησιμοποιήθηκε η Ευκλείδια απόσταση.

Σε αυτούς παρατηρούμε ότι το νευρωνικό δίκτυο επιδεικνύει πολύ καλύτερη συμπεριφορά και μεγαλύτερη ακρίβεια ομαδοποίησης όταν χρησιμοποιείται η κλασσική Ευκλείδια απόσταση.

Εκτιμούμενα

Πραγματικά

	1	2	3	4	5	6	7	8	9	10	Συνολικά
1								181			181
2								114			114
3			199								199
4								188			188
5								158			158
6								193			193
7								160			160
8								161			161
9									185		185
10	59	44		3	70	65	63			157	461
Συνολικά	59	44	199	3	70	65	63	1155	185	157	2000

Πίνακας Δ.1: NN-VWMSE: Μη επικαλυπτόμενα τετράγωνα δίχως θόρυβο (Αρχικοποίηση τυχαία)

Εκτιμούμενα

Πραγματικά

	1	2	3	4	5	6	7	8	9	10	Συνολικά
1									181		181
2									114		114
3			199								199
4									188		188
5					158						158
6									193		193
7									160		160
8									161		161
9									185		185
10	6	107		100		5	70	57		116	461
Συνολικά	6	107	199	100	158	5	70	57	1182	116	2000

Πίνακας Δ.2: NN-VWMSE: Μη επικαλυπτόμενα τετράγωνα δίχως θόρυβο

Εκτιμούμενα →
Πραγματικά ↓

	1	2	3	4	5	6	7	8	9	10	Συνολικά
1	181										181
2								114			114
3			199								199
4								188			188
5					158						158
6						193					193
7							160				160
8								161			161
9									185		185
10		153		132					2	174	461
Συνολικά	181	153	199	132	158	193	160	463	187	174	2000

Πίνακας Δ.3: NN-VWMSE: Μη επικαλυπτόμενα τετράγωνα με θόρυβο (Αρχικοποίηση τυχαία)

Εκτιμούμενα →
Πραγματικά ↓

	1	2	3	4	5	6	7	8	9	10	Συνολικά
1				181							181
2				114							114
3			199								199
4				188							188
5					158						158
6						193					193
7							160				160
8								161			161
9									185		185
10	152	136							1	172	461
Συνολικά	152	136	199	483	158	193	160	161	186	172	2000

Πίνακας Δ.4: NN-VWMSE: Μη επικαλυπτόμενα τετράγωνα με θόρυβο

Εκτιμούμενα

Πραγματικά

	1	2	3	4	5	6	7	8	9	10	Συνολικά
1						175					175
2						167					167
3				145		53					198
4				154		15					169
5				1		164	3				168
6						111					111
7						22	150		6		178
8							1		126		127
9	8	9	3		14			6	62	6	108
10	100	98	94		92			97		118	599
Συνολικά	108	107	97	300	106	707	154	103	194	124	2000

Πίνακας Δ.5: NN-VWMSE: Μερικώς επικαλυπτόμενα τετράγωνα δίχως θόρυβο (Αρχικοποίηση τυχαία)

Εκτιμούμενα

Πραγματικά

	1	2	3	4	5	6	7	8	9	10	Συνολικά
1						175					175
2						167					167
3				145		53					198
4				154		15					169
5				1		164	3				168
6						111					111
7						22	150		6		178
8							1		126		127
9	4	9	8		8			4	62	13	108
10	85	115	87		109			85		118	599
Συνολικά	89	124	95	300	117	707	154	89	194	131	2000

Πίνακας Δ.6: NN-VWMSE: Μερικώς επικαλυπτόμενα τετράγωνα δίχως θόρυβο

Εκτιμούμενα →

Πραγματικά ↓

	1	2	3	4	5	6	7	8	9	10	Συνολικά
1		175									175
2		119						48			167
3			121	24				53			198
4			1	153	15						169
5				1	125	42					168
6					1	110					111
7						22	150		6		178
8							1		126		127
9	21								62	25	108
10	294				1					304	599
Συνολικά	315	294	122	178	142	174	151	101	194	329	2000

Πίνακας Δ.7: NN-VWMSE: Μερικώς επικαλυπτόμενα τετράγωνα με θόρυβο (Αρχικοποίηση τυχαία)

Εκτιμούμενα →

Πραγματικά ↓

	1	2	3	4	5	6	7	8	9	10	Συνολικά
1		174			1						175
2		165			2						167
3		53	121	24							198
4			1	153	15						169
5				1	125	42					168
6					1	110					111
7						22	150		6		178
8							1		126		127
9	10							17	62	19	108
10	170							197		232	599
Συνολικά	180	392	122	178	144	174	151	214	194	251	2000

Πίνακας Δ.8: NN-VWMSE: Μερικώς επικαλυπτόμενα τετράγωνα με θόρυβο

Δ.2 Πειράματα με συνάρτηση κόστους τη MSE

Εκτιμούμενα →
Προγραμματικά ↓

	1	2	3	4	5	6	7	8	9	10	Συνολικά
1	172	9									181
2	1	105	8								114
3	1		191	7							199
4	1			181	6						188
5	1				152	5					158
6	1					188	4				193
7	1						156	3			160
8	1							158	2		161
9	1								183	1	185
10	1									460	461
Συνολικά	181	114	199	188	158	193	160	161	185	461	2000

