Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και
Υπολογιστών

Ευφυείς τεχνικές εξόρυξης δεδομένων για χρήσεις του διαδικτύου

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΔΗΜΗΤΡΙΟΣ ΚΟΥΤΣΟΥΚΟΣ

Επιβλέπων : Ανδρέας-Γεώργιος Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2016

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και
Υπολογιστών

Ευφυείς τεχνικές εξόρυξης δεδομένων για χρήσεις του διαδικτύου

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΔΗΜΗΤΡΙΟΣ ΚΟΥΤΣΟΥΚΟΣ

Επιβλέπων : Ανδρέας-Γεώργιος Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 22η Μαρτίου 2016.

Ανδρέας-Γεώργιος Σταφυλοπάτης Καθηγητής Ε.Μ.Π.	Στέφανος Κόλλιας Καθηγητής Ε.Μ.Π.	Γεώργιος Στάμου Επίκουρος Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2016

Δημήτριος Κουτσούκος

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Ο Παγκόσμιος Ιστός έχει πολύ μεγάλη ανάπτυξη στις μέρες μας. Εκατομμύρια σελίδες δέχονται επίσκεψη καθημερινά από δισεκατομμύρια χρήστες. Οι προσβάσεις τους καταγράφονται στα αρχεία καταγραφής των εξυπηρετητών. Η εξόρυξη χρήσεων του διαδικτύου εφαρμόζει τεχνικές εξόρυξης δεδομένων για να εξαγάγει την συμπεριφορά των χρηστών και να ανακαλύψει χρήσιμα μοτίβα πρόσβασης στο διαδίκτυο. Η ανακάλυψη αυτών των μοτίβων μπορεί να είναι χρήσιμη με μία πληθώρα τρόπων, όπως για παράδειγμα την εξατομίκευση μιας ιστοσελίδας, την προανάκληση συνδέσμων και τη βελτίωση της επίδοσης των εξυπηρετητών. Όμως, για την ανακάλυψη μοτίβων τα αρχεία καταγραφής πρέπει να υποστούν προεπεξεργασία προκειμένου να αφαιρεθεί ο “θόρυβος”. Σε αυτή τη διπλωματική εργασία, εξερευνάται η φάση της προεπεξεργασίας των δεδομένων και προτείνεται ένας νέος αλγόριθμος για την αναγνώριση της συνεδρίας χρήστη, που χρησιμοποιεί την ασαφή συσταδοποίηση c-κέντρων. Έπειτα, γίνεται μια έρευνα στους τρόπους που μπορούν να εξαχθούν μοτίβα και εφαρμόζεται η εξόρυξη κανόνων συσχέτισης σε πραγματικά αρχεία καταγραφής για την εξαγωγή ουσιωδών κανόνων προκειμένου να προβλεφθεί το επόμενο αίτημα ενός χρήστη από τα προηγούμενά του.

Λέξεις κλειδιά

εξόρυξη χρήσεων του διαδικτύου, αναγνώριση συνεδρίας χρήστη, εξόρυξη κανόνων συσχέτισης, εξόρυξη συχνών συνόλων αντικειμένων, ανακάλυψη μοτίβων, εξόρυξη δεδομένων

Abstract

World Wide Web has an enormous growth during these days. Millions of pages are added daily and billions of users access them. Their accesses are recorded in web server logs. Web Usage Mining applies data mining techniques in server logs in order to extract the behaviour of users and discover web access patterns. Discovering these patterns can be useful in a number of ways such as personalizing a website, prefetching links and improving the web server performance. However, for pattern discovery the web logs have to be preprocessed in order to remove “noise”. In this diploma thesis, the preprocessing phase is being explored and a new algorithm for session identification using Fuzzy C-Means Clustering is being proposed. Following, a survey on the techniques of pattern discovery is being done and association rule mining is being applied on real web logs in order to extract meaningful rules and to “guess” a user’s next request based on his previous ones.

Key words

web usage mining, session identification, association rule mining, frequent itemset mining, pattern discovery, data mining

Ευχαριστίες

Αυτή η διπλωματική εργασία σηματοδοτεί το τέλος της ακαδημαϊκής μου πορείας στο Εθνικό Μετσόβιο Πολυτεχνείο και ταυτόχρονα το τέλος μιας σημαντικής περιόδου της ζωής μου και για αυτό θα ήθελα να ευχαριστήσω όσα άτομα με βοήθησαν να φτάσω ως εδώ.

Καταρχάς, οφείλω να ευχαριστήσω τον κ. Ανδρέα-Γεώργιο Σταφυλοπάτη, καθηγητή ΕΜΠ, για την ευκαιρία που μου προσέφερε να εκπονήσω αυτή τη διπλωματική εργασία, τον κ. Γεώργιο Αλεξανδρίδη, διδάκτορα ΕΜΠ για την βοήθειά του σε όλα τα στάδια της εργασίας και τους κ.κ. Στέφανο Κόλλια, καθηγητή ΕΜΠ, και Γεώργιο Στάμου, επίκουρο καθηγητή ΕΜΠ, για την τιμή που μου κάνανε να είναι μέλη της επιτροπής εξέτασης της διπλωματικής μου εργασίας. Επίσης θα ήθελα να ευχαριστήσω την οικογένειά μου που μου συμπαραστάθηκε όλα αυτά τα χρόνια καθώς και τους Άντα, Βασίλη, Γιάννη, Παναγιώτη, Σοφία, Βασίλη Μ., Χρήστο, Αντώνη, Μιχάλη Βρ., Μιχάλη Λ., Ελεάννα, Τάνια, Κατερίνα, Βασιλική, Ελένη Μ., Νίκο, Ελένη Ψ., Βασίλη Γ., Ορέστη, Χλόη και Γιώργο. Τέλος, ένα ξεχωριστό ευχαριστώ στους Νικόλα και Διονύση με τους οποίους η συνεργασία στο μάθημα των μεταγλωττιστών με βοήθησε να γίνω καλύτερος και να προσπαθώ πάντα για το τέλειο.

Δημήτριος Κουτσούκος,

Αθήνα, 22η Μαρτίου 2016

Περιεχόμενα

Κατάλογος πινάκων

Κατάλογος σχημάτων

Κατάλογος Αλγορίθμων

Κατάλογος αρκτικόλεξων

Αρκτικόλεξο	Επεξήγηση
W3C	World Wide Web Consortium
ISP	Internet Service Provider
CLF	Common Log Format
ECLF	Extended Common Log Format
URL	Uniform Resource Locator
URI	Uniform Resource Identifier
VPC	Validity Partition Coefficient
VPE	Validity Partition Entropy
VMPC	Validity Modified Partition Coefficient
VXB	Validity Xie Beni
VFS	Validity Fukuyama Sugeno
VK	Validity Kwon
VT	Valitity Tang
VPCAES	Validity Partition Coefficient And Exponential Separation
ART1NN	Adaptive Resonance Theory1 Neural Network
SOM	Self-Organizing Maps
HMM	Hidden Markov Modesl
SVM	Support Vector Machines
kNN	k-Nearest Neighbours
TFIDF	Term Frequency–Inverse Document Frequency
CF	Collaborative Filtering
EDA	Exploratory Data Analysis

Κεφάλαιο 1

Εισαγωγή

1.1 Γενική επισκόπηση της Εξόρυξης Χρήσεων του Διαδικτύου

1.1.1 Διαδίκτυο

Ο παγκόσμιος ιστός έχει επηρεάσει κάθε πτυχή της ζωή μας. Με 4.84 δισεκατομμύρια σελίδες και 3.366 δισεκατομμύρια χρήστες έχει αναχθεί πλέον στην κύρια πηγή πληροφόρησης για σχεδόν όλο τον κόσμο. Το διαδίκτυο, πέρα από πηγή πληροφορίας, έχει αναχθεί σε μέσο διασκέδασης αλλά έχει λάβει και οικονομική, πολιτική και κοινωνική διάσταση. Υπάρχουν εκατομμύρια καταστήματα που είναι προσβάσιμα μόνο διαδικτυακά, αλλά και πολλά που παρότι έχουν φυσική μορφή, έχουν τη δική τους ιστοσελίδα για λόγους προώθησης. Ακόμη, η επικοινωνία μας με άλλους ανθρώπους έχει γίνει πολύ πιο εύκολη καθώς μπορούμε μέσω του διαδικτύου να μιλήσουμε ή να ανταλλάξουμε απόψεις με ανθρώπους που βρίσκονται στην άλλη άκρη του κόσμου. Θα μπορούσαμε να πούμε ότι πλέον το διαδίκτυο έχει αναχθεί σε μία “εικονική κοινωνία”.

1.1.2 Εξαγωγή χρήσεων του διαδικτύου

Η ραγδαία ανάπτυξη του διαδικτύου το έχει κάνει μία από τις πιο ενδιαφέρουσες πηγές εξόρυξης δεδομένων, λόγω κάποιων ιδιαίτερων χαρακτηριστικών του [6].

- Η πληροφορία του διαδικτύου αυξάνεται ολοένα και περισσότερο καθημερινά. Το εύρος της είναι απεριόριστο καθώς ο καθένας μπορεί να βρει πληροφορίες για οτιδήποτε.
- Τα δεδομένα που βρίσκει κανείς στο διαδίκτυο μπορεί να έχουν οποιαδήποτε μορφή.
- Η πληροφορία του διαδικτύου είναι ετερογενής, υπό την έννοια ότι πολλές φορές μερικές σελίδες μπορεί να έχουν το ίδιο περιεχόμενο, αλλά με άλλη μορφή ή λέξεις.
- Ένα μεγάλο μέρος της πληροφορίας που υπάρχει στο διαδίκτυο είναι συνδεδεμένη μεταξύ της. Οι σελίδες που αναφέρονται πολύ συχνά από άλλες θεωρούνται ότι είναι υψηλής ποιότητας, μιας και από ότι φαίνεται η κοινότητα των χρηστών έχει μεγάλη εμπιστοσύνη σε αυτές.
- Η πληροφορία που περιέχεται στον Παγκόσμιο Ιστό είναι θορυβώδης υπό την έννοια ότι δεν είναι σε όλες τις περιστάσεις ακριβής. Από την άλλη, επειδή υπάρχει ελευθερία λόγου, πολλές φορές μεγάλη ποσότητα πληροφορίας είναι λανθασμένη, αποπροσανατολιστική ή χαμηλής ποιότητας.

Όλα αυτά τα χαρακτηριστικά έχουν συμβάλλει στην ανάδυση ενός νέου επιστημονικού αντικειμένου, το οποίο πραγματεύεται την ανακάλυψη πληροφοριών και γνώσης από τον Παγκόσμιο Ιστό.

Η Εξόρυξη Χρήσεων του Διαδικτύου (Web Usage Mining) είναι υποκατηγορία της Εξόρυξης του Διαδικτύου (Web Mining) η οποία με τη σειρά της είναι υποκατηγορία της Εξόρυξης Δεδομένων.

Η εξόρυξη δεδομένων μπορεί να οριστεί ως η ανακάλυψη χρήσιμων μοτίβων ή γνώσης από πηγές δεδομένων[7]. Είναι ένα ευρύ επιστημονικό αντικείμενο που περιλαμβάνει πολλά άλλα όπως για παράδειγμα τη μηχανική μάθηση ή την τεχνητή νοημοσύνη. Σχετίζεται επίσης με τη επιβλεπόμενη μάθηση (ή κατηγοριοποίηση), την μη-επιβλεπόμενη μάθηση (ή συσταδοποίηση), την εξαγωγή κανόνων συσχέτισης αλλά και την εξαγωγή ακολουθιακών μοτίβων. Η διαδικασία της Εξόρυξης Δεδομένων μπορεί να χωριστεί σε 3 βασικά βήματα [7]:

- **Προπεξεργασία των δεδομένων**: Τα δεδομένα στην αρχική τους μορφή πολλές φορές δεν είναι κατάλληλα γιατί μπορεί να περιέχουν χαρακτηριστικά τα οποία δεν χρειαζόμαστε ή να είναι πολύ μεγάλα σε μέγεθος.

- **Εξόρυξη Δεδομένων**: Τα επεξεργασμένα δεδομένα δίνονται ως είσοδος σε αλγορίθμους Εξόρυξης Δεδομένων, οι οποίοι θα εξάγουν μοτίβα ή γνώση.

- **Μετεπεξεργασία**: Μετά την ανακάλυψη των μοτίβων θα πρέπει να γίνει μια επισκόπησή τους προκειμένου να βρούμε αυτά τα οποία είναι χρήσιμα για την εφαρμογή που εξετάζουμε. Πολλές φορές σε αυτό το βήμα χρειάζεται ανθρώπινη παρέμβαση για να βρούμε τα "χρήσιμα" μοτίβα.

Η εξόρυξη δεδομένων η οποία σχετίζεται με τον Παγκόσμιο Ιστό έχει ως σκοπό την εξαγωγή χρήσιμης πληροφορίας από 4 πηγές [1]:

- **Περιεχόμενο ιστοσελίδων**: Αυτή η κατηγορία περιέχει τα *πραγματικά* δεδομένα μιας ιστοσελίδας δηλαδή αυτά που απευθύνονται στους χρήστες, όπως για παράδειγμα είναι το κείμενο και οι εικόνες.

- **Δομή**: Εδώ περιέχονται τα δεδομένα που έχουν να κάνουν με την οργάνωση του περιεχομένου, όπως για παράδειγμα τα διάφορα HTML ή XML tags. Η κύρια πηγή από δομική πληροφορία μεταξύ διαφορετικών σελίδων είναι οι υπερσύνδεσμοι που συνδέουν αυτές τις σελίδες.

- **Χρήση**: Δεδομένα τα οποία περιγράφουν το μοτίβο της χρήσης μιας ιστοσελίδας, όπως είναι για παράδειγμα ο χρόνος και η ημερομηνία των προσβάσεων, οι IP διευθύνσεις, ανήκουν σε αυτή την κατηγορία.

- **Προφίλ Χρήστη**: Σε αυτή την κατηγορία περιέχονται δεδομένα τα οποία δίνουν δημογραφικά χαρακτηριστικά σχετικά με τους επισκέπτες μιας ιστοσελίδας.

Η εξόρυξη χρήσεων του διαδικτύου αφορά αποκλειστικά την 3η κατηγορία δεδομένων. Χρησιμοποιώντας κάποιες από τις πιο διαδεδομένες τεχνικές της εξόρυξης δεδομένων [8], όπως είναι η συσταδοποίηση και η εξόρυξη κανόνων συσχέτισης, μπορούμε να βρούμε σχέσεις μεταξύ των χρηστών ανάλογα με τα ενδιαφέροντά τους αλλά και τα μοτίβα με τα οποία πλοηγούνται σε διάφορες ιστοσελίδες. Αυτή η μελέτη μπορεί να έχει πολλές εφαρμογές όπως είναι για παράδειγμα η πρόταση συγκεκριμένων συνδέσμων ή προϊόντων ανάλογα με τις προτιμήσεις ενός χρήστη. Άλλωστε τα συστήματα συστάσεων είναι μία από τις πιο γνωστές εφαρμογές της εξόρυξης των χρήσεων του

διαδικτύου. Μία ακόμα χρήση είναι η καλύτερη οργάνωση των ιστοσελίδων από τους προγραμματιστές αναλόγως με το ποια μονοπάτια ακολουθούν οι περισσότεροι χρήστες ανάμεσα στις σελίδες κατά την πλοήγησή τους.

Σύμφωνα με τον Srivastava [1], η διαδικασία της εξόρυξης δεδομένων χρήσεων του διαδικτύου μπορεί να χωριστεί σε 4 στάδια:

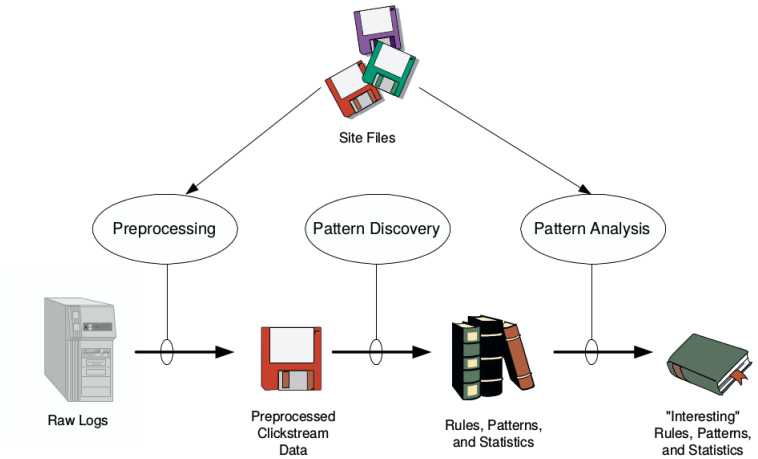
1. **Στάδιο εισόδου**: Στο στάδιο εισόδου συλλέγονται αρχεία καταγραφής από εξυπηρετητές καθώς και η τοπολογία ενός ιστοτόπου.

2. **Στάδιο προπεξεργασίας**: Τα αρχεία καταγραφής στην αρχική τους μορφή δεν είναι κατάλληλα για την εφαρμογή αλγορίθμων εξόρυξης δεδομένων. Για αυτό απαιτείται κάποια προεπεξεργασία, προκειμένου τα δεδομένα να "καθαριστούν", δηλαδή να αφαιρεθούν spiders /bots /crawlers, να αναγνωριστούν οι μοναδικοί επισκέπτες της ιστοσελίδας αλλά και τα session τους και τέλος να συμπληρωθεί η διαδρομή που έχουν ακολουθήσει κατά την πλοήγησή τους.

3. **Στάδιο ανακάλυψης μοτίβων**: Τα προεπεξεργασμένα δεδομένα είναι έτοιμα για την εφαρμογή αλγορίθμων Εξόρυξης Δεδομένων για την ανακάλυψη μοτίβων. Τέτοιοι αλγόριθμοι είναι αυτοί που πραγματοποιούν:

    - Στατιστική ανάλυση
    - Συσταδοποίηση
    - Εξαγωγή κανόνων συσχέτισης
    - Κατηγοριοποίηση
    - Εξαγωγή ακολουθιακών μοτίβων

4. **Στάδιο ανάλυσης μοτίβων**: Σε αυτό το στάδιο εξετάζονται τα ανακαλυφθέντα μοτίβα προκειμένου να διαχωριστούν αυτά τα οποία δεν περιέχουν ουσιώδη πληροφορία από αυτά που είναι πιο ενδιαφέροντα.