Πίνακας Δ.9: NN-MSE: Μή επικαλυπτόμενα τετράγωνα δίχως θόρυβο (Αρχικοποίηση τυχαία)

Εκτιμούμενα →
Προγραμματικά ↓

	1	2	3	4	5	6	7	8	9	10	Συνολικά
1	181										181
2		114									114
3			199								199
4				188							188
5					158						158
6						193					193
7							160				160
8								161			161
9									185		185
10										461	461
Συνολικά	181	114	199	188	158	193	160	161	185	461	2000

Πίνακας Δ.10: NN-MSE: Μή επικαλυπτόμενα τετράγωνα δίχως θόρυβο

Πραγματικά \ Εκτιμούμενα

	1	2	3	4	5	6	7	8	9	10	Συνολικά
1	172	9									181
2	1	105	8								114
3	1		191	7							199
4	1			181	6						188
5	1				152	5					158
6	1					188	4				193
7	1						156	3			160
8	1							158	2		161
9	1								183	1	185
10	1									460	461
Συνολικά	181	114	199	188	158	193	160	161	185	461	2000

Πίνακας Δ.11: NN-MSE: Μή επικαλυπτόμενα τετράγωνα με θόρυβο (Αρχικοποίηση τυχαία)

Πραγματικά \ Εκτιμούμενα

	1	2	3	4	5	6	7	8	9	10	Συνολικά
1	181										181
2		114									114
3			199								199
4				188							188
5					158						158
6						193					193
7							160				160
8								161			161
9									185		185
10										461	461
Συνολικά	181	114	199	188	158	193	160	161	185	461	2000

Πίνακας Δ.12: NN-MSE: Μή επικαλυπτόμενα τετράγωνα με θόρυβο

Εκτιμούμενα

Πραγματικά

	1	2	3	4	5	6	7	8	9	10	Συνολικά
1	175										175
2	1	110						56			167
3	1		122	30				45			198
4	1			148	20						169
5	1				121	46					168
6	1					110					111
7	1					18	151		8		178
8	1								126		127
9	1								60	47	108
10	1									598	599
Συνολικά	184	110	122	178	141	174	151	101	194	645	2000

Πίνακας Δ.13: NN-MSE: Μερικώς επικαλυπτόμενα τετράγωνα δίχως θόρυβο (Αρχικοποίηση τυχαία)

Εκτιμούμενα

Πραγματικά

	1	2	3	4	5	6	7	8	9	10	Συνολικά
1	175										175
2	9	110						48			167
3			121	24				53			198
4			1	153	15						169
5				1	125	42					168
6					1	110					111
7						22	150		6		178
8							1		126		127
9									62	46	108
10										599	599
Συνολικά	184	110	122	178	141	174	151	101	194	645	2000

Πίνακας Δ.14: NN-MSE: Μερικώς επικαλυπτόμενα τετράγωνα δίχως θόρυβο

Εκτιμούμενα

Πραγματικά

	1	2	3	4	5	6	7	8	9	10	Συνολικά
1	175										175
2	1	110						56			167
3	1		122	30				45			198
4	1			148	20						169
5	1				121	46					168
6	1					110					111
7	1					18	151		8		178
8	1								126		127
9	1								60	47	108
10	1									598	599
Συνολικά	184	110	122	178	141	174	151	101	194	645	2000

Πίνακας Δ.15: NN-MSE: Μερικώς επικαλυπτόμενα τετράγωνα με θόρυβο (Αρχικοποίηση τυχαία)

Εκτιμούμενα

Πραγματικά

	1	2	3	4	5	6	7	8	9	10	Συνολικά
1	175										175
2	9	110						48			167
3			121	24				53			198
4			1	153	15						169
5				1	125	42					168
6					1	110					111
7						22	150		6		178
8							1		126		127
9									62	46	108
10										599	599
Συνολικά	184	110	122	178	141	174	151	101	194	645	2000

Πίνακας Δ.16: NN-MSE: Μερικώς επικαλυπτόμενα τετράγωνα με θόρυβο

Παράρτημα Ε

Αποτελέσματα Αρχικών Πειραμάτων με τη Γραφοθεωρητική Μέθοδο

Στους πίνακες που ακολουθούν Ε.1, Ε.2, Ε.3, και Ε.4, παρουσιάζονται τα αποτελέσματα των πειραμάτων με τη γραφοθεωρητική μέθοδο ομαδοποίησης ομαδικής επεξεργασίας (βλέπε τον πίνακα 4.1 του υποκεφαλαίου 4.2).