Σχήμα 1.1: Αφαιρετική προσέγγιση της εξόρυξης χρήσεων του Διαδικτύου[1]

Τέλος θα πρέπει να αναφερθεί ότι η εξαγωγή χρήσεων του διαδικτύου είναι γνωστή και ως clickstream analysis. Ως clickstream ορίζουμε μια σειρά από clicks τα οποία έχουν γίνει από κάποιον κατά τη

διάρκεια χρήσης μιας ή πολλών ιστοσελίδων. Μια τέτοια ακολουθία μπορεί να περιέχει, πέρα από προβολές ιστοσελίδων, εικόνες ή αρχεία .css τα οποία "φορτώνονται" μαζί με την ιστοσελίδα. Αν αυτές οι προβολές σελίδων συνδεθούν σε ένα session τότε οι αναλυτές, μελετώντας αρκετές τέτοιες ακολουθίες, μπορούν να απαντήσουν ερωτήσεις όπως:

- Από πού εισέρχονται στην ιστοσελίδα οι περισσότεροι χρήστες;
- Με ποια σειρά οι χρήστες βλέπουν τις σελίδες;
- Ποιες ιστοσελίδες προσπελάζονται συχνά μαζί;
- Πόσες σελίδες έχει κατά μέσο όρο μια τυπική επίσκεψη;
- Πόσο χρόνο αφιερώνει ένας μέσος χρήστης σε μια μοναδική ιστοσελίδα;
- Από πού εξέρχονται από την ιστοσελίδα οι περισσότεροι χρήστες;

1.2 Οργάνωση κειμένου

Στο Κεφάλαιο 2 παρουσιάζονται οι απαιτήσεις προκειμένου να "καθαριστούν" τα δεδομένα καθώς και οι συνήθεις μορφές τους, όπως και οι τρόποι με τους οποίους αναγνωρίζεται ένας μοναδικός χρήστης.

Στο Κεφάλαιο 3 παρουσιάζονται οι κατευθύνσεις με τις οποίες γίνεται η αναγνώριση ενός session. Στη συνέχεια προτείνεται η δική μας μέθοδος για την αναγνώριση ενός session αφού πρώτα γίνει αναφορά στις προαπαιτούμενες γνώσεις που χρειάζονται για τον αλγόριθμό μας. Στη συνέχεια παρουσιάζονται κάποιες μετρήσεις αλλά και αποτελέσματα της μεθόδου μας.

Στο Κεφάλαιο 4 γίνεται μια γενική επισκόπηση της ανακάλυψης μοτίβων και πιο συγκεκριμένα της στατιστικής ανάλυσης.

Στο Κεφάλαιο 5 αναλύεται η εξαγωγή κανόνων συσχέτισης. Παρουσιάζονται κάποιες προαπαιτούμενες έννοιες αλλά και κάποιοι αλγόριθμοι οι οποίοι είναι αρκετά διαδεδομένοι και στη συνέχεια γίνεται εφαρμογή τους και αναλύονται τα αποτελέσματα.

Στο Κεφάλαιο 6 παρουσιάζουμε τη συνεισφορά και τα συμπεράσματα της εργασίας μας όπως και κάποιες μελλοντικές επεκτάσεις στο κομμάτι της ανακάλυψης μοτίβων.

Κεφάλαιο 2

Προεπεξεργασία δεδομένων

2.1 Εισαγωγή

Η προεπεξεργασία των δεδομένων ή αλλιώς data preprocessing που γίνεται στα αρχεία καταγραφής εξυπηρετητών (web server logs) είναι ένα από τα πιο σημαντικά κομμάτια της διαδικασίας εξόρυξης χρήσεων του διαδικτύου. Αποτελείται από διάφορα στάδια καθένα από τα οποία είναι κρίσιμο, διότι τα δεδομένα που προκύπτουν ως έξοδος αυτού του σταδίου, αποτελούν την είσοδο των αλγορίθμων εξόρυξης δεδομένων για την ανακάλυψη μοτίβων και χρήσιμων συμπερασμάτων. Με την προεπεξεργασία καταφέρνουμε να διώξουμε τον "θόρυβο" από τα δεδομένα μας. Πιο συγκεκριμένα, η προπεξεργασία των δεδομένων είναι απαραίτητη προκειμένου:

- Να καθαρίσουμε τα δεδομένα από εικόνες ή άλλα αρχεία που δεν έχουν ζητηθεί ρητώς από το χρήστη.
- Να αφαιρέσουμε τα requests που έχουν γίνει από spiders/crawlers/bots τα οποία δημιουργούν πολύ μεγάλη κίνηση σε μικρό χρονικό διάστημα κατεβάζοντας ολόκληρη την ιστοσελίδα ή μεγάλα κομμάτια της και επομένως η συμπεριφορά τους δεν παρουσιάζει κάποιο ενδιαφέρον.
- Να αναγνωρίσουμε κάθε χρήστη, καθώς η περιήγηση στον παγκόσμιο ιστό γίνεται ανώνυμα και θα πρέπει να χρησιμοποιήσουμε ένα συνδυασμό στοιχείων όπως για παράδειγμα IP, Λειτουργικό Σύστημα, έκδοση περιηγητή, cookies και πληροφορίες εγγραφής.
- Να αναγνωρίσουμε το κάθε session ενός χρήστη, το οποίο ορίζεται ως το σύνολο των σελίδων που ζητήθηκαν από έναν επισκέπτη του site για μία συγκεκριμένη περίοδο, η σειρά με την οποία ζητήθηκαν και η διάρκεια που ο χρήστης "πέρασε" σε μία σελίδα.

Στο συγκεκριμένο κεφάλαιο θα μελετήσουμε τα τρία πρώτα κομμάτια της προεπεξεργασίας των δεδομένων. Πρώτα όμως, θα κάνουμε μια επισκόπηση της μορφής των αρχείων καταγραφής των εξυπηρετητών προκειμένου αφενός να καταλάβουμε πώς χρειάζεται να παρέμβουμε στα δεδομένα και αφετέρου πόσο σημαντική είναι η προπεξεργασία τους.

2.2 Αρχεία καταγραφής εξυπηρετητών

Οι εξυπηρετητές του Παγκοσμίου Ιστού διατηρούν ένα ή περισσότερα αρχεία τα οποία καταγράφουν τις δραστηριότητες των επισκεπτών τους. Πιο συγκεκριμένα, σε αυτά τα αρχεία βρίσκεται ένα ιστορικό από τις σελίδες που έχουν ζητηθεί από τον εξυπηρετητή. Ο οργανισμός W3C[9] διατηρεί μία τυποποιημένη μορφή για τα αρχεία καταγραφής εξυπηρετητών του παγκοσμίου ιστού

η οποία ονομάζεται κοινή μορφή αρχείου καταγραφής (Common Log Format - CLF) και αποτελείται από διάφορα πεδία τα οποία θα αναλύσουμε στη συνέχεια.

2.2.1 Common Log Format

Το Common Log Format (CLF[10]) είναι μια τυποποιημένη μορφή που χρησιμοποιείται από τους εξυπηρετητές παγκοσμίου ιστού για τα αρχεία καταγραφής τους. Μια συνηθισμένη μορφή κάθε γραμμής του CLF είναι η εξής:

Common Log Format fields	Short Explanation
Host	IP address or domain name of the user who made the request to the server
Ident	Identification information
Authuser	Authentication information
Date	Date and time the request was made
Request	The request made by the client
Status	HTTP(S) status code
Size	Size of the object returned to the client in bytes

Πίνακας 2.1: Πίνακας εξήγησης του CLF

ενώ ένα παράδειγμα που θα μπορούσε να υπάρχει σε έναν πραγματικό εξυπηρετητή είναι το εξής:

127.0.0.1 user-identifier frank [10/Oct/2000:13:55:36 -0700] ”GET /index.html HTTP/1.0” 200 2326
127.0.0.1 user-identifier frank [10/Oct/2000:13:55:36 -0700] ”GET /styling.css HTTP/1.0” 200 336
127.0.0.1 user-identifier frank [10/Oct/2000:13:55:36 -0700] ”GET /myscript.js HTTP/1.0” 200 2473
127.0.0.1 user-identifier frank [10/Oct/2000:13:55:36 -0700] ”GET /image1.gif HTTP/1.0” 200 243
127.0.0.1 user-identifier frank [10/Oct/2000:13:55:36 -0700] ”GET /image2.png HTTP/1.0” 200 233

Πίνακας 2.2: Παράδειγμα CLF

Τα επιμέρους πεδία του CLF είναι:

- **Πεδίο διεύθυνσης επισκέπτη**: Αυτό το πεδίο περιλαμβάνει την IP διεύθυνση του επισκέπτη της ιστοσελίδας ο οποίος ζητάει να δει μια συγκεκριμένη σελίδα.
- **Πεδίο Ταυτοποίησης**: Περιέχει στοιχεία ταυτοποίησης του χρήστη της ιστοσελίδας αν ο εξυπηρετητής έχει ρυθμιστεί κατάλληλα.
- **Πεδίο Authuser**: Σε περίπτωση που κάποιο κομμάτι της ιστοσελίδας χρειάζεται πιστοποίηση, τότε αυτό το πεδίο περιέχει το όνομα του χρήστη που είχε πρόσβαση. Σε περίπτωση που είναι κενό, περιέχει και σε αυτό παύλα.
- **Πεδίο Ημερομηνίας/Ώρας**: Εδώ δίνεται η μορφή της ημερομηνίας και της ώρας καθώς η ζώνη ώρας. Κάθε server έχει διαφορετική μορφή για τον τρόπο και τα στοιχεία που υπάρχουν σε αυτό το πεδίο, αλλά η πιο συνηθισμένη μορφή είναι η %d/%b/%Y:%H:%M:%S %z, δηλαδή δίνονται με τη σειρά η μέρα (d), ο μήνας (b), η χρονιά (Y), η ώρα(H), τα λεπτά (M), τα δευτερόλεπτα (S) και τέλος πόσες ώρες βρίσκεται πριν ή μετά ο διακομιστής από τη ζώνη ώρας GMT (z).

- **Πεδίο αιτήματος HTTP/HTTPS**: Αποτελείται από το αίτημα που έκανε ο περιηγητής του επισκέπτη της ιστοσελίδας στον εξυπηρετητή. Αυτό το πεδίο περιέχεται μέσα σε εισαγωγικά και μπορεί να διαχωριστεί στα ακόλουθα κομμάτια:
  - Ο τρόπος που έγινε το αίτημα από το πρόγραμμα περιήγησης (π. χ. GET, POST).
  - Το κομμάτι της ιστοσελίδας που ζητήθηκε, γνωστό και ως Uniform Resource Identifier (URI).
  - Το πρωτόκολλο με το οποίο ζητήθηκε περιεχόμενο από τον εξυπηρετητή (HTTP, HTTPS, FTP κ.α.).
- **Πεδίο Κωδικού Κατάστασης**: Το πεδίο αυτό περιέχει έναν τριψήφιο κωδικό ο οποίος δείχνει την κατάσταση του αιτήματος, δηλαδή το αν πέτυχε ή όχι και αν όχι τι είδους σφάλμα προκλήθηκε. Η πλήρης λίστα με τους κωδικούς κατάστασης [11] δείχνει ότι οι επιτυχείς κωδικοί έχουν τη μορφή "2xx", οι κωδικοί σφάλματος από την πλευρά του επισκέπτη της ιστοσελίδας τη μορφή "4xx", οι κωδικοί ανακατεύθυνσης, που απαιτούν επιπλέον κινήσεις από την πλευρά του εξυπηρετητή για την εξυπηρέτηση του αιτήματος, τη μορφή "3xx" και τέλος οι κωδικοί που υποδεικνύουν κάποιο σφάλμα από την πλευρά του εξυπηρετητή έχουν τη μορφή "5xx".
- **Πεδίο όγκου μεταφερομένων δεδομένων**: Περιέχει την ποσότητα των δεδομένων σε bytes που μεταφέρθηκαν σε περίπτωση που έχουμε επιτυχές αίτημα της μορφής GET. Σε κάθε άλλη περίπτωση αυτό το πεδίο περιέχει την τιμή 0.

2.2.2 Extended Common Log Format

Επέκταση του Common Log Format είναι το Extended Common Log Format ή αλλιώς ECLF το οποίο περιέχει 2 παραπάνω πεδία σε σχέση με το CLF.

- **Πεδίο αναφοράς**: Αυτό το πεδίο περιέχει το URL της σελίδας προέλευσης κάποιου χρήστη μιας ιστοσελίδας. Σε περίπτωση που δεν υπάρχει η πληροφορία το πεδίο έχει παύλα.
- **Πεδίο μέσου του χρήστη**: Εδώ περιέχονται πληροφορίες όπως είναι η έκδοση του προγράμματος περιήγησης του επισκέπτη, το λειτουργικό του σύστημα και αν ο χρήστης είναι κάποιο web crawler.

2.3 Καθαρισμός και φιλτράρισμα δεδομένων

Τα προαναφερθέντα πεδία από τα αρχεία καταγραφής πρέπει να έρθουν σε συγκεκριμένη μορφή προκειμένου να μπορούν να υποστούν επεξεργασία και ανάλυση. Καταρχάς, μερικά πεδία περιέχουν παραπάνω πληροφορίες από κάποια άλλα και κατά δεύτερον κάποια πεδία περιέχουν περισσότερες μεταβλητές ως προς την ανάλυση που θέλουμε να κάνουμε. Ένα τέτοιο πεδίο είναι αυτό της Ημερομηνίας/Ώρας αλλά και αυτό του αιτήματος HTTP/HTTPS το οποίο περιέχει τον τρόπο με τον οποίο έγινε η αίτηση, το κομμάτι της ιστοσελίδας που ζητήθηκε, αλλά και το πρωτόκολλο που έγινε αυτό. Προκειμένου να αναλύσουμε τα δεδομένα μας πρέπει να προβούμε στην απαραίτητη εξαγωγή μεταβλητών από το αρχείο καταγραφής και πιο αναλυτικά πρέπει να βρούμε τα εξής:

1. Την ημερομηνία που έγινε το αίτημα.
2. Την ώρα που έγινε το αίτημα.
3. Τον τρόπο με τον οποίο έγινε το αίτημα.
4. Το κομμάτι της σελίδας που ζητήθηκε.
5. Το πρωτόκολλο με το οποίο ζητήθηκε η σελίδα.

Στη συνέχεια οργανώνουμε τα αιτήματα με βάση το χρόνο. Έχοντας ως αναφορά μια συγκεκριμένη ημερομηνία και ώρα που βρίσκεται μετά το χρόνο του τελευταίου αιτήματος στο αρχείο καταγραφής και θεωρώντας την ως χρόνο βάσης, βρίσκουμε τον απόλυτο χρόνο κάθε αιτήματος. Στη συνέχεια, ταξινομούμε τα αιτήματα σε φθίνουσα σειρά ως προς το χρόνο βάσης.

Ακόμα και μετά από αυτή την προεπεξεργασία το αρχείο καταγραφής δεν είναι έτοιμο για εξαγωγή συμπερασμάτων, ακόμα και της πιο απλής μορφής. Για παράδειγμα, θα πρέπει να εξετάσουμε πόσα από τα αιτήματα που έχουν καταγραφεί έχουν γίνει πραγματικά από επισκέπτες της σελίδας και δεν έχουν προκληθεί αυτόματα από την κλήση μιας συγκεκριμένης σελίδας και είναι π. χ. αρχεία εικόνων.

Αρχικά θα πρέπει να χωρίσουμε τα δεδομένα ανάλογα με τη μέθοδο με την οποία ζητήθηκαν. Οι πιο συνηθισμένες μέθοδοι είναι οι GET, POST οπότε κρατάμε όσα αιτήματα έχουν γίνει με αυτές τις δύο μεθόδους. Στη συνέχεια, πρέπει να δούμε τους τύπους των αρχείων προκειμένου να αποφασίσουμε ποιοι από αυτούς δεν ζητήθηκαν ρητά από τον επισκέπτη της ιστοσελίδας. Αρχεία όπως εικόνες (.jpg, .gif, .png, .jpeg) ή μορφοποίησης (.css) ή scripts (.js) συνήθως δεν ζητούνται άμεσα από το χρήστη και για αυτό θα πρέπει να αφαιρεθούν. Βέβαια, αρχεία εικόνων μπορεί να προδώσουν περιπτώσεις hotlinking[12] και να είναι χρήσιμες για κάποιες εφαρμογές της εξόρυξης δεδομένων. Άλλες περιπτώσεις στις οποίες μπορεί να κρατηθούν τα αιτήματα για εικόνες ή κάποιου είδους από τα προαναφερθέντα αρχεία στο αρχείο καταγραφής είναι σε περίπτωση που η ιστοσελίδα που βρισκόμαστε θεωρεί ότι αυτά ανήκουν στο περιεχόμενο που θέλει να προσφέρει και επομένως θέλει να δει πόσοι επισκέπτες "κατέβασαν" τα συγκεκριμένα αρχεία, ή σε περίπτωση που ένας αναλυτής ενδιαφέρεται να μελετήσει την κατανάλωση εύρους ζώνης στην ιστοσελίδα.

Αυτή η διαδικασία που περιγράψαμε παραπάνω, είναι γνωστή ως φιλτράρισμα διότι αφαιρούνται από το αρχείο καταγραφής όλα τα αρχεία που δεν μας ενδιαφέρουν προκειμένου να μείνουν τα αρχεία που είναι χρήσιμα για ανάλυση. Μετά από αυτό το βήμα το αρχείο καταγραφής είναι έτοιμο για το επόμενο στάδιο, που είναι η αφαίρεση spiders/bots/crawlers.

2.4 Αφαίρεση crawlers/bots/spiders

Ως bot ή spider ή crawler, ορίζουμε αυτοματοποιημένα πρόγραμμα και όχι ανθρώπους τα οποία κατεβάζουν μαζικά κομμάτια ιστοσελίδων για διάφορους σκοπούς. Τέτοια προγράμματα είναι για παράδειγμα οι μηχανές αναζήτησης. Πέρα από τις μηχανές αναζήτησης, αυτά τα είδη προγραμμάτων μπορεί να ελέγχουν μια ιστοσελίδα ή ένα ιστολόγιο προκειμένου να ενημερώσουν κάποιο χρήστη για νέο περιεχόμενο. Άλλοτε είναι ρυθμισμένα από κάποιον διαχειριστή μιας σελίδας προκειμένου να δει αν είναι λειτουργική ή αν ο χρόνος απόκρισης είναι φυσιολογικός ή προκειμένου να εξάγουν στατιστικά συμπεράσματα για την επισκεψιμότητά της (Google Analytics). Υπάρχουν

κατηγορίες bots/spiders/crawlers τα οποία μπορούν να φανούν στο ECLF στο Πεδίο Μέσου του Χρήστη (User Agent Field). Μερικές φορές, για λόγους επικοινωνίας, μερικά bots έχουν κάποιο URL ή κάποια διεύθυνση ηλεκτρονικού ταχυδρομείου. Υπάρχει ένα ειδικό αρχείο σε κάθε εξυπηρετητή που ονομάζεται "robots.txt", που είναι το πρώτο που προσπελάζεται από bots προκειμένου οι διαχειριστές να καταλάβουν ότι ο χρήστης που προσπελαύνει τη σελίδα είναι πρόγραμμα και όχι άνθρωπος. Επομένως, κάποιος ο οποίος έχει στην κατοχή του αρχεία καταγραφής εξυπηρετητών του Παγκοσμίου Ιστού μπορεί με τους εξής δύο τρόπους να βρει τις αιτήσεις για ιστοσελίδες που έχουν γίνει από bots/crawlers/spiders και να τα αφαιρέσει από το αρχείο καταγραφής:

- Αν το User Agent Field προδίδει την παρουσία ενός bot τότε η συγκεκριμένη αίτηση καθώς και όσες αυτή "πυροδότησε", όπως για παράδειγμα αρχεία .css, .js ή εικόνες, αφαιρούνται.

- Αν κάποιος έχει ζητήσει κατά το πρώτο αίτημά του στον εξυπηρετητή το αρχείο "robots.txt", τότε τα επακόλουθα αιτήματα που μπορεί να συσχετιστούν με αυτό το αίτημα είτε μέσω της IP, είτε χρονικά, είτε μέσω του User Agent Field αφαιρούνται από το αρχείο καταγραφής.

Η αφαίρεση αυτών των requests είναι ιδιαιτέρως σημαντική καθώς επηρεάζουν σε μεγάλο βαθμό το αρχείο καταγραφής. Κατ' αρχάς, η συμπεριφορά τους δεν είναι ενδιαφέρουσα από την άποψη της εξόρυξης χρήσεων του διαδικτύου καθώς "κατεβάζουν" τη μία σελίδα μετά την άλλη, μέχρι να "κατεβάσουν" όλες τις σελίδες ενός ιστοτόπου. Επίσης, επηρεάζουν τα διάφορα στατιστικά της ιστοσελίδας σχετικά με την κίνηση της ιστοσελίδας, την επισκεψιμότητα αλλά και τον αριθμό των sessions που έχουν οι επισκέπτες. Υπάρχουν περιπτώσεις που μερικά bots μπορεί να μιμηθούν τη συμπεριφορά ενός συγκεκριμένου ακροατηρίου, οδηγώντας σε λάθος συμπεράσματα τους αλγορίθμους εξόρυξης δεδομένων. Για όλους αυτούς τους λόγους θα πρέπει να είμαστε σίγουροι ότι αυτές οι μη-ανθρώπινες προσβάσεις έχουν αφαιρεθεί από το αρχείο καταγραφής πριν προχωρήσουμε στα επόμενα βήματα.

2.5 Αναγνώριση χρήστη

Τα προηγούμενα βήματα της διαδικασίας της προεπεξεργασίας μπορούν να θεωρηθούν κατά μία έννοια "ντετερμινιστικά" διότι αυτό που πρέπει να γίνει προκειμένου να εκπληρωθούν στις περισσότερες των περιπτώσεων είναι σαφώς ορισμένο. Τα επόμενα βήματα, όπως είναι η αναγνώριση χρήστη και η αναγνώριση συνεδρίας χρήστη (session), είναι αρκετά πιο δύσκολα. Ακόμα και με τις μεθόδους που έχουν εφαρμοσθεί μετά από έρευνες τα τελευταία χρόνια, ο αναλυτής των αρχείων καταγραφής δεν μπορεί να είναι απόλυτα σίγουρος ότι έχει χωρίσει σωστά τους μοναδικούς επισκέπτες της ιστοσελίδας και τα session τους. Συγκεκριμένα, η αναγνώριση χρήστη θεωρείται το πιο δύσκολο κομμάτι της Εξόρυξης Χρήσεων του Διαδικτύου λόγω του "θορύβου" που περιέχουν τα δεδομένα εισόδου αλλά και της σημερινής μορφής του Διαδικτύου και των αρχείων καταγραφής [13], [14]. Θα μπορούσαμε να είμαστε σίγουροι ότι βρήκαμε τα αιτήματα ενός μοναδικού χρήστη μόνο αν υπήρχε κάποιος συνδυασμός παρακολούθησης από τη μεριά του επισκέπτη. Στη δική μας περίπτωση έχουμε μόνο την IP διεύθυνσή του στην περίπτωση του CLF που μπορεί να συμπληρωθεί από το πεδίο του User Agent στην περίπτωση του ECLF. Μερικά από τα προβλήματα που μπορεί να συναντήσουμε είναι[1]:

- Συνήθως οι πάροχοι υπηρεσιών Διαδικτύου (ISPs) έχουν ένα απόθεμα από proxy εξυπηρετητές με τους οποίους οι χρήστες μπορούν να επισκεφτούν τον Παγκόσμιο Ιστό. Έτσι ένας proxy εξυπηρετητής μπορεί να "κρύβει" από πίσω του πολλούς χρήστες που επισκέφτηκαν μια ιστοσελίδα, ακόμα και κατά το ίδιο χρονικό διάστημα.

- Μερικοί ISPs ή κάποια εργαλεία αναθέτουν στον ίδιο επισκέπτη πολλές IP διευθύνσεις κατά τη διάρκεια μιας επίσκεψης και έτσι αιτήματα που έχουν γίνει από τον ίδιο χρήστη φαίνεται ότι έχουν διαφορετική IP διεύθυνση.

- Κάποιος χρήστης μπορεί να επισκεφθεί μια σελίδα από διαφορετικές συσκευές. Ιδιαίτερα στη σημερινή εποχή με την έκρηξη των φορητών συσκευών (tablets, smartphones, smartwatches) κάποιος επισκέπτης θα έχει διαφορετική διεύθυνση IP, περιηγητή ιστού και λειτουργικό σύστημα ανάμεσα σε διαδοχικές επισκέψεις του.

- Τέλος, ακόμα και ο ίδιος χρήστης στην ίδια συσκευή μπορεί να χρησιμοποιεί διαφορετικά λειτουργικά συστήματα και διαφορετικά προγράμματα περιήγησης οπότε το πεδίο User Agent μπορεί να διαφέρει από αίτημα σε αίτημα και ακόμα και με την ίδια IP διεύθυνση δεν είμαστε σίγουροι ότι πρόκειται για ένα μοναδικό επισκέπτη.

Υπάρχουν διάφοροι τρόποι[15] με τους οποίους μπορεί να γίνει η αναγνώριση ενός χρήστη οι οποίοι θα περιγραφούν στις επόμενες ενότητες. Αυτοί οι τρόποι μπορούν να χωριστούν σε δύο μεγάλες κατηγορίες, τις "προληπτικές", όπως είναι τα cookies ή η εγγραφή στην ιστοσελίδα, και αυτές που δρουν μόνο μέσω των αρχείων καταγραφής. Στη δεύτερη κατηγορία, οι χρήστες διαχωρίζονται μέσω των μοτίβων περιήγησής τους ή με άλλες ευριστικές μεθόδους (όπως η συσταδοποίηση) σχετικά με την "ηλεκτρονική συμπεριφορά" τους [16], [17]. Πέρα από μεμονωμένες τεχνικές, έχουν εφαρμοσθεί και συνδυασμοί αυτών με σκοπό την καλύτερη αναγνώριση μεμονωμένων χρηστών[18].

2.5.1 Αναγνώριση χρήστη μέσω διεύθυνσης IP

Η διεύθυνση IP χρησιμοποιείται προκειμένου να ανατεθεί μια μοναδική διεύθυνση σε μία συσκευή η οποία βρίσκεται εντός του Διαδικτύου. Όπως έχει αναφερθεί, είναι το πρώτο πεδίο στο CLF του αρχείου καταγραφής. Μερικές φορές[16] χρησιμοποιείται μόνο η IP διεύθυνση προκειμένου να αναγνωριστεί ένας χρήστης, μια προσέγγιση η οποία μπορεί να έχει πολύ καλά αποτελέσματα για μικρές χρονικές περιόδους ή όταν δεν ενδιαφερόμαστε τόσο πολύ για ακριβή αναγνώριση των χρηστών μέσα στο αρχείο καταγραφής. Όμως με τη δομή του Διαδικτύου σήμερα, πολλοί χρήστες μπορεί να "κρύβονται" πίσω από έναν proxy εξυπηρετητή όπως αναφέρθηκε παραπάνω και έτσι να φαίνεται μόνο μια IP διεύθυνση στο αρχείο καταγραφής, παρόλο που υπήρχαν πολλοί χρήστες οι οποίοι προσπέλασαν την ιστοσελίδα. Επίσης το caching, είναι ένα ακόμα πρόβλημα στην αναγνώριση χρηστών. Αν κάποιος χρήστης έχει προσπελάσει στο παρελθόν μια ιστοσελίδα τότε το πρόγραμμα περιήγησης που χρησιμοποιεί θα λάβει τα αιτήματα από την τοπική μνήμη της συσκευής και όχι από τον εξυπηρετητή, με αποτέλεσμα στο αρχείο καταγραφής να μην φαίνονται όλα τα αιτήματα του χρήστη.

Για την λύση των παραπάνω προβλημάτων έχουν προταθεί αρκετές μέθοδοι, όπως για παράδειγμα [19] να απορρίπτονται τα αιτήματα του αρχείου καταγραφής τα οποία προέρχονται από

proxy εξυπηρετητές και τα οποία μπορούν να αναγνωριστούν επειδή το πεδίο διεύθυνσης επισκέπτη περιέχει τις λέξεις “proxy” ή “cache”. Βέβαια μια τέτοια αντιμετώπιση δεν είναι η πλέον κατάλληλη διότι έτσι μπορεί να αφαιρεθεί ένα σημαντικό κομμάτι του αρχείου καταγραφής με αποτέλεσμα τα μοτίβα που θα ανακαλυφθούν στη συνέχεια να είναι ελλιπή, λανθασμένα ή να έχουν εξαχθεί από πολύ λίγη πληροφορία.

2.5.2 Αναγνώριση χρήστη μέσω στοιχείων ταυτότητας

Τα δεδομένα εγγραφής μιας ιστοσελίδας μπορούν να χρησιμοποιηθούν για την αναγνώριση ενός χρήστη. Αυτά τα δεδομένα μπορούν να φανούν στα πεδία ταυτοποίησης ή authuser, σε περίπτωση που ο χρήστης έχει κάνει κάποια σύνδεση ή ταυτοποίηση πριν το αίτημα μιας συγκεκριμένης σελίδας. Όμως, στη σημερινή μορφή του Παγκόσμιου Ιστού που η ανωνυμία είναι μία από τις πρώτες προτεραιότητες των χρηστών, τέτοιες ιστοσελίδες δεν προσπελάζονται συχνά [20] επομένως αυτή η μέθοδος δεν είναι τόσο διαδεδομένη.

2.5.3 Αναγνώριση χρήστη μέσω cookies

Τα cookies αποτελούν μία συμβολοσειρά κειμένου, που θέτονται από έναν εξυπηρετητή του Παγκοσμίου Ιστού και περιέχουν ό,τι πληροφορία έχει θέσει ο διαχειριστής του εξυπηρετητή. Οι πιο συνήθεις λειτουργίες των cookies είναι να συνδέουν τις προσβάσεις ενός χρήστη μέσα στην ιστοσελίδα, να θυμούνται τις ενεργές συνδέσεις ενός εγγεγραμμένου χρήστη στην ιστοσελίδα έτσι ώστε να μην χρειάζεται να συνδέεται ξανά σε κάθε επίσκεψή του, να προτείνουν προϊόντα ή άλλες σελίδες αναλόγως με τις προτιμήσεις του χρήστη και τέλος να διατηρούν κάποια στοιχεία από προηγούμενες προσβάσεις, όπως είναι για παράδειγμα το καλάθι αγορών σε ηλεκτρονικά καταστήματα.

Τα cookies είναι επίσης ένας πολύ αξιόπιστος τρόπος με τον οποίο μπορεί να αναγνωριστεί ένας μοναδικός χρήστης[21]. Όμως, και πάλι λόγω ανωνυμίας υπάρχουν αρκετοί χρήστες ή προγράμματα περιήγησης τα οποία απενεργοποιούν αυτή τη δυνατότητα. Επίσης τα cookies μπορούν να σβηστούν από το ιστορικό του προγράμματος περιήγησης του επισκέπτη ή από τον εξυπηρετητή καθώς μετά από ένα συγκεκριμένο χρονικό διάστημα “λήγουν”.

2.5.4 Αναγνώριση χρήστη μέσω πληροφοριών πελάτη

Οι πιο συχνές κατευθύνσεις στην αναγνώριση χρήστη τα τελευταία χρόνια είναι χρησιμοποιώντας το πεδίο του User Agent, το οποίο περιέχει το λειτουργικό σύστημα του επισκέπτη και την έκδοση αλλά και το όνομα του προγράμματος περιήγησης. Σε περίπτωση που δύο αιτήματα έχουν ακριβώς τις ίδιες IP διευθύνσεις αλλά διαφέρουν στο λειτουργικό σύστημα ή/και στο πρόγραμμα περιήγησης, τότε κατά πάσα πιθανότητα έχουμε να κάνουμε με δύο διαφορετικούς χρήστες[13]. Παρόλο που αυτή η τεχνική δεν φαίνεται τόσο αξιόπιστη με μία πρώτη ματιά, γιατί όπως έχουμε προαναφέρει πολλοί χρήστες έχουν παραπάνω από ένα λειτουργικό σύστημα ή προγράμματα περιήγησης για χρονικές περιόδους μικρότερες της μίας μέρας, μπορούμε να κάνουμε την υπόθεση εργασίας ότι ένας επισκέπτης είναι δύσκολο να έχει αλλάξει και τα δύο αυτά χαρακτηριστικά.

2.5.5 Αναγνώριση χρήστη μέσω τοπολογίας ιστοσελίδας

Μια άλλη ευριστική μέθοδος, όπως αυτή που χρησιμοποιεί το πεδίο User Agent, είναι να χρησιμοποιήσουμε την τοπολογία της ιστοσελίδας προκειμένου να εντοπίσουμε τις συνεδρίες χρηστών [13]. Πιο συγκεκριμένα, αν μία σελίδα που ζητήθηκε από έναν χρήστη δεν είναι προσβάσιμη με βάση το πεδίο αναφοράς από τις προηγούμενες σελίδες τις οποίες έχει ζητήσει, τότε μπορούμε να πούμε ότι έχουμε έναν καινούριο χρήστη. Βέβαια και αυτή η τεχνική μπορεί να μην είναι αποδοτική σε περίπτωση που ένας επισκέπτης μπαίνει σε μία ιστοσελίδα από κάποιο σελιδοδείκτη ή κάποια καρφιτσωμένη καρτέλα ή από κάποια μηχανή αναζήτησης.

Κεφάλαιο 3

Αναγνώριση συνεδρίας χρήστη

3.1 Εισαγωγή

Ως συνεδρία χρήστη (user session) ορίζεται το σύνολο των ιστοσελίδων οι οποίες προσπελάστηκαν κατά τη διάρκεια μιας επίσκεψης. Η αναγνώριση των sessions ενός χρήστη, που έπεται της αναγνώρισης ενός χρήστη από τα αρχεία καταγραφής είναι το τελευταίο κομμάτι της προεπεξεργασίας των δεδομένων και παράλληλα το πιο σημαντικό από όλα τα κομμάτια, γιατί τα αιτήματα που ανήκουν σε ένα session αποτελούν τα δεδομένα εισόδου για τους αλγορίθμους Εξόρυξης Δεδομένων. Μέχρι σήμερα έχουν αναπτυχθεί αρκετές μέθοδοι προκειμένου να αναγνωριστεί ένα session, μερικοί από τους οποίους είναι τα cookies, ένα ειδικό αναγνωριστικό στο URI, αλλά και plugins της Javascript (π.χ. Google Analytics). Όμως, όλες αυτές οι τεχνικές μπορούν να "κατασταλθούν" με διάφορα πρόσθετα προγράμματα στους περιηγητές ιστού τα οποία απενεργοποιούν τη Javascript αλλά και τα cookies. Σε αντίθεση με τις παραπάνω μεθόδους, τα αρχεία καταγραφής εξυπηρετητών του Παγκόσμιου Ιστού δεν μπορούν να τροποποιηθούν από έναν επισκέπτη μιας ιστοσελίδας και τα session των χρηστών μπορούν να αναγνωριστούν είτε με αλγορίθμους που χρησιμοποιούν χρονικές ευριστικές μεθόδους, είτε με αλγορίθμους που χρησιμοποιούν ευριστικές πλοήγησης, είτε με συνδυασμό των δύο παραπάνω μεθόδων. Σε αυτό το κεφάλαιο θα παρουσιάσουμε μια νέα μέθοδο αναγνώρισης των session η οποία βασίζεται στη ασαφή συσταδοποίηση. Αρχικά θα κάνουμε μια επισκόπηση της σχετικής βιβλιογραφίας και στη συνέχεια θα παρουσιάσουμε τις προαπαιτούμενες γνώσεις για την καλύτερη κατανόηση του προτεινόμενου αλγορίθμου. Οι προαπαιτούμενες γνώσεις περιλαμβάνουν την ασαφή συσταδοποίηση, την αφαιρετική συσταδοποίηση, η οποία προήλθε από τη συσταδοποίηση με τη μέθοδο συνάρτησης πυκνότητας, τις μετρικές απόστασης και τους δείκτες εγκυρότητας (validity indices). Τέλος, παρουσιάζεται αναλυτικά ο προτεινόμενος αλγόριθμος καθώς και τα αποτελέσματά του σε πραγματικά δεδομένα αρχείων εξυπηρετητών και συγκεκριμένα αυτά της διαδικτυακής κοινότητας φοιτητών της σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών[22].