Εκτιμούμενα →
Πραγματικά ↓

	1	2	3	4	5	6	7	8	9	10	Συνολικά
1	181										181
2		114									114
3			199								199
4				188							188
5					158						158
6						193					193
7							160				160
8								161			161
9									185		185
10										461	461
Συνολικά	181	114	199	188	158	193	160	161	185	461	2000

Πίνακας Ε.1: Γραφοθεωρητική Μέθοδος: Μη επικαλυπτόμενα τετράγωνα, δίχως θόρυβο

Εκτιμούμενα →
Πραγματικά ↓

	1	2	3	4	5	Συνολικά
1				181		181
2		114				114
3				199		199
4				188		188
5					158	158
6				193		193
7				160		160
8				161		161
9			185			185
10	461					461
Συνολικά	461	114	185	1082	158	2000

Πίνακας Ε.2: Γραφοθεωρητική Μέθοδος: Μη επικαλυπτόμενα τετράγωνα, με θόρυβο

Εκτιμούμενα →
Πραγματικά ↓

	1	2	3	4	5	6	7	8	9	10	11	Συνολικά
1	184											184
2		110										110
3			101									101
4				122								122
5					178							178
6						141						141
7							174					174
8								151				151
9									194			194
10										624	21	645
Συνολικά	184	110	101	122	178	141	174	151	194	624	21	2000

Πίνακας Ε.3: Γραφοθεωρητική Μέθοδος: Μερικώς επικαλυπτόμενα τετράγωνα, δίχως θόρυβο

Εκτιμούμενα →
Πραγματικά ↓

	1	2	3	4	Συνολικά
1			184		184
2			110		110
3			101		101
4				122	122
5			178		178
6			141		141
7			174		174
8	151				151
9		194			194
10			645		645
Συνολικά	151	194	1533	122	2000

Πίνακας Ε.4: Γραφοθεωρητική Μέθοδος: Μερικώς επικαλυπτόμενα τετράγωνα, με θόρυβο

Παράρτημα ΣΤ

Τελικά Πειράματα

Στο παράρτημα αυτό περιγράφονται τα αρχεία εισόδου που χρησιμοποιήθησαν κατά τη διεξαγωγή των πειραμάτων ώστε να είναι δυνατή η επανάληψή τους.

ΣΤ.1 Διαχωρισμός των συνόλων δεδομένων

Τα σύνολα δεδομένων που χρησιμοποιήθησαν στα πειράματα, διαχωρίσθησαν σε δύο υποσύνολα, ένα που χρησιμοποιήθηκε για την εκπαίδευση των αλγόριθμων ομαδοποίησης ''εν πτήσει'' και ένα που χρησιμοποιήθηκε για την αξιολόγησή τους. Ο διαχωρισμός έγινε με τη χρήση του προγράμματος TTS (συντομογραφία του training & testing sets). Το πρόγραμμα αυτό λαμβάνει τα εξής ορίσματα:

- `-f` *Υποχρεωτικό*

  Πρέπει να ακολουθείται από το όνομα του αρχείου που περιέχει τα προς διάσπαση δεδομένα σε μορφή εισόδου του CLUE (βλέπε το παράρτημα Γ για την περιγραφή της μορφής αυτής).

- `-p` *Υποχρεωτικό*

  Πρέπει να ακολουθείται από το ποσοστό των δεδομένων που θα χρησιμοποιηθούν για εκπαίδευση, στο διάστημα (0,1).

- `-c` *Προαιρετικό*

  Πρέπει να ακολουθείται από το πλήθος των ομάδων που σχηματίζουν τα δεδομένα. Όταν δίνεται, το TTS βάζει στα δεδομένα που θα χρησιμοποιηθούν για εκπαίδευση ίσο αριθμό από κάθε ομάδα.

- `-s` *Προαιρετικό*

  Πρέπει να ακολουθείται από έναν αριθμό ο οποίος θα χρησιμοποιηθεί για να αρχικοποιήσει την γεννήτρια τυχαίων αριθμών που χρησιμοποιεί το TTS. Εάν δεν δοθεί, τότε χρησιμοποιείται ο τρέχων χρόνος του συστήματος για να υπολογισθεί ένας τέτοιος αριθμός και το TTS τον αναφέρει στην έξοδό του.

Το TTS παράγει δύο αρχεία τα οποία ονομάζει προσθέτοντας στο τέλος του ονόματος του αρχείου εισόδου τα επιθέματα -training και -testing. Έτσι, αν εκτελέσει κάποιος την εντολή:

```
tts -f input -c 100 -p .10 -s 32481
```

θα δημιουργηθούν δύο αρχεία, το `input-training` και το `input-testing` με πλήθος δεδομένων 10% και 90% αντίστοιχα.

Οι εντολές που θα παράγουν τα ίδια αρχεία δεδομένων με αυτά που χρησιμοποιήθησαν στα τελικά πειράματα του κεφαλαίου 6, είναι οι κάτωθι:

```
tts -f 1000x1000=100:diagsR -c 100 -p .10 -s 845291227
tts -f 1000x1000=100:diagsR -c 100 -p .10 -s 845411653
tts -f 1000x1000=100:diagsR -c 100 -p .10 -s 845637782
tts -f 1000x1000=100:diagsR -c 100 -p .10 -s 845653660
tts -f 1000x1000=100:diagsR -c 100 -p .10 -s 845728222
tts -f 1000x1000=100:diagsR -c 100 -p .10 -s 845741067
tts -f 1000x1000=100:diagsR -c 100 -p .10 -s 845899157
tts -f 1000x1000=100:diagsR -c 100 -p .10 -s 845908929
tts -f 1000x1000=100:diagsR -c 100 -p .10 -s 845915609
tts -f 1000x1000=100:diagsR -c 100 -p .10 -s 845919532

tts -f 10000x1000=100:diagsR -c 100 -p .10 -s 845925843
tts -f 10000x1000=100:diagsR -c 100 -p .10 -s 845927624
tts -f 10000x1000=100:diagsR -c 100 -p .10 -s 846174694
tts -f 10000x1000=100:diagsR -c 100 -p .10 -s 846181050
tts -f 10000x1000=100:diagsR -c 100 -p .10 -s 846237748
tts -f 10000x1000=100:diagsR -c 100 -p .10 -s 846238604
tts -f 10000x1000=100:diagsR -c 100 -p .10 -s 846341981
tts -f 10000x1000=100:diagsR -c 100 -p .10 -s 846342013
tts -f 10000x1000=100:diagsR -c 100 -p .10 -s 846416416
tts -f 10000x1000=100:diagsR -c 100 -p .10 -s 846446662

tts -f 30000x1000=100:diagsR -c 100 -p .03333333333333333333 -s 846777221
tts -f 30000x1000=100:diagsR -c 100 -p .03333333333333333333 -s 846777222
tts -f 30000x1000=100:diagsR -c 100 -p .03333333333333333333 -s 846796739
tts -f 30000x1000=100:diagsR -c 100 -p .03333333333333333333 -s 846938197
tts -f 30000x1000=100:diagsR -c 100 -p .03333333333333333333 -s 847039933
tts -f 30000x1000=100:diagsR -c 100 -p .03333333333333333333 -s 847039935
tts -f 30000x1000=100:diagsR -c 100 -p .03333333333333333333 -s 847109283
tts -f 30000x1000=100:diagsR -c 100 -p .03333333333333333333 -s 847109286
tts -f 30000x1000=100:diagsR -c 100 -p .03333333333333333333 -s 847191772
tts -f 30000x1000=100:diagsR -c 100 -p .03333333333333333333 -s 847191787

tts -f PULS.clue -p .10869565217391304347 -s 847720836
tts -f PULS.clue -p .10869565217391304347 -s 847828982
tts -f PULS.clue -p .10869565217391304347 -s 847829089
tts -f PULS.clue -p .10869565217391304347 -s 847829234
tts -f PULS.clue -p .10869565217391304347 -s 847829441
tts -f PULS.clue -p .10869565217391304347 -s 847829575
tts -f PULS.clue -p .10869565217391304347 -s 847829709
tts -f PULS.clue -p .10869565217391304347 -s 847829844
tts -f PULS.clue -p .10869565217391304347 -s 847829984
tts -f PULS.clue -p .10869565217391304347 -s 847830090

tts -f DOA.clue -p .10 -s 847923159
tts -f DOA.clue -p .10 -s 847923329
tts -f DOA.clue -p .10 -s 847923499
tts -f DOA.clue -p .10 -s 847923671
tts -f DOA.clue -p .10 -s 847923841
tts -f DOA.clue -p .10 -s 847924010
```

```
tts -f DOA.clue -p .10 -s 847924179
tts -f DOA.clue -p .10 -s 847924439
tts -f DOA.clue -p .10 -s 847924815
tts -f DOA.clue -p .10 -s 847925176
```

Ο λόγος για τον οποίο το ποσοστό των δεδομένων προς εκπαίδευση στην περίπτωση του συνόλου 30000x1000=100:diagsR είναι .033... αντί για .10 είναι ότι θέλαμε να έχουμε μόνο 1000 δεδομένα προς εκπαίδευση, όπως και στο σύνολο 10000x1000=100:diagsR.

Στην περίπτωση του συνόλου PULS, χρησιμοποιήθηκε το ποσοστό .10869565217391304347 γιατί αυτό έδινε 30 δεδομένα προς εκπαίδευση όσες ήταν και οι ομάδες που ζητούσαμε. Έτσι, οι νευρώνες των δύο δικτύων μπορούσαν να αρχικοποιηθούν όλοι.