3.2 Σχετική Βιβλιογραφία

Η ανάκτηση των συνεδριών χρήστη από τα αρχεία καταγραφής των εξυπηρετητών μπορεί να γίνει με διάφορους τρόπους. Κατ' αρχάς θα μπορούσαμε να αναγνωρίσουμε ένα session με βάση ένα αναγνωριστικό στο URI ή μέσω των cookies. Σε περίπτωση που αυτό δεν είναι δυνατό θα πρέπει να εφαρμόσουμε ευριστικές τεχνικές. Η πιο προφανής τεχνική είναι να εφαρμόσουμε ένα όριο στο χρόνο μετά από το τελευταίο αίτημα που έχει κάνει ένας χρήστης. Η τιμή αυτού του ορίου

παίρνει διάφορες τιμές, που προέρχονται από εμπειρικά δεδομένα. Για παράδειγμα οι Catledge και Pitkow[23] κατέληξαν ότι η εξίσωση αυτού του ορίου με τα 25.5 λεπτά έχει τα πιο αντιπροσωπευτικά αποτελέσματα. Άλλες τιμές για αυτό το όριο κυμαίνονται από 10 λεπτά έως 24 ώρες [24], αλλά ένα προκαθορισμένο όριο που έχει προταθεί από τον Cooley [25], και χρησιμοποιείται συχνά στη βιβλιογραφία είναι τα 30 λεπτά[26]. Όμως αυτή η τεχνική δεν λαμβάνει υπόψιν ότι διαφορετικές σελίδες

- έχουν περιεχόμενο που διαφοροποιείται σε ποσότητα
- δεν έχουν το ίδιο ενδιαφέρον για κάθε επισκέπτη τους

Επιπρόσθετα, άλλες παράμετροι οι οποίες μπορεί να επηρεάσουν το χρόνο που ένας χρήστης περνάει σε μία ιστοσελίδα είναι η ταχύτητα με την οποία διαβάζει, το ότι μπορεί να κάνει και άλλα πράγματα παράλληλα καθώς και η τοπολογία της σελίδας. Προκειμένου να ανταποκριθούν σε αυτά τα χαρακτηριστικά έχουν προταθεί δυναμικά όρια, τα οποία βασίζονται σε παραδοσιακές μεθόδους αναγνώρισης ενός session [27], [28], [29] . Οι He Xinhua και Wang Quiongan[27] προτείνουν τον ακόλουθο αλγόριθμο: Καταρχάς, τίθεται ένα αρχικό όριο με βάση μια γραμμική εξίσωση, η οποία αποτελείται από δύο κομμάτια. Το πρώτο κομμάτι είναι ένας συντελεστής εξομάλυνσης και το δεύτερο αρχικοποιείται ανάλογα με τον αριθμό των σελίδων με τις οποίες συνδέεται η συγκεκριμένη σελίδα. Μετά από αυτό το βήμα το όριο προσαρμόζεται με βάση το πόσο διήρκεσαν τα προηγούμενα sessions και το χρόνο που διήρκεσε το τελευταίο session.

Εκτός από τις μεθόδους που βασίζονται σε ευριστικές χρόνου, μία δημοφιλής προσέγγιση στην αναγνώριση ενός session είναι οι αλγόριθμοι που βασίζονται σε ευριστικές πλοήγησης [30]. Αυτές οι μέθοδοι κατασκευάζουν το γράφο μεταξύ των σελίδων ενός ιστοτόπου. Όταν μία σελίδα έχει ένα σύνδεσμο που οδηγεί σε μία άλλη σελίδα, τότε αυτές οι δύο συνδέονται με μία ακμή. Τα sessions χωρίζονται αναλόγως με τον αν ένας χρήστης έκανε δύο διαδοχικά αιτήματα, τα οποία συνδέονται με κάποια ακμή στο γράφο του ιστοτόπου. Αν δεν υπάρχει κάποια σύνδεση μεταξύ τους, τότε έχουμε ένα νέο session.

Μία ακόμα ευριστική που συνδυάζεται με τις προηγούμενες μεθόδους είναι αυτή του πεδίου referrer όταν αυτό είναι διαθέσιμο[26]. Επίσης, οι ευριστικές μέθοδοι με βάση το χρόνο έχουν συνδυαστεί με αυτές που βασίζονται σε ευριστικές πλοήγησης για την καλύτερη εύρεση των session [31], [32].

Πρόσφατα, η ασαφής συσταδοποίηση χρησιμοποιήθηκε από τους Ansari et al προκειμένου να ανακαλυφθούν τα sessions των χρηστών[33]. Πιο συγκεκριμένα, προτείνεται ένας αλγόριθμος με τον οποίο τα sessions χωρίζονται με ευριστικές μεθόδους βασισμένες στο χρόνο, και στη συνέχεια ανατίθενται βάρη με βάση ασαφείς συναρτήσεις συμμετοχής ανάλογα με:

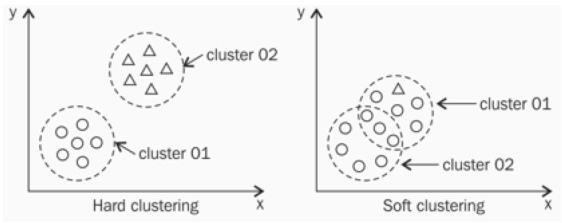
- τη συχνότητα που μία συγκεκριμένη σελίδα ζητήθηκε κατά τη διάρκεια ενός session
- τον αριθμό των bytes που μεταφέρθηκαν ανά σελίδα κατά τη διάρκεια ενός session
- τον χρόνο που πέρασε κάποιος επισκέπτης σε μία σελίδα κατά τη διάρκεια ενός session

Μετά από αυτό το στάδιο, τα προεπεξεργασμένα δεδομένα συσταδοποιούνται χρησιμοποιώντας τους αλγορίθμους της ασαφούς συσταδοποίησης c-κέντρων[34] και fuzzy c-medoids, οι οποίοι αρχικοποιούνται τυχαία ή με τη χρήση μιας συνάρτησης πυκνότητας. Η συσταδοποίηση με τη μέθοδο συνάρτησης πυκνότητας χρησιμοποιείται προκειμένου να γίνει μια εκτίμηση του αριθμού των

συστάδων. Στη συνέχεια η ποιότητα της συσταδοποίησης καθορίζεται από τον υπολογισμό των δεικτών εγκυρότητας. Η μέθοδος που προτείνουμε εμείς, χρησιμοποιεί μια διαφορετική αναπαράσταση των δεδομένων, η οποία δεν χρησιμοποιεί καθόλου τη συχνότητα προσπέλασης μιας σελίδας. Επιπρόσθετα, η αναπαράστασή μας προσπαθεί να απλοποιήσει τα αιτήματα που έχουν γίνει σε χρονικά δεδομένα, προκειμένου να τρέξουν οι αλγόριθμοι συσταδοποίησης. Παρόλο που χρησιμοποιούμε ασαφή συσταδοποίηση c-κέντρων, που αρχικοποιείται με αφαιρετική συσταδοποίηση, πειραματιζόμαστε με διαφορετικές μετρικές απόστασης μεταξύ των κέντρων των συστάδων και των δεδομένων που επηρεάζουν τα αποτελέσματα της συσταδοποίησης όπως και τη σύγκλιση των αλγορίθμων. Επίσης, κάνουμε δοκιμές σε σχέση με την τιμή του δείκτη ασάφειας (fuzzifier). Τα αποτελέσματα της συσταδοποίησης αξιολογούνται με βάση ορισμένους δείκτες εγκυρότητας και με επιθεώρηση των αρχείων καταγραφής παρατηρούμε ότι ο αλγόριθμός μας παράγει ικανοποιητικά αποτελέσματα.

3.3 Στοιχεία της προτεινόμενης μεθόδου

3.3.1 Ασαφής συσταδοποίηση



Σχήμα 3.1: Σαφής και ασαφής συσταδοποίηση[2]

Η συσταδοποίηση δεδομένων είναι η διαδικασία με την οποία τα δεδομένα χωρίζονται σε παρόμοιες κλάσεις ή συστάδες, με σκοπό τα αντικείμενα που ανήκουν στις ίδιες συστάδες να είναι όσο πιο όμοια γίνεται και τα αντικείμενα που ανήκουν σε διαφορετικές συστάδες να είναι όσο πιο ανόμοια γίνεται. Στη σαφή συσταδοποίηση ή αλλιώς hard clustering, κάθε σημείο των δεδομένων ανήκει σε ακριβώς μία συστάδα, σε αντίθεση με την ασαφή συσταδοποίηση, που τα δεδομένα μπορεί να ανήκουν σε δύο ή περισσότερες συστάδες. Ένας από τους πιο διαδεδομένους αλγόριθμους στην ασαφή συσταδοποίηση είναι αυτός της ασαφούς συσταδοποίησης c-κέντρων. Αναπτύχθηκε από τον Dunn το 1973[35] και βελτιώθηκε από τον Bezdek το 1981[34]. Στην επόμενη υποενότητα περιγράφεται αναλυτικότερα ο αλγόριθμος της ασαφούς συσταδοποίησης [36].

3.3.2 Πίνακας ασαφούς διαμέρισης

Στην ασαφή συσταδοποίηση, κάθε σημείο των δεδομένων μπορεί να είναι μέλος μιας ασαφούς συστάδας/κλάσης. Για αυτό το σκοπό, ορίζουμε ως τιμή συμμετοχής του k-στού σημείου δεδομένων, x_k, στην i-στή κλάση:

$$\mu_{ik} = \mu_{A_i}(x_k) \in [0, 1] \tag{3.1}$$

με τον περιορισμό το άθροισμα όλων των τιμών συμμετοχής για ένα σημείο των δεδομένων για όλες τις κλάσεις να είναι ίσο με ένα:

$$\sum_{i=1}^{c} \mu_{ik} = 1, \text{για κάθε } k = 1, 2, \ldots, n \tag{3.2}$$

Επίσης, δεν μπορούν να υπάρχουν κενές κλάσεις και δεν μπορεί να υπάρχει κλάση, η οποία να περιέχει όλα τα δεδομένα, περιορισμοί που κωδικοποιούνται ως εξής:

$$0 < \sum_{i=1}^{c} \mu_{ik} = 1 < n \tag{3.3}$$

Για την ασαφή συσταδοποίηση ισχύουν και οι ακόλουθες εξισώσεις:

$$\bigcup_{i=1}^{c} \mu_{A_i}(x_k) = 1 \tag{3.4}$$

$$0 < \sum_{i=1}^{c} \mu_{A_i}(x_k) < n \tag{3.5}$$

Τώρα, μπορούμε να ορίσουμε ένα σύνολο από πίνακες ασαφούς διαμέρισης (fuzzy partition matrix) για μία συσταδοποίηση που περιλαμβάνει c κλάσεις και n σημεία δεδομένων ως

$$M_{fc} = \left\{ \underset{\sim}{U} | \mu_{ik} \in [0, 1]; \sum_{i=1}^{c} \mu_{ik} = 1; 0 < \sum_{k=1}^{n} \mu_{ik} < n \right\} \tag{3.6}$$

όπου $i = 1, 2, \ldots, c$ και $k = 1, 2, \ldots, n$. Κάθε $\underset{\sim}{U} \in M_{fc}$ είναι ένας πίνακας ασαφούς διαμέρισης c-συστάδων.

3.3.3 Ασαφής Συσταδοποίηση C-Κέντρων

Αρχικά ορίζεται η αντικειμενική συνάρτηση (objective function) J_m για μία ασαφή διαμέριση c-κλάσεων:

$$J_m(\underset{\sim}{U}, v) = \sum_{k=1}^{n} \sum_{i=1}^{c} (\mu_{ik})^{m'} (d_{ik})^2 \tag{3.7}$$

όπου $d_{ik} = d(x_k - v_i)$ είναι η απόσταση μεταξύ του k-στού σημείου δεδομένων και της i-στής συστάδας και μ_{ik} είναι η συμμετοχή του k-στού σημείου δεδομένων στη i-στή κλάση. Η καλύτερη συσταδοποίηση επιτυγχάνεται εκεί που η τιμή της συνάρτησης J_m είναι μικρότερη. Η παράμετρος m' ονομάζεται δείκτης ασάφειας (fuzzifier)[34] και η τιμή της είναι στο διάστημα $[1, \infty]$. Αυτή η παράμετρος ελέγχει τον βαθμό της ασάφειας στη διαδικασία της συσταδοποίησης. Επιπλέον, v_i είναι το κέντρο της i-στης συστάδας, που περιγράφεται από m συντεταγμένες και μπορεί να γραφεί σε μορφή διανύσματος ως $\mathbf{v_i} = [v_{i1}, v_{i2}, \ldots, v_{im}]$. Κάθε μία από τις συντεταγμένες του

κέντρου κάθε κλάσης υπολογίζεται από την ακόλουθη εξίσωση:

$$v_{ij} = \frac{\sum_{k=1}^{n} \mu_{ik}^{m'} \cdot x_{ki}}{\sum_{k=1}^{n} \mu_{ik}^{m'}} \tag{3.8}$$

Η καλύτερη ασαφής c-διαμέριση περιγράφεται από την εξίσωση:

$$J_m^*(\underset{\sim}{U}^*, v^*) = \min_{M_{fc}} J(\underset{\sim}{U}, v) \tag{3.9}$$

Η λύση στην παραπάνω εξίσωση δεν μπορεί να είναι εγγυημένα η καλύτερη καθολικά. Μία προσέγγιση που έχει εφαρμοστεί για να βρεθεί η καλύτερη δυνατή λύση μέσα σε ένα συγκεκριμένο επίπεδο ακρίβειας, είναι η μέθοδος της επαναληπτικής βελτιστοποίησης που έχει προταθεί και αυτή από τον Bezdek[34]. Τα βήματα του αλγορίθμου είναι τα εξής:

Αλγόριθμος 1 Αλγόριθμος συσταδοποίησης c-κέντρων

1. Θέσε τιμές στις παραμέτρους c ($2 \leq c \leq n$) και m'. Αρχικοποίησε τον πίνακα ασαφούς διαμέρισης $\underset{\sim}{U}$. Κάθε βήμα αυτού του αλγορίθμου θα επισημαίνεται ως r όπου $r = 0, 1, 2, \ldots$.

2. Υπολόγισε τα c κέντρα $\mathbf{v_i}^{(r)}$ για κάθε βήμα.

3. Ενημέρωσε τον πίνακα ασαφούς διαμέρισης για το r-οστό βήμα $\underset{\sim}{U}^{(r)}$, ως εξής:

$$\mu_{ik}^{(r+1)} = \left[\sum_{j=1}^{c} \left(\frac{d_{ik}^{(r)}}{d_{jk}^{(r)}} \right)^{\frac{2}{m'-1}} \right]^{-1} \text{για } I_k = \emptyset \tag{3.10}$$

   ή

$$\mu_{ik}^{(r+1)} = 0, \text{για όλες τις συστάδες } i \text{ όπου } i \in \underset{\sim}{I}_k, \tag{3.11}$$

   όπου

$$I_k = \left\{ i \mid 2 \leq c < n; d_{ik}^{(r)} = 0 \right\} \tag{3.12}$$

   και

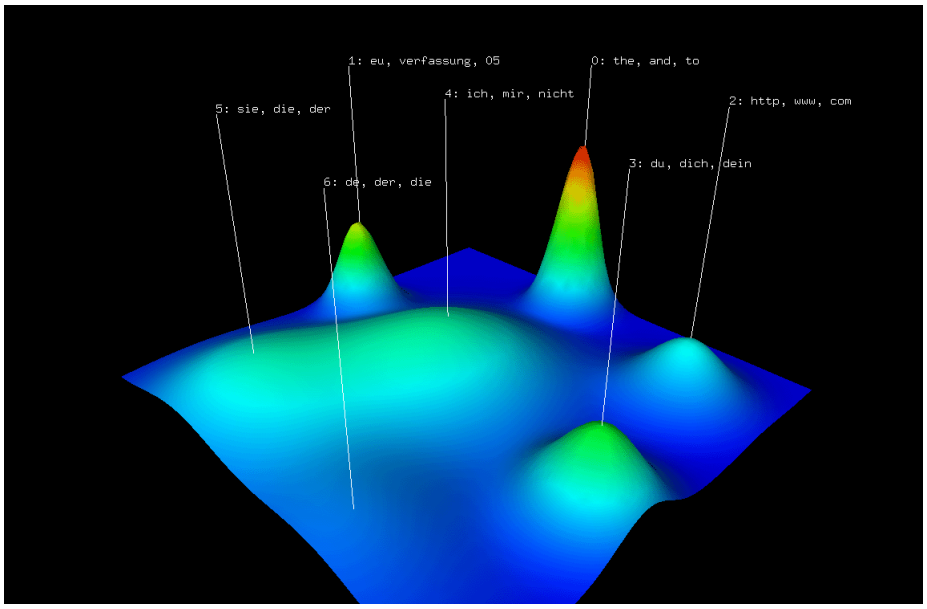
$$\underset{\sim}{I}_k = \{1, 2, \ldots, c\} - I_k \tag{3.13}$$

   και

$$\sum_{i \in I_k} \mu_{ik}^{(r+1)} = 1 \tag{3.14}$$

4. Αν $\left|\left| \underset{\sim}{U}^{(r+1)} - \underset{\sim}{U}^{(r)} \right|\right| \leq \epsilon_L$ σταμάτα την εκτέλεση του αλγορίθμου. Αλλιώς θέσε $r = r + 1$ και επίστρεψε στο βήμα 2. Η σταθερά ϵ_L είναι ένα προκαθορισμένο επίπεδο ακρίβειας για να καθοριστεί αν η λύση του αλγορίθμου είναι αρκετά καλή.