ΣΤ.2 Αρχεία εντολών του CLUE που χρησιμοποιήθησαν

Τα πειράματα με το CLUE έγιναν τρέχοντάς το ως εξής (αναλόγως με τον αλγόριθμο που θα εκτελούνταν):

```
clue -in 1000x1000=100:diagsR-training -fly 1000x1000=100:diagsR-testing \
 -out snapshot -par script-1000-nnmse
clue -in 1000x1000=100:diagsR-training -fly 1000x1000=100:diagsR-testing \
 -out snapshot -par script-1000-nnvwmse
clue -in 1000x1000=100:diagsR-training -fly 1000x1000=100:diagsR-testing \
 -out snapshot -par script-1000-kmeans
clue -in 1000x1000=100:diagsR-training -fly 1000x1000=100:diagsR-testing \
 -out snapshot -par script-1000-halc
clue -in 1000x1000=100:diagsR-training -fly 1000x1000=100:diagsR-testing \
 -out snapshot -par script-1000-graphos

clue -in 10000x1000=100:diagsR-training -fly 10000x1000=100:diagsR-testing \
 -out snapshot -par script-10000-nnmse
clue -in 10000x1000=100:diagsR-training -fly 10000x1000=100:diagsR-testing \
 -out snapshot -par script-10000-nnvwmse
clue -in 10000x1000=100:diagsR-training -fly 10000x1000=100:diagsR-testing \
 -out snapshot -par script-10000-kmeans
clue -in 10000x1000=100:diagsR-training -fly 10000x1000=100:diagsR-testing \
 -out snapshot -par script-10000-halc
clue -in 10000x1000=100:diagsR-training -fly 10000x1000=100:diagsR-testing \
 -out snapshot -par script-10000-graphos

clue -in 30000x1000=100:diagsR-training -fly 30000x1000=100:diagsR-testing \
 -out snapshot -par script-30000-nnmse
clue -in 30000x1000=100:diagsR-training -fly 30000x1000=100:diagsR-testing \
 -out snapshot -par script-30000-nnvwmse
clue -in 30000x1000=100:diagsR-training -fly 30000x1000=100:diagsR-testing \
 -out snapshot -par script-30000-kmeans
clue -in 30000x1000=100:diagsR-training -fly 30000x1000=100:diagsR-testing \
 -out snapshot -par script-30000-halc

clue -in PULS.clue-training -fly PULS.clue-testing \
 -out snapshot -par script-PULS-nnmse
clue -in PULS.clue-training -fly PULS.clue-testing \
 -out snapshot -par script-PULS-nnvwmse
clue -in PULS.clue-training -fly PULS.clue-testing \
```

```
 -out snapshot -par script-PULS-kmeans
clue -in PULS.clue-training -fly PULS.clue-testing \
 -out snapshot -par script-PULS-halc
clue -in PULS.clue-training -fly PULS.clue-testing \
 -out snapshot -par script-PULS-graphos

clue -in DOA.clue-training -fly DOA.clue-testing \
 -out snapshot -par script-DOA-nnmse
clue -in DOA.clue-training -fly DOA.clue-testing \
 -out snapshot -par script-DOA-nnvwmse
clue -in DOA.clue-training -fly DOA.clue-testing \
 -out snapshot -par script-DOA-kmeans
clue -in DOA.clue-training -fly DOA.clue-testing \
 -out snapshot -par script-DOA-halc
```

Τα αρχεία script-... περιγράφονται στα ακόλουθα υπο-υποκεφάλαια:

ΣΤ.2.1 script-1000-nnmse

```
enable DEBUG_OTHER

#-------------------------- Settings --------------------------
set maximum number of utilization classes = 100
set metric = VDM
set random split = On

set snapshots at every 50 triplets

store zeroes in column0
store sums in column0
wipeout zero rows

# Use KMeans to compute the clustering to be used as an initial one
# for the on-the-fly algorithms
 timestamp KMeans_Start
 do clustering with ISODATA
 timestamp KMeans_End

 timestamp OntheFly_MSE_Start
 do clustering on the fly using Kmeans minimizing MSE
 timestamp OntheFly_MSE_End
```

ΣΤ.2.2 script-1000-nnvwmse

```
enable DEBUG_OTHER

#-------------------------- Settings --------------------------
set maximum number of utilization classes = 100
set metric = VDM
set random split = On

set snapshots at every 50 triplets

store zeroes in column0
store sums in column0
```

```
wipeout zero rows

# Use KMeans to compute the clustering to be used as an initial one
# for the on-the-fly algorithms
 timestamp KMeans_Start
 do clustering with ISODATA
 timestamp KMeans_End

 timestamp OntheFly_VWMSE_Start
 do clustering on the fly using Kmeans minimizing VWMSE
 timestamp OntheFly_VWMSE_End
```

ΣΤ.2.3 script-1000-graphos

```
enable DEBUG_OTHER

#------------------------- Settings --------------------------
set maximum number of utilization classes = 100
set metric = VDM
set random split = On

set snapshots at every 50 triplets

store zeroes in column0
store sums in column0
wipeout zero rows

# Use KMeans to compute the clustering to be used as an initial one
# for the on-the-fly algorithms
 timestamp KMeans_Start
 do clustering with ISODATA
 timestamp KMeans_End

 timestamp OntheFly_Graphos_Start
 do clustering on the fly using graph 5 180%
 timestamp OntheFly_Graphos_End
```

ΣΤ.2.4 script-1000-kmeans

```
enable DEBUG_OTHER

#------------------------- Settings --------------------------
set maximum number of utilization classes = 100
set metric = VDM
set random split = On

store zeroes in column0
store sums in column0
wipeout zero rows

copy matrix

 timestamp KMeans_Start
 do clustering with ISODATA
 timestamp KMeans_End
```

```
swap matrix
evaluate results
```

ΣΤ.2.5 script-1000-halc

```
enable DEBUG_OTHER

#-------------------------- Settings --------------------------
set maximum number of utilization classes = 100
set automatic clustering = ON
 set metric = PLDM
set clustering distance = 0.2
set clustering distance increase ratio = 10%
set clustering distance decrease ratio = 10%
set maximum clustering distance increase ratio = -1.0
set maximum clustering distance decrease ratio = 99.99%
set clustering distance lower bound = 0.00001
set clustering distance upper bound = 0.95 # clust. dist. must be <<1.0
set maximum number of ping-pongs = 0
solve ping-pongs using last
set max compression rate = 10%
set k factor = 15%
set goal almost reached at 5%
set unify style = avg
set stop at no change = 500
set write weight = 1
set read weight = 1
set maximum intervals coefficient = 299%
set minimum intervals coefficient = 43%

#-------------------------- PLDM --------------------------
store zeroes in column0
store sums in column0
wipeout zero rows
downscale

copy matrix

 timestamp HALC_PLDM_Start
 do clustering
 timestamp HALC_PLDM_End

#-------------------------- Settings --------------------------
set metric = VDM
set clustering distance = sqrt(2.0) / 100
set clustering distance increase ratio = 50%
set clustering distance decrease ratio = 30%
set clustering distance lower bound = 0.0001
set clustering distance upper bound = 100.0
set maximum number of ping-pongs = 0
solve ping-pongs using last

store zeroes in column0
store sums in column0
```

```
wipeout zero rows
store squares in column0

 timestamp HALC_VDM_Start
 do clustering
 timestamp HALC_VDM_End

swap matrix
evaluate results
```

Τα αρχεία script-10000-... είναι ίδια με τα αντίστοιχα script-1000-..., με μόνη διαφορά ότι όπου υπήρχε η γραμμή:

```
set snapshots at every 50 triplets
```

έχει αντικατασταθεί με τη γραμμή:

```
set snapshots at every 1000 triplets
```

Αντιστοίχως, στα αρχεία script-30000-..., η γραμμή αυτή έχει γίνει:

```
set snapshots at every 3625 triplets
```

Στα αρχεία script-DOA-..., η γραμμή αυτή ήταν:

```
set snapshots at every 283 triplets
```

Τέλος, στα αρχεία script-PULS-..., η γραμμή αυτή ήταν:

```
set snapshots at every 30 triplets
```

Παράλληλα, στα αρχεία script-PULS-... αλλάχθηκε και ο αριθμός των ομάδων που ζητούσαμε, με το να αλλάξουμε τη γραμμή:

```
set maximum number of utilization classes = 100
```

με τη γραμμή:

```
set maximum number of utilization classes = 30
```

Σημείωση:
Όταν εκτελούνταν το CLUE με τους αλγόριθμους ομαδοποίησης ''εν πτήσει'', τότε δημιουργούνταν, εκτός από το τελικό αρχείο εξόδου snapshot και τα αρχεία snapshot-XXX-triplets, όπου XXX ο αριθμός των τριάδων που είχαν συμπληρωθεί στο εκάστοτε snapshot. Αυτά ήταν τα αρχεία εκείνα που χρησιμοποιήθησαν ως είσοδος στους αλγόριθμους ομαδικής επεξεργασίας (K-MEANS και HALC), προκειμένου να υπολογισθεί η επίδοσή τους για τις ενδιάμεσες φάσεις της ομαδοποίησης.

ΣΤ.3 Πλήθη ομάδων που κατασκευάσθησαν

Στους πίνακες ΣΤ.1, ΣΤ.2, ΣΤ.3, ΣΤ.4 και ΣΤ.5 που ακολουθούν, παρουσιάζονται τα πλήθη των ομάδων τις οποίες δημιούργησαν ο αλγόριθμος HALC και ο γραφοθεωρητικός αλγόριθμος ομαδοποίησης ''εν πτήσει'' (όποτε αυτός χρησιμοποιήθηκε) κατά την εκτέλεση των τελικών πειραμάτων που περιγράφονται στο κεφάλαιο 6. Σε όλες τις περιπτώσεις, πλην του συνόλου πραγματικών δεδομένων PULS, το πλήθος των ζητούμενων ομάδων ήταν ίσο με 100. Στην περίπτωση του συνόλου PULS (πίνακας ΣΤ.4), το πλήθος των ζητούμενων ομάδων ήταν ίσο με 30.

Triplets read on the fly	HALC	GRAPHOS
50	96.2	1.0
100	90.6	1.0
150	92.8	1.0
200	90.6	1.6
250	89.8	2.5
300	88.9	8.8
350	88.5	15.3
400	85.1	21.5
450	87.1	24.9
500	84.7	24.5
550	85.3	25.9
600	88	30.4
650	83.9	25.8
700	91.6	33.9
750	87.5	30.4
800	83.4	32.0
850	81.8	32.3
900	92	32.3

Πίνακας ΣΤ.1: Πλήθος ομάδων για το σύνολο συνθετικών δεδομένων 1000x1000=100:diagsR

Triplets read on the fly	HALC	GRAPHOS
1000	84.9	26.9
2000	85.2	45.3
3000	91.7	50.3
4000	93.1	58.6
5000	87.4	61.4
6000	83.6	63.2
7000	87.7	63.2
8000	92.6	67.9
9000	97.1	71.6

Πίνακας ΣΤ.2: Πλήθος ομάδων για το σύνολο συνθετικών δεδομένων 10000x1000=100:diagsR

Triplets read on the fly	HALC
3625	82.6
7250	89.4
10875	92.1
14500	87.7
18125	85.1
21750	87.1
25375	89.7
29000	86

Πίνακας ΣΤ.3: Πλήθος ομάδων για το σύνολο συνθετικών δεδομένων 30000x1000=100:diagsR

Triplets read on the fly	HALC	GRAPHOS
30	27.6	26.6
60	29	34.1
90	28.9	75
120	28.4	100.9
150	28.7	123.5
180	28.6	135.9
210	29.3	144.7
240	28.9	163.4
246	28.4	167.2

Πίνακας ΣΤ.4: Πλήθος ομάδων για το σύνολο πραγματικών δεδομένων PULS

Triplets read on the fly	HALC
283	99
566	91.8
849	97.3
1132	95
1415	96.4
1698	96.5
1748	93.9

Πίνακας ΣΤ.5: Πλήθος ομάδων για το σύνολο πραγματικών δεδομένων DOA

Βιβλιογραφία

[CB92] Maureen Caudill and Charles Butler. *Understanding Neural Networks: Computer Explorations*, volume Basic Networks. MIT Press, 1992.