3.3.4 Συσταδοποίηση με τη μέθοδο συνάρτησης πυκνότητας



Σχήμα 3.2: Συσταδοποίηση με τη μέθοδο συνάρτησης πυκνότητας [3]

Η συσταδοποίηση με τη μέθοδο συνάρτησης πυκνότητας (Mountain Clustering) προτάθηκε από τους Yager και Filev[37], προκειμένου να γίνει μια εκτίμηση του αριθμού και των τοποθεσιών των κέντρων των συστάδων. Αρχικά, σχηματίζεται ένα πλέγμα στο χώρο των δεδομένων και τα πιθανά κέντρα των συστάδων θεωρείται ότι βρίσκονται μέσα στο πλέγμα. Στη συνέχεια, μία συνάρτηση πυκνότητας υπολογίζεται για κάθε σημείο στο πλέγμα. Ο τύπος αυτής της συνάρτησης είναι:

$$m(\mathbf{v}) = \sum_{i=1}^{N} \exp\left(- \frac{\left|\left|\mathbf{v} - \mathbf{x}_i\right|\right|^2}{2\sigma^2} \right) \tag{3.15}$$

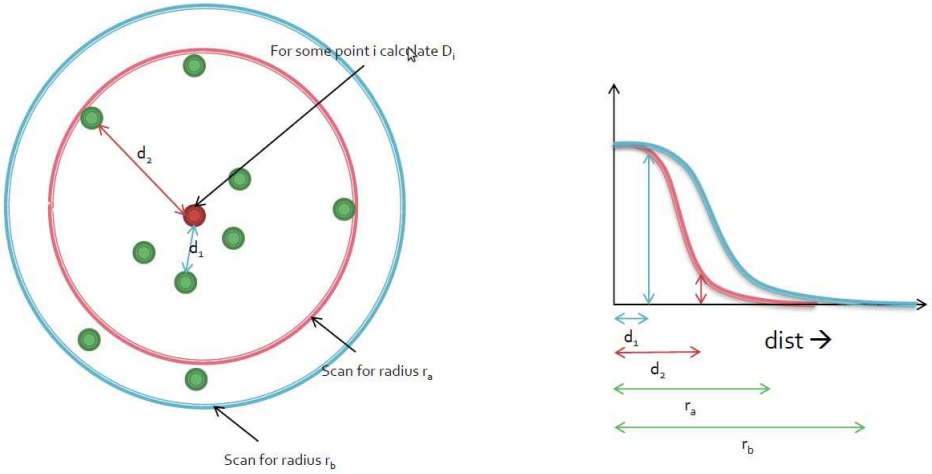
όπου $\mathbf{x}_i$ είναι το i-στό σημείο των δεδομένων και σ μια ορισμένη σταθερά. Η ερμηνεία αυτής της εξίσωσης είναι ότι η πυκνότητα ενός σημείου $\mathbf{v}$ επηρεάζεται από το πόσα άλλα σημεία των δεδομένων βρίσκονται κοντά του. Όσο περισσότερα είναι αυτά, τόσο μεγαλύτερη τιμή θα έχει αυτή η συνάρτηση. Η σταθερά σ καθορίζει το ύψος και την ομαλότητα της συνάρτησης πυκνότητας. Το τρίτο βήμα του αλγορίθμου είναι η επιλογή των σημείων που θα είναι τα κέντρα των συστάδων αναλόγως με την τιμή της συνάρτησης. Το πρώτο κέντρο επιλέγεται να είναι το σημείο για το οποίο η συνάρτηση έχει τη μεγαλύτερη τιμή. Μετέπειτα, πρέπει να εξαλείψουμε το αποτέλεσμα της επιλογής του πρώτου κέντρου, αναθεωρώντας τη συνάρτηση πυκνότητας. Η ανανεωμένη συνάρτηση πυκνότητας υπολογίζεται από τον ακόλουθο τύπο:

$$m_{new}(\mathbf{v}) = m(\mathbf{v}) - m(\mathbf{c}_1) \exp\left(- \frac{\left|\left|\mathbf{v} - \mathbf{c}_1\right|\right|}{2\beta^2} \right) \tag{3.16}$$

Η ποσότητα που αφαιρείται εξαλείφει το αποτέλεσμα της επιλογής του πρώτου κέντρου. Άλλωστε η ανανεωμένη συνάρτηση έχει τιμή μηδέν για το σημείο που επιλέχθηκε ως κέντρο. Μετά από αυτό το βήμα, επιλέγεται ως επόμενο κέντρο αυτό που έχει τη μεγαλύτερη τιμή της ανανεωμένης συνάρτησης πυκνότητας και αυτή η διαδικασία συνεχίζεται μέχρι να φτάσουμε τον επιθυμητό

αριθμό συστάδων.

3.3.5 Αφαιρετική συσταδοποίηση



Σχήμα 3.3: Αφαιρετική συσταδοποίηση [4]

Το πρόβλημα με την μέθοδο του Mountain Clustering είναι ότι είναι υπολογιστικά ακριβή, επειδή ο αριθμός των υπολογισμών αυξάνεται εκθετικά με τη διάσταση του προβλήματος. Λόγω αυτού του προβλήματος, ο Chiu[38] πρότεινε μία αναθεώρηση της μεθόδου, που είναι γνωστή ως αφαιρετική συσταδοποίηση (subtractive clustering). Το προαναφερθέν υπολογιστικό πρόβλημα λύνεται χρησιμοποιώντας μόνο τα σημεία των δεδομένων ως υποψήφια για τα κέντρα των συστάδων, αντί να εξεταστούν όλα τα σημεία του πλέγματος. Με αυτή την τροποποίηση, οι υπολογισμοί τώρα είναι ανάλογοι με το μέγεθος του προβλήματος. Παρόλο που τα κέντρα των συστάδων δεν βρίσκονται πάντα εκεί που είναι τα σημεία των δεδομένων, η προσέγγιση μπορεί να θεωρηθεί αρκετά καλή, λαμβάνοντας υπόψιν τους μειωμένους υπολογισμούς. Κατ' αρχάς, κάθε σημείο των δεδομένων θεωρείται ένα πιθανό κέντρο των συστάδων και υπολογίζεται η ακόλουθη συνάρτηση πυκνότητας για όλα τα σημεία:

$$D_i = \sum_{j=1}^{n} \exp\left(- \frac{\left|\left| \mathbf{x}_i - \mathbf{x}_j \right|\right|^2}{(\frac{r_a}{2})^2} \right) \tag{3.17}$$

όπου r_a είναι μια ακτίνα γειτονιάς. Επομένως, υψηλή πυκνότητα σημαίνει ότι ένα σημείο έχει πολλούς γείτονες. Το πρώτο κέντρο των συστάδων x_{c_i} είναι αυτό που έχει τη μεγαλύτερη τιμή της συνάρτησης πυκνότητας. Ακολούθως, η τιμή της συνάρτησης πυκνότητας για τα άλλα σημεία $\mathbf{x}_i$ αναθεωρείται από τον ακόλουθο τύπο:

$$D_i = D_i - D_{c_i} \exp\left(- \frac{\left|\left| \mathbf{x}_i - \mathbf{x}_{c_i} \right|\right|^2}{(\frac{r_b}{2})^2} \right) \tag{3.18}$$

όπου r_b είναι μια θετική σταθερά, που ορίζει μια γειτονιά στην οποία θα μειωθεί η συνάρτηση πυκνότητας. Σαν αποτέλεσμα, κάθε κέντρο μιας συστάδας θα έχει σημαντικά μειωμένη τιμή της συνάρτησης πυκνότητας. Μετά από αυτό το βήμα, επιλέγεται ως επόμενο κέντρο αυτό που έχει

τη μεγαλύτερη τιμή της ανανεωμένης συνάρτησης πυκνότητας και αυτή η διαδικασία συνεχίζεται μέχρι να φτάσουμε τον επιθυμητό αριθμό συστάδων.

3.3.6 Μετρικές απόστασης

Ως μετρική απόστασης ορίζουμε μία συνάρτηση η οποία μας δίνει την απόσταση δύο σημείων. Παρακάτω, θα δώσουμε τους τύπους των μετρικών που χρησιμοποιήσαμε σαν παράμετρο στον αλγόριθμο ασαφούς συσταδοποίησης c-κέντρων.

Ευκλείδεια απόσταση

Η Ευκλείδεια απόσταση είναι η πιο ευρέως χρησιμοποιούμενη απόσταση και υπολογίζει την απόσταση δύο σημείων στον Ευκλείδειο χώρο. Για δύο σημεία p, q με n διαστάσεις ο τύπος της είναι:

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \tag{3.19}$$

Απόσταση Manhattan

Η απόσταση Manhattan, ορίζεται στα πλαίσια της γεωμετρίας taxicab, και υπολογίζεται ως το άθροισμα της απόλυτης τιμής της διαφοράς των καρτεσιανών συντεταγμένων. Για δύο σημεία p, q με n διαστάσεις ο τύπος της είναι:

$$d(p,q) = \sum_{i=1}^{n}|q_i - p_i| \tag{3.20}$$

Απόσταση Chebyshev

Η απόσταση Chebyshev δύο διανυσμάτων ορίζεται ως η μεγαλύτερη διαφορά των συντεταγμένων τους σε οποιαδήποτε διάσταση. Για δύο σημεία p, q με n διαστάσεις ο τύπος της είναι:

$$d(p,q) = \max_i(p_i - q_i) \tag{3.21}$$

Απόσταση Minkowski

Η απόσταση Minkowski θεωρείται μια γενίκευση της Ευκλείδειας απόστασης και της απόστασης Manhattan. Για δύο σημεία x, y με n διαστάσεις ο τύπος της είναι:

$$d(x,y) = \left(\sum_{i=1}^{n}|x_i - y_i|^p\right)^{\frac{1}{p}} \tag{3.22}$$

3.3.7 Δείκτες εγκυρότητας

Ένα από τα προβλήματα στον αλγόριθμο της ασαφούς συσταδοποίησης c-κέντρων, είναι ότι ο αριθμός των κέντρων (c) πρέπει να προσδιοριστεί εξ' αρχής. Όμως, διαφορετικός αριθμός κέντρων έχει ως αποτελέσματα διαφορετικές συσταδοποιήσεις. Επομένως, η ποιοτική ανάλυση των συστάδων που έχουν βρεθεί, κρίνεται απαραίτητη. Υπάρχουν πολλοί δείκτες εγκυρότητας στη βιβλιογραφία, που μπορούν να χωριστούν σε δύο κατηγορίες. Στην πρώτη χρησιμοποιούνται μόνο

τις τιμές συμμετοχής από τον πίνακα της ασαφούς διαμέρισης, ενώ στη δεύτερη χρησιμοποιείται και ο πίνακας ασαφούς διαμέρισης και τα δεδομένα. Στη συνέχεια του κεφαλαίου, θα αναλύσουμε λεπτομερώς τους δείκτες εγκυρότητας που χρησιμοποιήθηκαν στην προτεινόμενή μας μέθοδο. Η επιλογή των δεικτών αυτών έγινε σύμφωνα με την έρευνα των Wang και Zhang [39].

Κατηγορίες δεικτών εγκυρότητας	**Παραδείγματα από κάθε κατηγορία**
Δείκτες εγκυρότητας που χρησιμοποιούν μόνο τον πίνακα ασαφούς διαμέρισης	VPC, VPE, VMPC
Δείκτες εγκυρότητας που χρησιμοποιούν τον πίνακα ασαφούς διαμέρισης και τα δεδομένα	VXB, VFS, VT, VK, VPCAES

Πίνακας 3.1: Πίνακας κατηγοριοποίησης δεικτών εγκυρότητας

Validity Partition Coefficient (VPC)

Ο Bezdek [34] πρότεινε ένα δείκτη εγκυρότητας για την ασαφή συσταδοποίηση (συντελεστή κατανομής - partition coefficient) που ορίζεται ως:

$$V_{PC} = \frac{1}{n} \sum_{i=1}^{c} \sum_{i=1}^{n} u_{ij}^2 \tag{3.23}$$

Αυτός ο δείκτης υπολογίζει τον βαθμό συμμετοχής των ζευγών που μοιράζονται τα ασαφή υποσύνολα. Ο καλύτερος αριθμός συστάδων c^* βρίσκεται λύνοντας το πρόβλημα μεγιστοποίησης $\underset{2 \leq c \leq n-1}{max} V_{PC}$.

Validity Partition Entropy (VPE)

Ο Bezdek [34], [40], [41] πρότεινε τον δείκτη χωρισμού εντροπίας που υπολογίζεται ως:

$$V_{PE} = -\frac{1}{n} \sum_{i=1}^{c} \sum_{i=1}^{n} u_{ij}^2 \log_{\alpha} u_{ij} \tag{3.24}$$

Ο δείκτης αυτός υπολογίζει τον βαθμό ασάφειας σε ένα δοσμένο πίνακα ασαφούς διαμέρισης. Ο καλύτερος αριθμός συστάδων c^* βρίσκεται λύνοντας το πρόβλημα ελαχιστοποίησης $\underset{2 \leq c \leq n-1}{min} V_{PE}$.

Validity Modified Partition Coefficient (VMPC)

Ο Dave [42] πρότεινε μια βελτίωση του δείκτη εγκυρότητας PC που ονομάζεται MPC, η οποία μειώνει την μονοτονική ροπή που έχουν οι δύο προηγούμενοι δείκτες εγκυρότητας σε σχέση με τον αριθμό των συστάδων και υπολογίζεται ως:

$$V_{MPC} = 1 - \frac{c}{c-1}(1 - V_{PC}) \tag{3.25}$$

Ο καλύτερος αριθμός συστάδων c^* βρίσκεται λύνοντας το πρόβλημα μεγιστοποίησης $\underset{2 \leq c \leq n-1}{max} V_{MPC}$.

VFS

Οι Fukuyama και Sugeno, όρισαν τον ακόλουθο δείκτη εγκυρότητας [43]:

$$V_{FS} = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^{m}\left|\left|x_j - v_i\right|\right|^2 - \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^{m}\left|\left|v_i - \bar{v}\right|\right|^2 \tag{3.26}$$

όπου

$$\bar{v} = \sum_{i=1}^{c}\frac{v_i}{c} \tag{3.27}$$

Ο πρώτος όρος συνδυάζει την συμπάγεια της αναπαράστασης των δεδομένων με το βαθμό ασάφειας του πίνακα ασαφούς διαμέρισης, και ο δεύτερος όρος συνδυάζει το βαθμό ασάφειας σε κάθε γραμμή του πίνακα ασαφούς διαμέρισης με την απόσταση του i-στού κέντρου κάθε συστάδας σε σχέση με το μέσο όρο όλων των κέντρων των συστάδων. Ο καλύτερος αριθμός συστάδων c^* βρίσκεται λύνοντας το πρόβλημα ελαχιστοποίησης $\underset{2\leq c\leq n-1}{min} V_{FS}$.

Validity Xie Beni (VXB)

Οι Xiu και Beni [44] πρότειναν τον ακόλουθο δείκτη εγκυρότητας, ο οποίος αργότερα βελτιώθηκε από τον Bezdek [45]. Ο τύπος του είναι:

$$V_{XB} = \frac{\sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^{m}\left|\left|x_j - v_i\right|\right|^2}{n \cdot \min\limits_{i,j}\left|\left|v_i - v_j\right|\right|^2} \tag{3.28}$$

Ο αριθμητής υπολογίζει πόσο συμπαγής είναι η ασαφής διαμέριση και ο παρονομαστής υπολογίζει πόσο καλά διαχωρισμένα είναι τα κέντρα των συστάδων. Ο καλύτερος αριθμός συστάδων c^* βρίσκεται λύνοντας το πρόβλημα ελαχιστοποίησης $\underset{2\leq c\leq n-1}{min} V_{XB}$.

Validity Kwon (VK)

Ο δείκτης εγκυρότητας XB μειώνεται ανάλογα με τον αριθμό των συστάδων, όσο ο τελευταίος τείνει να γίνεται ίσος με τον αριθμό των σημείων δεδομένων. Ο Kwon [46] έλυσε αυτό το θέμα προτείνοντας τον ακόλουθο δείκτη εγκυρότητας:

$$V_K = \frac{\sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^{2}\left|\left|x_j - v_i\right|\right|^2 + \frac{1}{c}\sum_{i=1}^{c}\left|\left|v_i - \bar{v}\right|\right|^2}{\min\limits_{i\neq k}\left|\left|v_i - v_k\right|\right|^2} \tag{3.29}$$

Ο πρώτος όρος του αριθμητή μετράει πόσο καλά διακεκριμένες είναι οι συστάδες. Ο δεύτερος όρος του αριθμητή χρησιμοποιείται για να εξαλειφθεί η μείωση του δείκτη ανάλογα με τον αριθμό των συστάδων όσο αυτές προσεγγίζουν τον αριθμό των σημείων δεδομένων. Ο παρονομαστής υπολογίζεται πόσο καλά διακεκριμένα είναι τα κέντρα των συστάδων. Ο καλύτερος αριθμός συστάδων c^* βρίσκεται λύνοντας το πρόβλημα ελαχιστοποίησης $\underset{2\leq c\leq n-1}{min} V_K$.

Validity Tang (VT)

Ένας βελτιωμένος δείκτης εγκυρότητας με παρόμοια ιδέα με τον V_K[47] έχει τον ακόλουθο τύπο:

$$V_T(U,V;X) = \frac{\sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^2||x_j - v_i||^2 + \frac{1}{c(c-1)}\sum_{i=1}^{c}\sum_{\substack{k=1\\k\neq i}} i^c||v_i - v_k||^2}{\min\limits_{i\neq k}||v_i - v_k||^2 + \frac{1}{c}} \tag{3.30}$$

Η διαφορά με τον προηγούμενο δείκτη είναι ο δεύτερος όρος του αριθμητή που χρησιμεύει στο να εξαλειφθεί η μείωση της τιμής του δείκτη ανάλογα με τον αριθμό των συστάδων, όσο αυτές γίνονται ίσες με τον αριθμό των σημείων δεδομένων. Ο καλύτερος αριθμός συστάδων c^* βρίσκεται λύνοντας το πρόβλημα ελαχιστοποίησης $\underset{2\leq c\leq n-1}{min} V_T$.

Validity Partition Coefficient And Exponential Separation (VPCAES)

Οι Wu και Yang [48] πρότειναν τον ακόλουθο δείκτη εγκυρότητας:

$$V_{PCAES} = \sum_{i=1}^{c}\sum_{j=1}^{n}\frac{u_{ij}^2}{\mu_M} - \sum_{i=1}^{c}\exp\left(-\min_{k\neq i}\{\frac{||v_i - v_k||^2}{\beta_T}\}\right) \tag{3.31}$$

όπου

$$\mu_M = \underset{1\leq i\leq c}{min}\Big\{\sum_{j=1}^{n} u_{ij}^2\Big\}, \beta_T = \frac{\sum_{l=1}^{c}||v_l - \bar{v}||^2}{c} \text{ και } \bar{v} = \sum_{j=1}^{n}\frac{x_j}{n} \tag{3.32}$$

Όσο μεγαλύτερη είναι η τιμή του VPCAES, τόσο καλύτερα διαχωρισμένες και συμπαγείς είναι οι συστάδες. Ο καλύτερος αριθμός συστάδων c^* βρίσκεται λύνοντας το πρόβλημα μεγιστοποίησης $\underset{2\leq c\leq n-1}{max} V_{PCAES}$.