[CG87] Gail A. Carpenter and Stephen Grossberg. A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine. In Lau [Lau92], pages 147--208.

[Cha83] Jean Pierre Changeux. *L' homme neuronal.* Fayard, 1983.

[Cor89] Thomas H. Cormen. *Introduction to Algorithms.* The MIT electrical engineering and computer science series. MIT Press, 1989.

[CS95] Chedsada Chinrungrueng and Carlo H. Séquin. Optimal Adaptive K-Means Algorithm with Dynamic Adjustment of Learning Rate. *IEEE Transactions on Neural Networks*, 6(1):157 -- 169, January 1995.

[Des88] D. Desieno. Adding a conscience to competitive learning. In *Proc. 2nd IEEE International Conference on Neural Networks (ICNN-88)*, volume I, July 1988.

[DM90a] Christian J. Darken and John Moody. Fast adaptive k-means clustering: Some empirical results. In *Proc. International Joint Conference on Neural Networks (IJCNN-90)*, June 1990.

[DM90b] Christian J. Darken and John Moody. Learning schedules for stochastics optimization. In *1990 IEEE Conf. Neural Information Processing Systems-Natural and Synthetic*, November 1990.

[Dui96] Robert P. W. Duin. A note on comparing classifiers. *Pattern Recognition Letters*, 17:529--536, 1996.

[ESP] ESPRIT III Basic Research Action Project 8144. Load Balancing on High Performance Parallel and Distributed Systems.
Dr. Christos Nikolaou (nikolau@ics.forth.gr) is the coordinator of the LYDIA project. More information about the LYDIA project can be found at the URL: http://www.ics.forth.gr/proj/pleiades/projects/LYDIA/.

[Ger79] Allen Gersho. Asymptotically optimal block quantization. *IEEE Transactions on Information Theory*, IT-25(4):373--380, 1979.

[GH85] R. L. Graham and Pavol Hell. On the history of the minimum spanning tree problem. *Annals of the History of Computing*, 7(1):43--57, 1985.

[Har75] John A. Hartigan. *Clustering Algorithms.* John Wiley and Sons, Inc., 1975.

[Hay94] Simon Haykin. *Neural Networks: A Comprehensive Foundation.* Macmillan Publishing Company, 113 Sylvan Avenue, Englewood Cliffs, NJ 07632, 1994.

[Heb49] D. O. Hebb. *The Organization of Behavior: A Neuropsychological Theory.* New York: Wiley, 1949.

[HKP91] John Hertz, Anders Krogh, and Richard G. Palmer. *Introduction to the Theory of Neural Computation*, volume Lecture notes of *Computation and neural systems.* Addison-Wesley Publishing Company, 1991.

[HL88] W. Y. Huang and R. P. Lippman. Neural net and traditional classifiers. In D. Z. Anderson, editor, *Neural Information Processing Systems*, pages 387--396. American Institute of Physics, 1988.

[Hop82] John J. Hopfield. Neural Networks and Physical Systems with Emergent Collective Computational Abilities. In Lau [Lau92], pages 142--146.

[JM94] Anil K. Jain and Jianchang Mao. *Neural Networks and Pattern Recognition*, chapter 5, pages 194--212. In Zurada et al. [ZMIR94], 1994.

[JM96] Anil K. Jain and Jianchang Mao. Artificial neural networks: A tutorial. *IEEE Computer*, pages 31--44, March 1996.

[JZ96] Anil K. Jain and Douglas Zongker. Feature selection: Evaluation, application, and small sample performance. *To Appear in IEEE Transactions on PAMI*, 1996.

[Koh82] Teuvo Kohonen. Clustering, taxonomy, and topological maps of patterns. In M. Lang, editor, *Proc. Sixth International Conference on Pattern Recognition.* Silver Spring, MD: IEEE Computer Society Press, 1982.

[Koh90] Teuvo Kohonen. The Self-Organizing Map. In Lau [Lau92], pages 74--90.

[Kru56] Joseph B. Kruskal, Jr. On the Shortest Spanning Subtree of a Graph and the Travelling Salesman Problem. In *Proceedings of the American Mathematical Society*, volume 7, pages 48--50, February 1956.

[Lab95] Alexandros Labrinidis. Methods to cluster transactions into utilization classes with similar workload characteristics. Master's thesis, Dept. of Comp. Science, Univ. of Crete, Greece, P.O. 1470, Heraklion, Crete, Greece, August 1995.
Also: Technical Report No. TR95-0135, ICS-FORTH, Heraklion, Crete, Greece.
(Both in Greek)
Note: ICS Technical Reports can be obtained through the ICS/Pleiades `Dienst` server at the URL: http://www.ics.forth.gr/TR.