3.4 Προτεινόμενη διαδικασία

3.4.1 Προεξεργασία δεδομένων

Ένα σημαντικό κομμάτι του προτεινόμενου αλγορίθμου, είναι η προπεξεργασία των αρχείων καταγραφής. Κατ' αρχάς, πρέπει να "καθαρίσουμε" τα αρχεία από τους κωδικούς σφαλμάτων. Σύμφωνα με το [11], αποφασίσαμε να κρατήσουμε μόνο εκείνους τους κωδικούς που υποδεικνύουν ότι η ζητούμενη σελίδα ανακτήθηκε επιτυχώς. Στη συνέχεια, αναγνωρίζουμε τους χρήστες. Χωρίζουμε το αρχείο καταγραφής ανά μέρα και και αναγνωρίζουμε ένα μοναδικό χρήστη από την τριπλέτα *(Διεύθυνση IP, Λειτουργικό σύστημα, Έκδοση προγράμματος περιήγησης)*. Μπορούμε να κάνουμε την υπόθεση εργασίας ότι ένας χρήστης δεν έχει αλλάξει όλα αυτά τα στοιχεία κατά τη διάρκεια μιας ημέρας και επομένως δεν έχουμε αναγνωρίσει ένα μοναδικό user σαν πολλούς ξεχωριστούς. Μετά από αυτό το στάδιο, αφαιρούμε από το αρχείο καταγραφής τα spiders/crawlers/bots. Ακολουθούμε δύο διαφορετικά μονοπάτια για αυτό το βήμα. Καθαρίζουμε το αρχείο καταγραφής κατά χρήστη και κατά session. Για την πρώτη περίπτωση, αν παρατηρήσουμε ότι ένας χρήστης έχει στο πεδίο του User Agent την πληροφορία ότι είναι bot, τότε αφαιρούμε όλα τα αιτήματά του. Κατά session, αν δούμε ότι ένας χρήστης σαν πρώτο αίτημα σε μια σειρά από διαδοχικά αιτήματα ζητάει το αρχείο robots.txt τότε αφαιρούμε όλα τα μετέπειτα αιτήματά του. Τέλος, για τους χρήστες που

έχουμε βρει, επεξεργαζόμαστε τα αιτήματά τους. Τα μοντέρνα συστήματα ανάπτυξης εφαρμογών διαδικτύου, μαζί με ένα αίτημα το οποίο κάνει ο χρήστης, στέλνουν πολλά αρχεία, όπως για παράδειγμα .css, .js, εικόνες κ.α., που δεν έχουν ζητηθεί ρητώς από τον επισκέπτη. Φιλτράρουμε τα αιτήματα, με σκοπό να κρατήσουμε μόνο τα "χρήσιμα", δηλαδή τα αρχεία .html, .php, ή αυτά που δεν έχουν καμία κατάληξη.

3.4.2 Αναπαράσταση

Μετά την προεπεξεργασία του αρχείου καταγραφής, διαθέτουμε ένα σύνολο από χρήστες και για καθέναν από αυτούς μία λίστα από τα αιτήματά τους. Προκειμένου τα δεδομένα μας να επεξεργαστούν από τον αλγόριθμο ασαφούς συσταδοποίησης c-κέντρων πρέπει να τα αναπαραστήσουμε σε μία αριθμητική μορφή. Για αυτό το σκοπό, κάνουμε μετασχηματισμούς στο χρόνο του αιτήματος. Ο χρόνος του αιτήματος αντικαθίσταται με τη διαφορά του χρόνου του συγκεκριμένου αιτήματος από το πρώτο αίτημα που έκανε ο χρήστης μέσα στην ημέρα (σε δευτερόλεπτα). Στη συνέχεια, αναθέτουμε σε κάθε αίτημα ένα υποψήφιο αναγνωριστικό session. Το αναγνωριστικό αυτό ξεκινάει από ένα και αυξάνεται κάθε φορά που δύο διαδοχικά αιτήματα έχουν χρονική διαφορά παραπάνω από 10 λεπτά ή κάθε φορά που το πεδίο παραπομπής είναι κενό ή έχει άλλες ιστοσελίδες, πέρα από αυτή της οποίας εξετάζουμε τα αρχεία καταγραφής. Μετά από αυτό το στάδιο, έχουμε για κάθε χρήστη ένα διάνυσμα της μορφής *(Δευτερόλεπτα από το πρώτο αίτημα της ημέρας, Υποψήφιο αναγνωριστικό session)*. Το όριο των δέκα λεπτών τέθηκε αφενός με βάση τη βιβλιογραφία, αφετέρου διότι προσπαθήσαμε να εφαρμόσουμε μια υπερευαίσθητη προσέγγιση σε αυτό το πεδίο.

3.4.3 Συσταδοποίηση δεδομένων

Προκειμένου να χωρίσουμε τα δεδομένα σε συστάδες, χρησιμοποιούμε τον αλγόριθμο της ασαφούς συσταδοποίησης c-κέντρων. Προτιμήσαμε αυτόν τον αλγόριθμο από την συσταδοποίηση k-κέντρων επειδή τα sessions ενός χρήστη μπορεί μερικές φορές να έχουν όρια που δεν είναι τόσο ευδιάκριτα. Ο αλγόριθμος της ασαφούς συσταδοποίησης c-κέντρων με μία τυχαία αρχικοποίηση του πίνακα ασαφούς διαμέρισης μπορεί να "κολλήσει" σε ένα τοπικό ελάχιστο και προκειμένου να αποφύγουμε αυτή την κατάσταση, αρχικοποιήσαμε τον πίνακα ασαφούς διαμέρισης χρησιμοποιώντας τον αλγόριθμο της ασαφούς συσταδοποίησης. Επειδή η αναπαράσταση των δεδομένων μας στον x-άξονα είναι σε δευτερόλεπτα και οι αποστάσεις των σημείων δεδομένων μπορεί να γίνουν πολύ μεγάλες, με αποτέλεσμα η συνάρτηση που υπολογίζει την πυκνότητα ενός σημείου για την επιλογή των κέντρων να οδηγείται συνεχώς στην μονάδα, οδηγώντας σε κοντινά ή ακόμα και τα ίδια σημεία ως κέντρα των συστάδων σε διαδοχικές εκτελέσεις του αλγορίθμου, μετατρέπουμε τα δευτερόλεπτα σε λεπτά. Επίσης σύμφωνα με το [49], θέσαμε την παράμετρο $r_\alpha = 0.5$ και την παράμετρο $r_b = 1.5 r_\alpha$. Στη συνέχεια, τρέχουμε τον αλγόριθμο ασαφούς συσταδοποίησης c-κέντρων με τον πίνακα που μας έδωσε η ασαφής συσταδοποίηση για αριθμό συστάδων από ένα μέχρι το μεγαλύτερο υποψήφιο αναγνωριστικό session, τιμές του δείκτη ασάφειας $m = 2, 3, 5, 8$ για την ευκλείδεια, τη manhattan, τη chebyshev και τη minkowski ως μετρικές απόστασης. Από τους δείκτες εγκυρότητας που αναφέρθηκαν στην προηγούμενη παράγραφο υπολογίζουμε τους $V_{PC}, V_{FS}, V_{XB}, V_{PCAES}$. Στη συνέχεια αναλύουμε τα αποτελέσματα της συσταδοποίησης προκειμένου να δούμε αν είναι λογικά με βάση την αναπαράστασή μας και αποφασίζουμε για τον κα-

τάλληλο αριθμό συστάδων από τα αποτελέσματα των δεικτών. Συγκεκριμένα, υπολογίζουμε όλους τους δείκτες εγκυρότητας για κάθε συνδυασμό δείκτη ασάφειας και απόστασης και κρατάμε εκείνες τις τιμές για τις οποίες η μέση απόκλιση των τιμών των τεσσάρων δεικτών είναι ελάχιστη. Στη συνέχεια, για εκείνες τις τετράδες για τις οποίες η μέση απόκλιση ελαχιστοποιείται, στρογγυλοποιούμε τον μέσο όρο των τιμών των δεικτών στον πλησιέστερο ακέραιο και αυτό το αποτέλεσμα θεωρούμε ότι είναι ο αριθμός των sessions που βρήκε το σύστημά μας. Στις περισσότερες περιπτώσεις υπάρχουν αρκετοί συνδυασμοί δείκτη ασάφειας και απόστασης που "συμφωνούν" μεταξύ τους, αλλά ο μέσος όρος τους είναι ίδιος. Αν υπάρχουν παραπάνω από ένας μέσος όρος ως αριθμός sessions, δηλαδή έχουν παραχθεί παραπάνω από μία συσταδοποιήσεις που έχουν νόημα, τότε κρατάμε αυτή με τις περισσότερες συστάδες. Ακολουθεί ο αλγόριθμός συγκεντρωτικά:

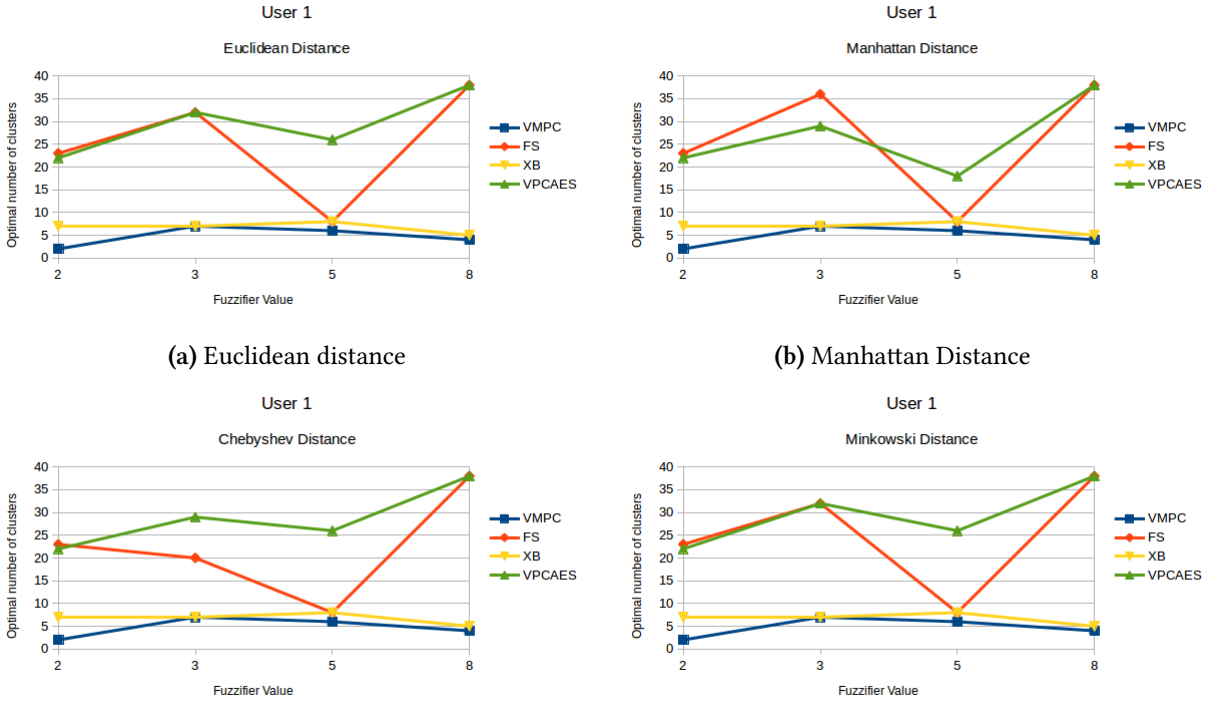
Αλγόριθμος 2 Ψευδοκώδικας προτεινόμενου αλγορίθμου

```
function FindSession(web logs)
    Split the web logs per day
    Keep only useful status codes
    Identify every user by the triple (IP, OS, Browser)
    for every user and his requests do
        if agent is bot OR first request of user is robots.txt then
            Remove user and all of his requests
        end if
    end for
    for every request a user has made do
        Keep only "useful" links(those who end with .php, .html or null)
        Remove all other information except referrer and time the request was done
        candidate_session_id ← 0
        if two consecutive requests have more than 10 minutes between them
             OR referrer is null OR referrer does not come from the website then
            candidate_session_id += 1
        end if
        Every request is represented by the tuple
             (offset from first request of the day, candidate_session_id)
        Store all these requests in an array
        for the array of requests as tuples do
            for distance = euclidean, manhattan, chebyshev, minkowski do
                for fuzzifier = 2, 3, 4, 8 do
                    for i in (1, max(candidate_session_id)) do
                        Cluster the data using subtractive clustering with
                                number of clusters equal to i
                        Compute the fuzzy partition matrix with the centers
                                of the subtractive clustering
                        Run Fuzzy C-Means Clustering with the above array for specific fuzzifier
                                value and distance
                        Compute V_MPC, V_XB, V_FS, V_PCAES
                    end for
                end for
                Store the values of the indices in an row in an array
            end for
            for the array of indices do
                Compute standard deviation for every row
                Find the row(s) that have the minimum standard deviation
                Compute the mean of these rows
            end for
            Number of session is the maximum of the above means
        end for
    end for
end function
```

3.5 Μετρήσεις-Αποτελέσματα

Σε αυτό το σημείο της εργασίας μας, θα παρουσιάσουμε τα αποτελέσματα από την εκτέλεση του παραπάνω αλγορίθμου σε πραγματικά αρχεία καταγραφής. Τα αρχεία καταγραφής στα οποία εκτελέστηκε ο παραπάνω αλγόριθμος προέρχονταν από τη διαδικτυακή κοινότητα φοιτητών της σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών[22]. Απομονώθηκαν 20 χρήστες οι οποίοι είχαν τουλάχιστον πενήντα "χρήσιμα" requests και ο αλγόριθμος εκτελέστηκε σε αυτούς. Ακολουθούν οι γραφικές παραστάσεις των τεσσάρων δεικτών για κάθε χρήστη ανά απόσταση και ανά τιμή του δείκτη ασάφειας. Μετά την παρουσίαση των διαγραμμάτων ακολουθεί ο συγκεντρωτικός πίνακας για όλους τους δείκτες εγκυρότητας που υλοποιήθηκαν καθώς για τον πραγματικό αριθμό sessions μετά από επιθεώρηση, αλλά και την τιμή του υποψήφιου αναγνωριστικού session.

3.5.1 Παρουσίαση διαγραμμάτων των δεικτών εγκυρότητας



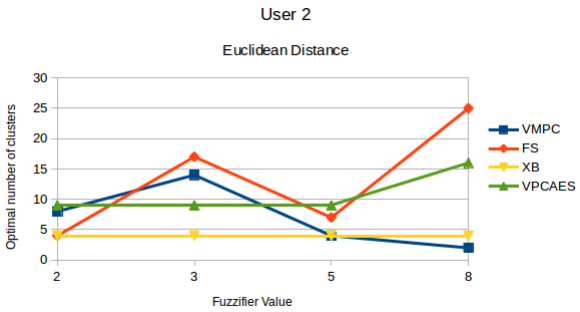
(a) Euclidean distance

(b) Manhattan Distance

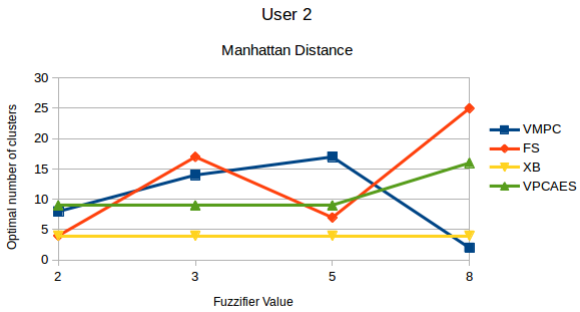
(c) Chebyshev Distance

(d) Minkowski Distance

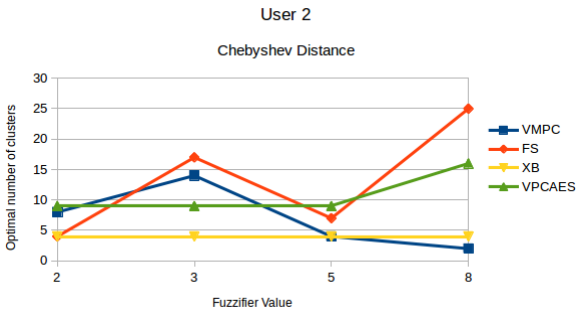
Σχήμα 3.4: Διαγράμματα για τον 1ο χρήστη



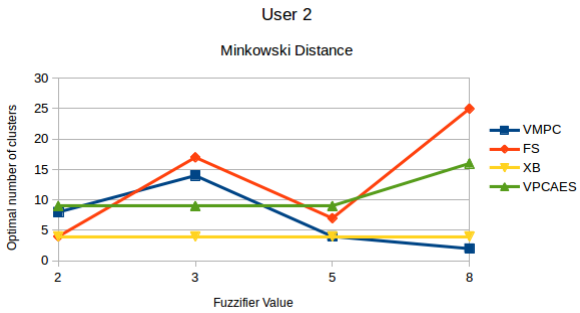
(a) Euclidean distance



(b) Manhattan Distance

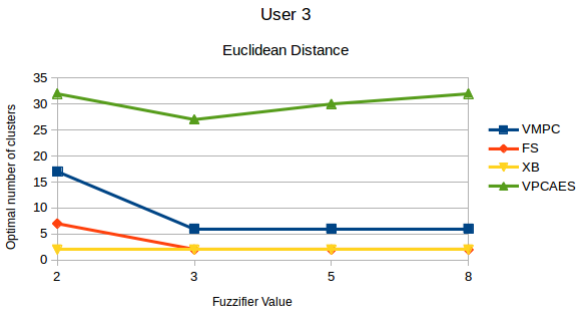


(c) Chebyshev Distance

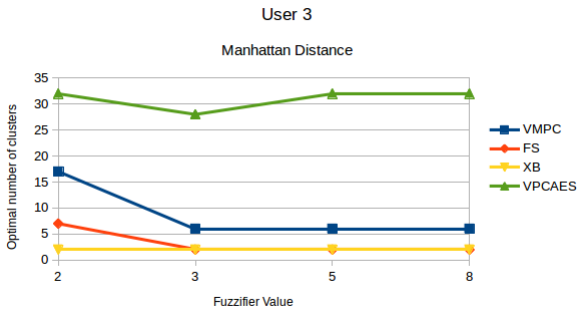


(d) Minkowski Distance

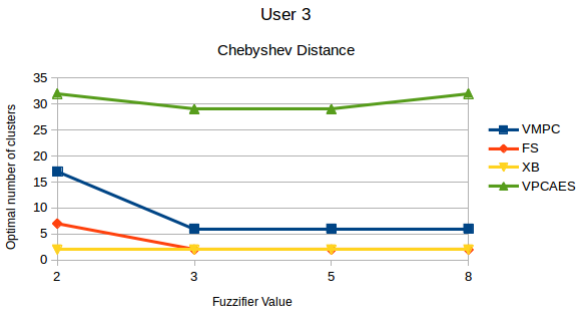
Σχήμα 3.5: Διαγράμματα για τον 2ο χρήστη