[Lau92] Clifford G. Y. Lau, editor. *Neural Networks: Theoretical Foundations and Analysis.* IEEE Press, 1992.

[Lip87] R. P. Lippman. An Introduction to Computing with Neural Nets. In Lau [Lau92], pages 5--23.

[Llo82] S. P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, IT-28(2):129--137, March 1982.

[Mac67] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. 5th Berkeley Symposium on Mathematics, Statistics, and Probability*, volume 1, pages 281--298, 1967.

[Mar95] Manolis Marazakis. Simulation of transaction processing systems and a study of methods for performance goal satisfaction. Master's thesis, Dept. of Comp. Science, Univ. of Crete, Greece, P.O. 1470, Heraklion, Crete, Greece, October 1995.
Also: Technical Report No. TR95-0140, ICS-FORTH, Heraklion, Crete, Greece.
(Both in Greek)
Note: ICS Technical Reports can be obtained through the ICS/Pleiades `Dienst` server at the URL: http://www.ics.forth.gr/TR.

[MD89] John Moody and Christian J. Darken. Fast Learning in Network of Locally-Tuned Processing Units. *Neural Computation*, 1(2):281--294, 1989.

[MJ95] Jianchang Mao and Anil K. Jain. Artificial neural networks for feature extraction and multivariate data projection. *IEEE Transactions on Neural Networks*, 6(2):296 -- 317, March 1995.

[MMJ94] Jianchang Mao, K. Mohiuddin, and Anil K. Jain. Parsimonious network design and feature selection through node pruning. In *Proc. 12th International Conference on Pattern Recognition*, volume 2, pages 622 -- 624, Jerusalem, Israel, October 1994.

[MN89] Kurt Mehlhorn and Stefan Näher. LEDA: A Library of Efficient Data Types and Algorithms. Technical Report TR A 04/89, FB10, Universität des Saarlandes, Saarbrücken, 1989.
Can be obtained from the LEDA Library home page at the URL: ftp://ftp.mpi-sb.mpg.de/pub/LEDA/leda.html.

[MN95] Kurt Mehlhorn and Stefan Näher. LEDA: A Platform for Combinatorial and Geometric Computing. *Communications of the ACM*, 38(1):96--102, January 1995.

[MR90] Berndt Müller and Joachim Reinchardt. *Neural Networks: An Introduction*. Physics of Neural Networks. Springer-Verlag, 1990.

[MSW72] W. McCormick, P. Schweitzer, and T. White. Problem Decomposition and Data Reorganization by a Clustering Technique. *Operation Research*, 20, September/October 1972.

[Näh90] Stefan Näher. *LEDA 2.0 User Manual*. Fachbereich Informatik, Universität des Saarlandes, Saarbrücken, technischer bericht a 17/90 edition, 1990.
Can be obtained from the LEDA Library home page at the URL: ftp://ftp.mpi-sb.mpg.de/pub/LEDA/leda.html.

[Pri57] R. C. Prim. Shortest connection networks and some generalizations. *Bell System Technical Journal*, (36):1389--1401, 1957.

[RM86a] David E. Rumelhart and James L. McClelland. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume Foundations. MIT Press, 1986.

[RM86b] David E. Rumelhart and James L. McClelland. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume Psychological and biological models. MIT Press, 1986.

[RyC11] S. Ramón y Cajál. *Histologie du système nerveux de l' homme et des vertébrés.* Paris: Maloine; Edition Francaise Revue: Tome I, 1952; Tome II, 1955; Madrid: Consejo Superior de Investigaciones Cientificas, 1911.

[RZ86] David E. Rumelhart and David Zipser. *Feature Discovery by Competitive Learning*, chapter 5, pages 151--193. Volume Foundations of Parallel Distributed Processing [RM86a], 1986.

[SNI] Siemens Nixdorf Informationssysteme AG.
More information on Siemens Nixdorf Informationssysteme AG can be found at the URL: http://www.sni.de/.

[Tar83a] Robert Endre Tarjan. *Data Structures and Network Algorithms*, chapter 6, pages 71--83. Volume 44 of network-algorithms [Tar83b], 1983.

[Tar83b] Robert Endre Tarjan. *Data Structures and Network Algorithms*, volume 44 of *CBMS-NSF Regional Conference Series in Applied Mathematics.* Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, 1983.

[VR92] N. B. Venkateswarlu and P. S. V. S. K. Raju. Fast ISODATA Clustering Algorithms. *Pattern Recognition*, 25(3):335--342, 1992.

[Wer90] Paul J. Werbos. Backpropagation Through Time: What it Does and How to Do it. In Lau [Lau92], pages 211--221.

[WL90] Bernard Widrow and Michael A. Lehr. 30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation. In Lau [Lau92], pages 27--53.

[Zah71] Charles T. Zahn. Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters. *IEEE Transactions on Computers*, C-20(1):68 -- 86, January 1971.

[ZMIR94] J. M. Zurada, R. J. Marks II, and C. J. Robinson, editors. *Computational Intelligence: Imitating Life.* IEEE Press, 1994.