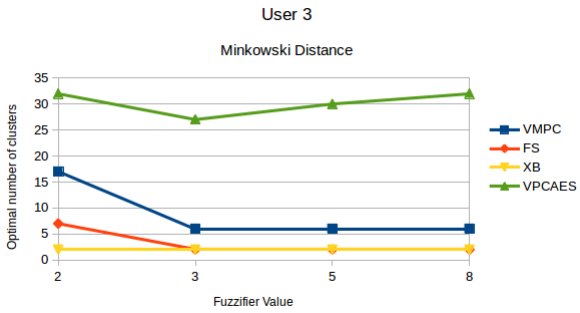
(a) Euclidean distance



(b) Manhattan Distance

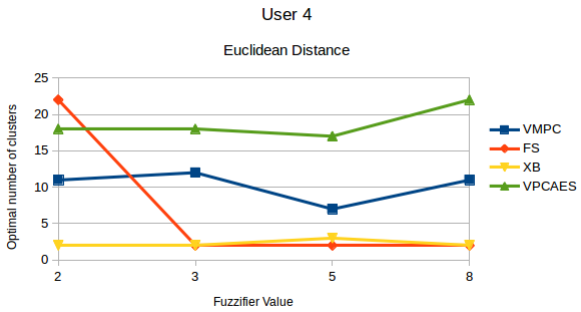


(c) Chebyshev Distance

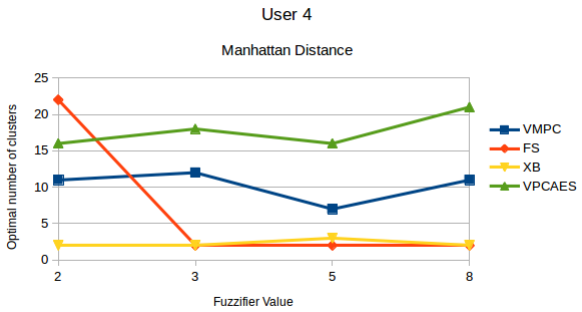


(d) Minkowski Distance

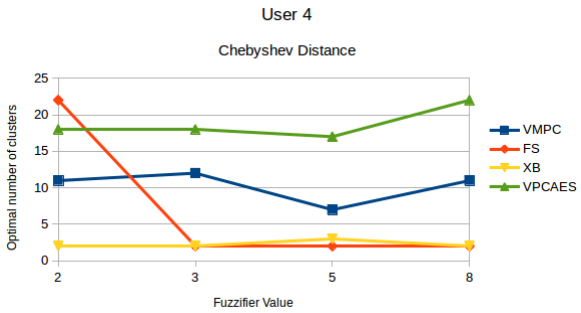
Σχήμα 3.6: Διαγράμματα για τον 3ο χρήστη



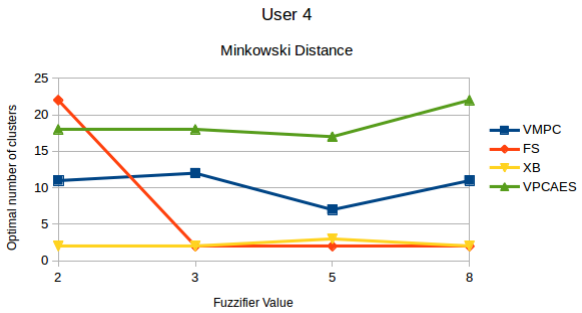
(a) Euclidean distance



(b) Manhattan Distance

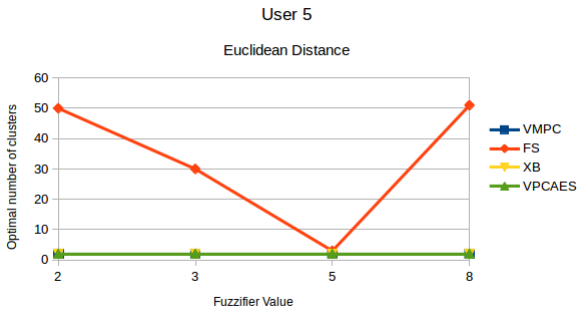


(c) Chebyshev Distance

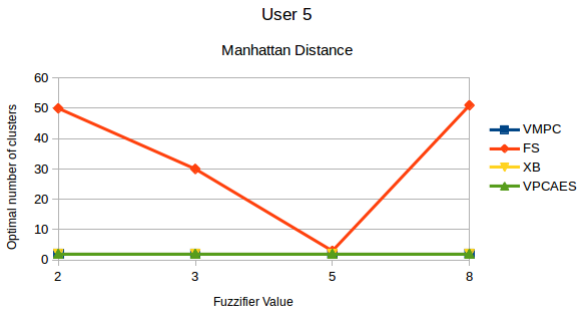


(d) Minkowski Distance

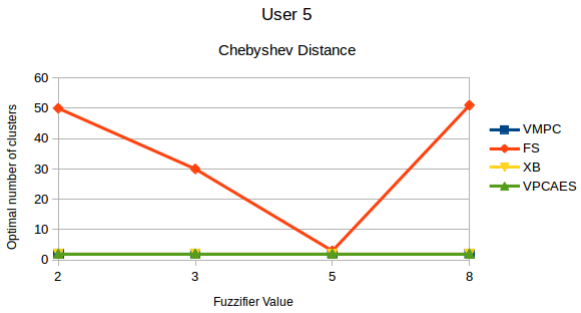
Σχήμα 3.7: Διαγράμματα για τον 4ο χρήστη



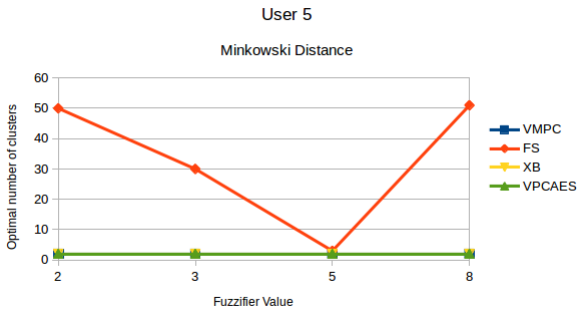
(a) Euclidean distance



(b) Manhattan Distance

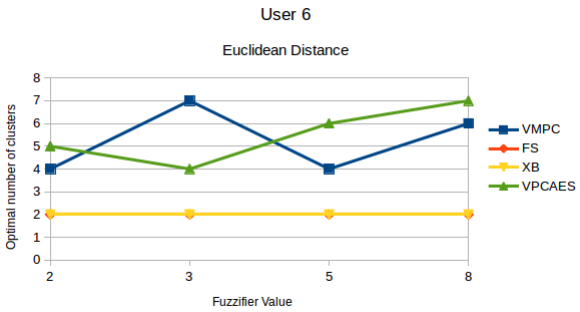


(c) Chebyshev Distance

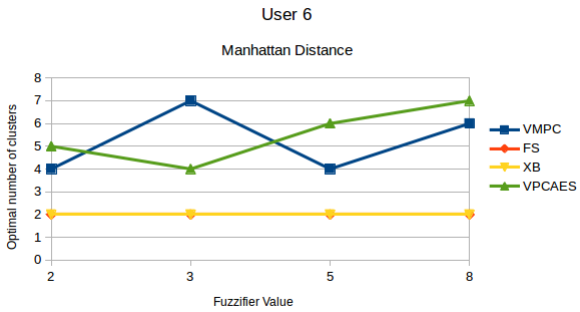


(d) Minkowski Distance

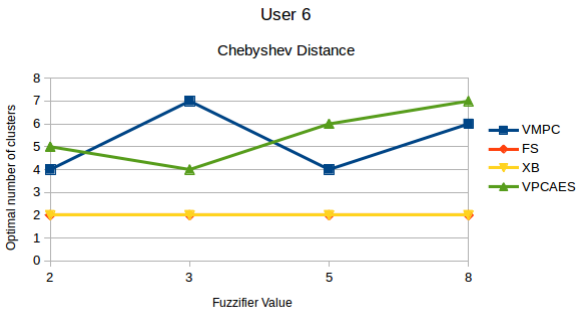
Σχήμα 3.8: Διαγράμματα για τον 5ο χρήστη



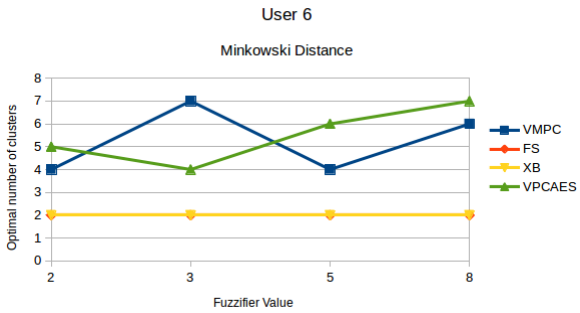
(a) Euclidean distance



(b) Manhattan Distance

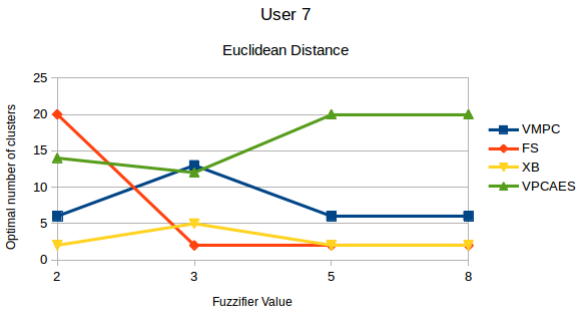


(c) Chebyshev Distance

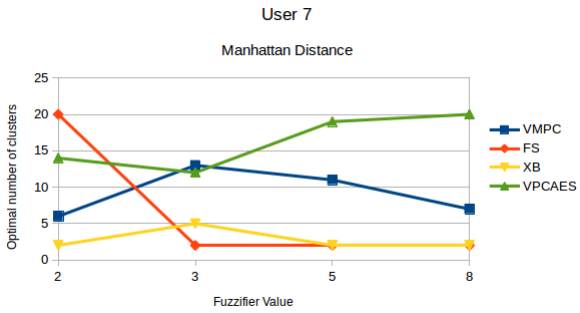


(d) Minkowski Distance

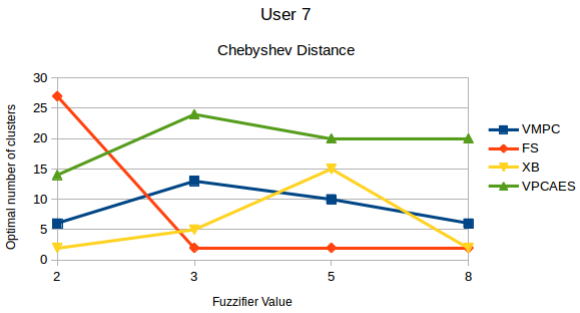
Σχήμα 3.9: Διαγράμματα για τον 6ο χρήστη



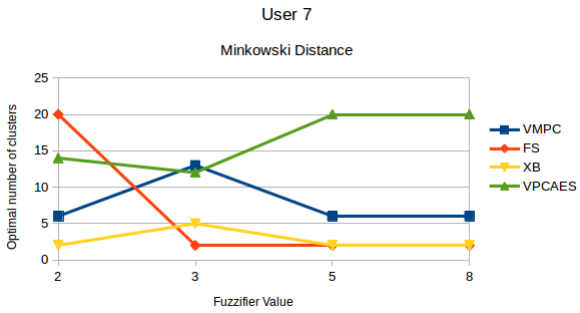
(a) Euclidean distance



(b) Manhattan Distance

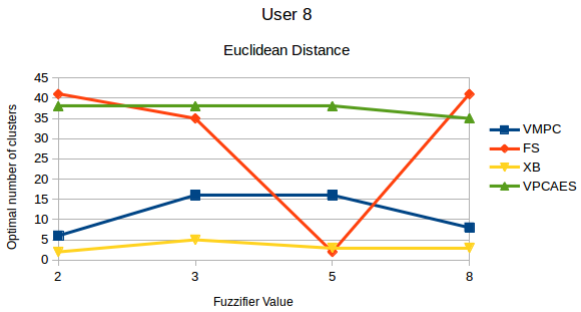


(c) Chebyshev Distance

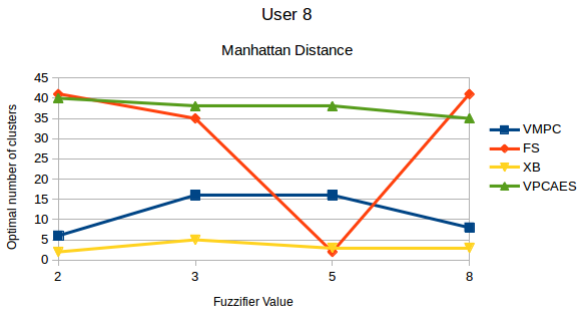


(d) Minkowski Distance

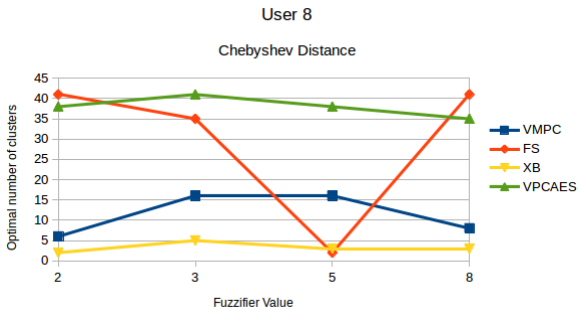
Σχήμα 3.10: Διαγράμματα για τον 7ο χρήστη



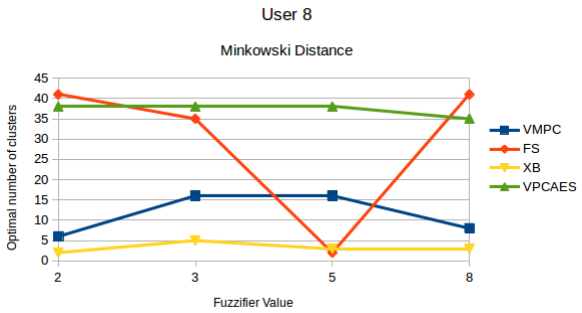
(a) Euclidean distance



(b) Manhattan Distance

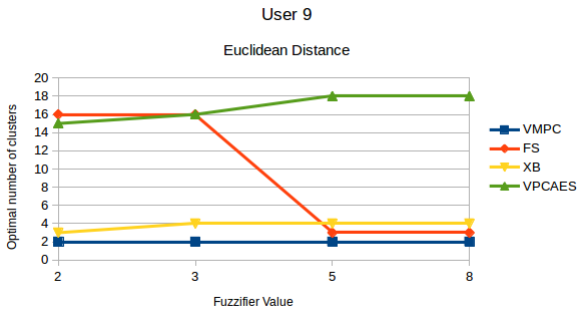


(c) Chebyshev Distance

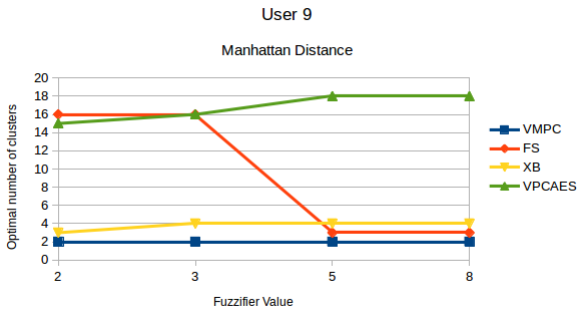


(d) Minkowski Distance

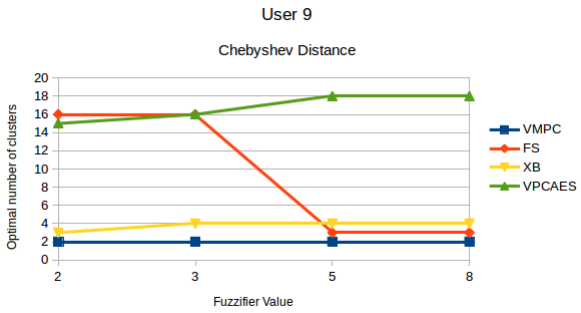
Σχήμα 3.11: Διαγράμματα για τον 8ο χρήστη



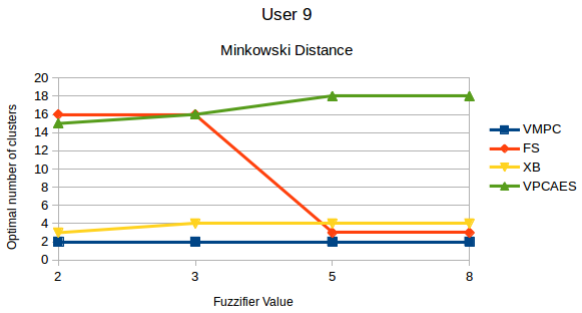
(a) Euclidean distance



(b) Manhattan Distance

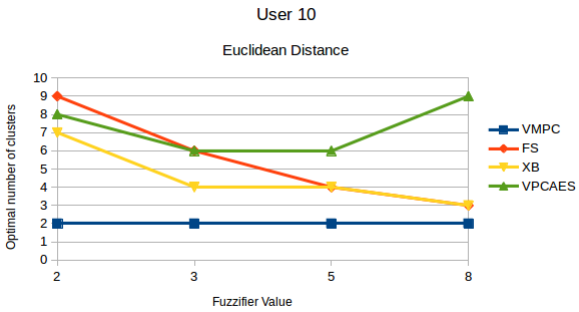


(c) Chebyshev Distance

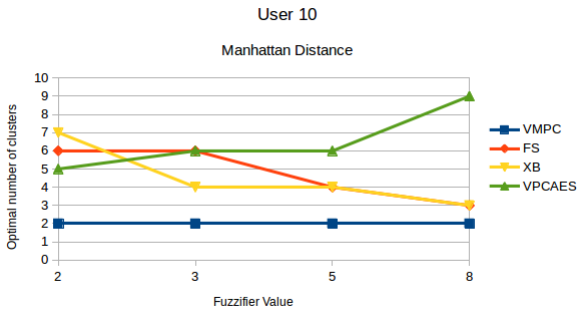


(d) Minkowski Distance

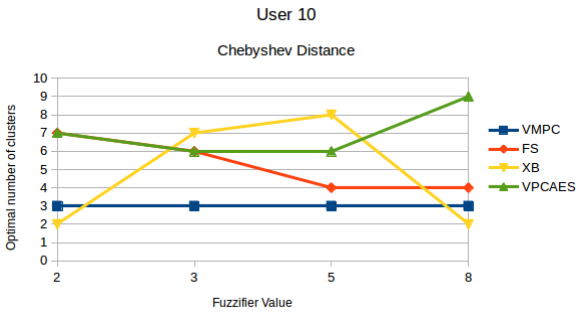
Σχήμα 3.12: Διαγράμματα για τον 9ο χρήστη



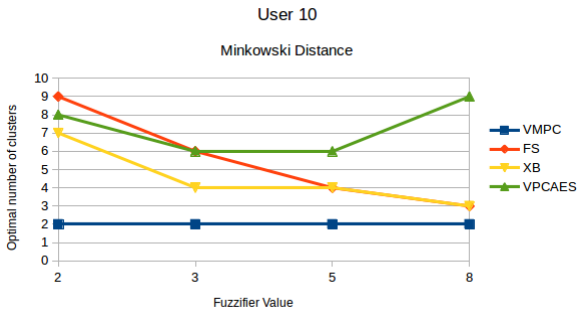
(a) Euclidean distance



(b) Manhattan Distance

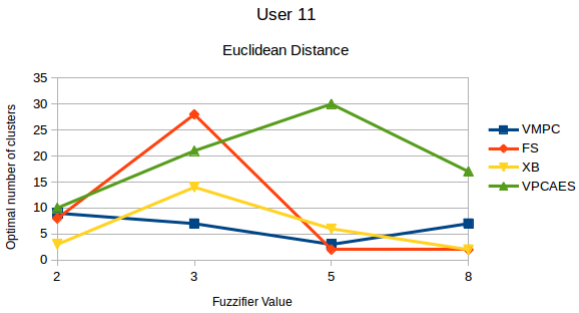


(c) Chebyshev Distance

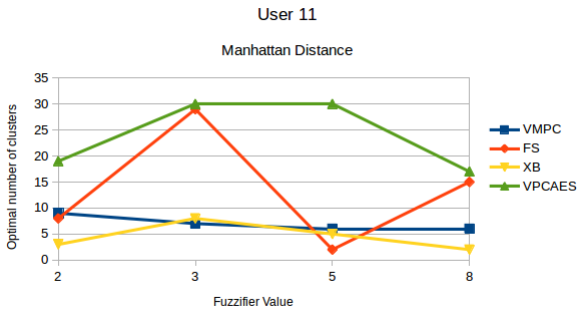


(d) Minkowski Distance

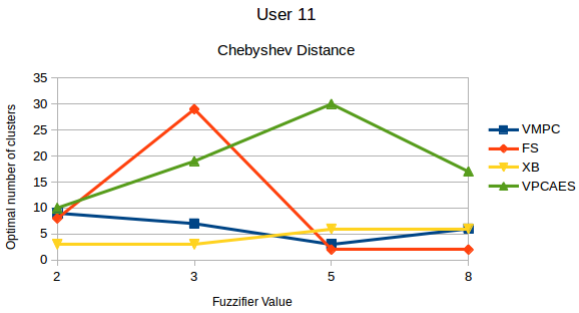
Σχήμα 3.13: Διαγράμματα για τον 10ο χρήστη



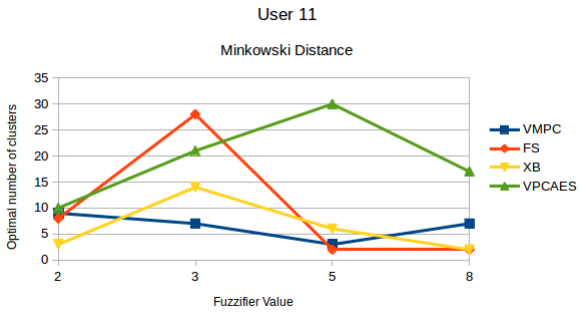
(a) Euclidean distance



(b) Manhattan Distance

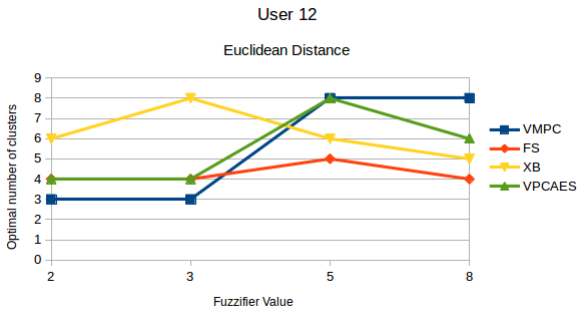


(c) Chebyshev Distance

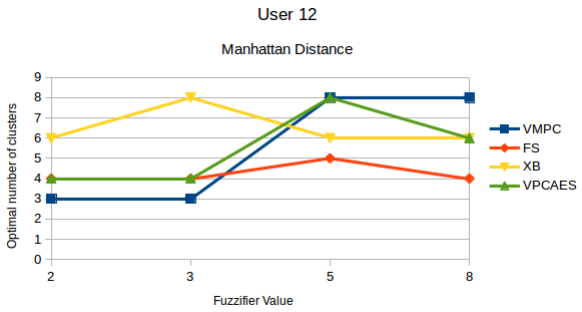


(d) Minkowski Distance

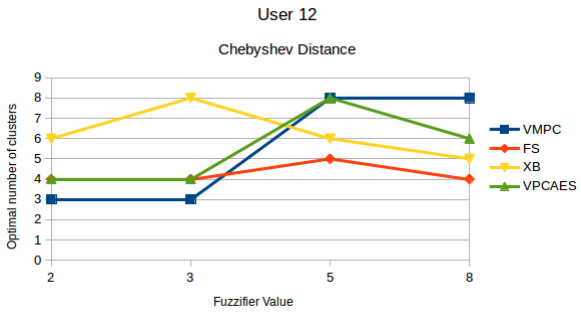
Σχήμα 3.14: Διαγράμματα για τον 11ο χρήστη



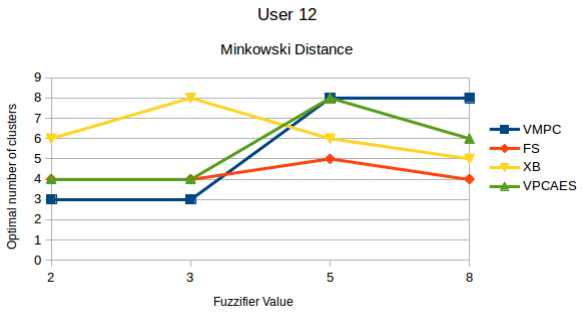
(a) Euclidean distance



(b) Manhattan Distance

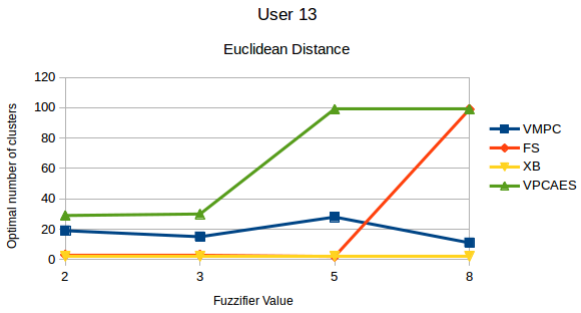


(c) Chebyshev Distance

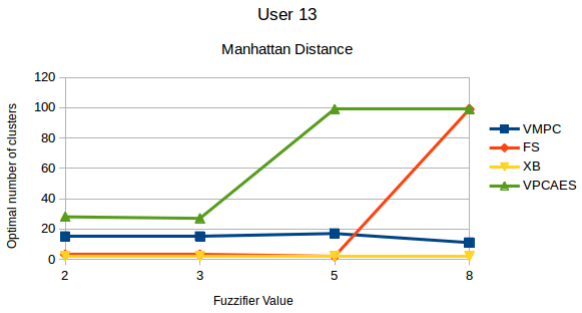


(d) Minkowski Distance

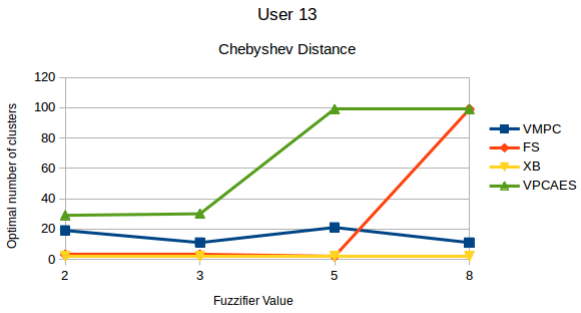
Σχήμα 3.15: Διαγράμματα για τον 12ο χρήστη



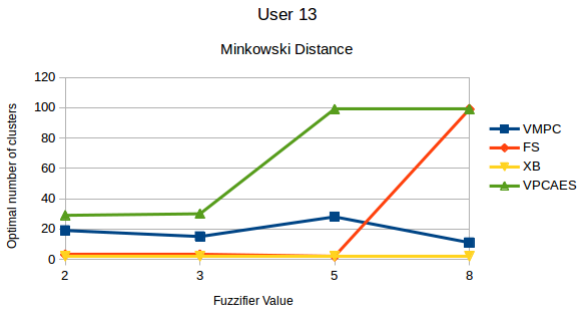
(a) Euclidean distance



(b) Manhattan Distance

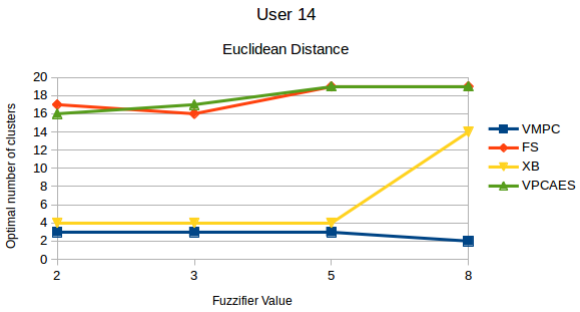


(c) Chebyshev Distance

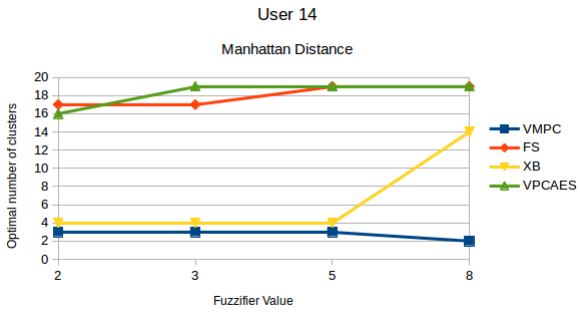


(d) Minkowski Distance

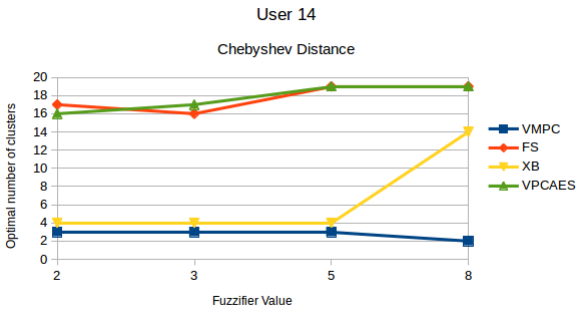
Σχήμα 3.16: Διαγράμματα για τον 13ο χρήστη



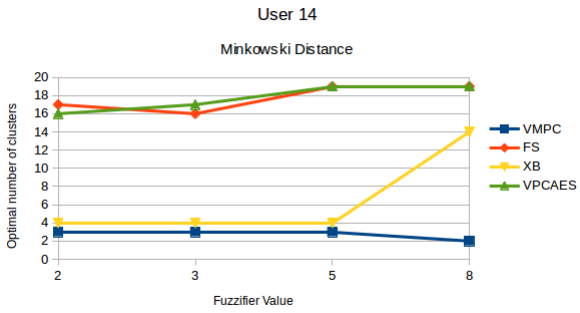
(a) Euclidean distance



(b) Manhattan Distance

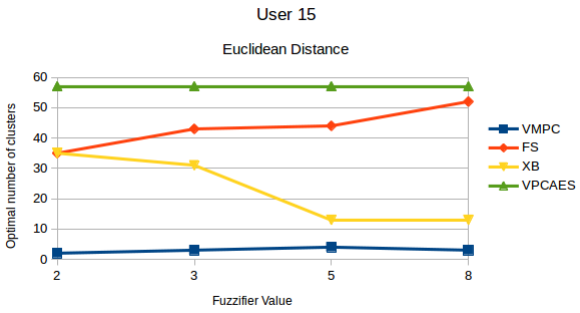


(c) Chebyshev Distance

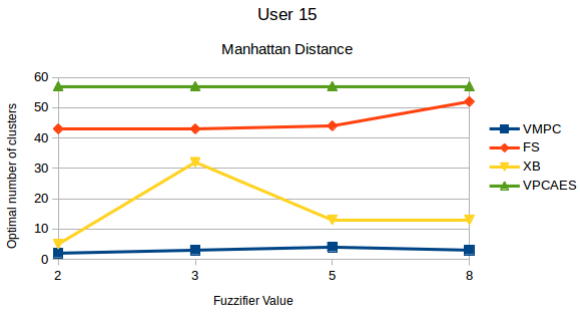


(d) Minkowski Distance

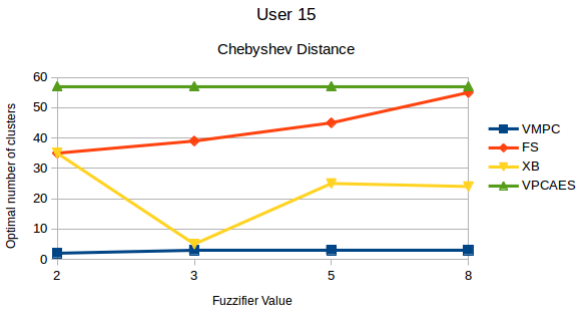
Σχήμα 3.17: Διαγράμματα για τον 14ο χρήστη



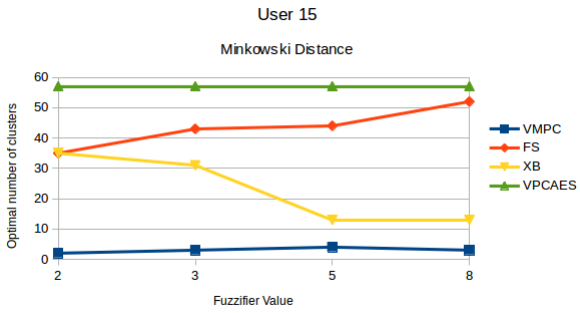
(a) Euclidean distance



(b) Manhattan Distance

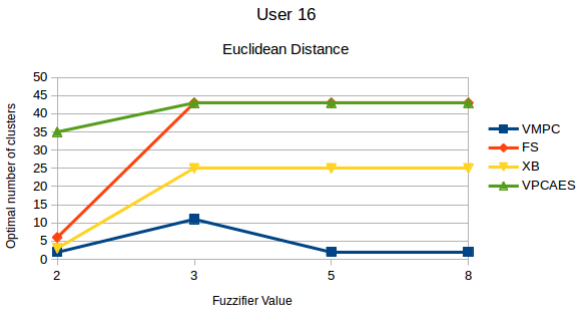


(c) Chebyshev Distance

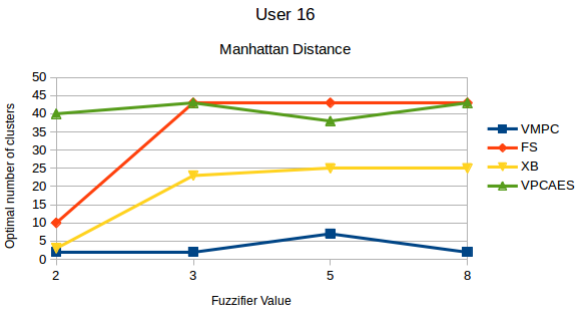


(d) Minkowski Distance

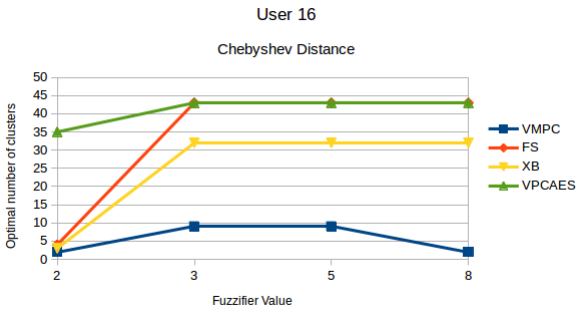
Σχήμα 3.18: Διαγράμματα για τον 15ο χρήστη



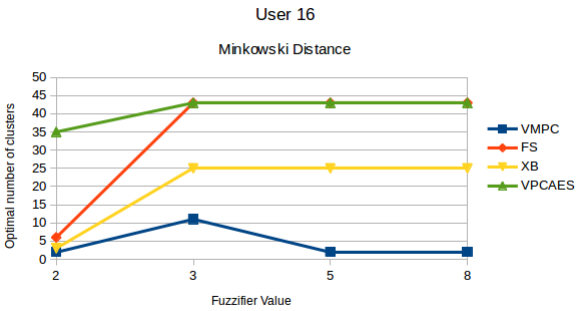
(a) Euclidean distance



(b) Manhattan Distance

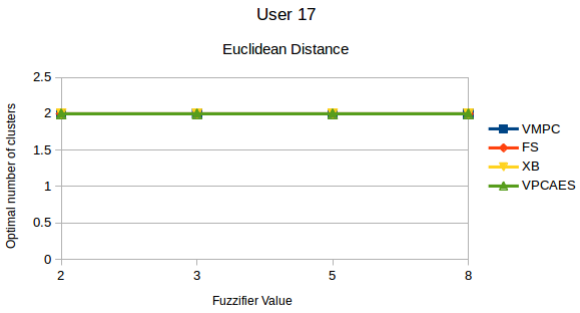


(c) Chebyshev Distance

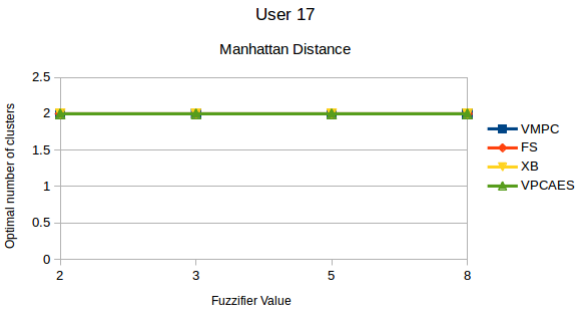


(d) Minkowski Distance

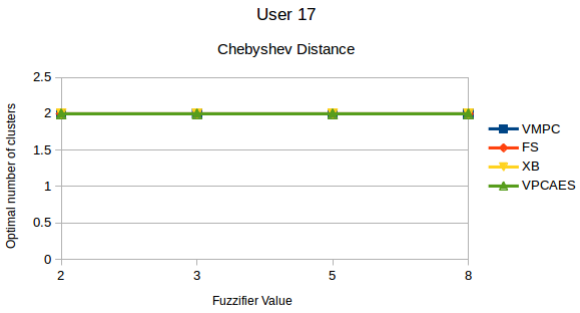
Σχήμα 3.19: Διαγράμματα για τον 16ο χρήστη



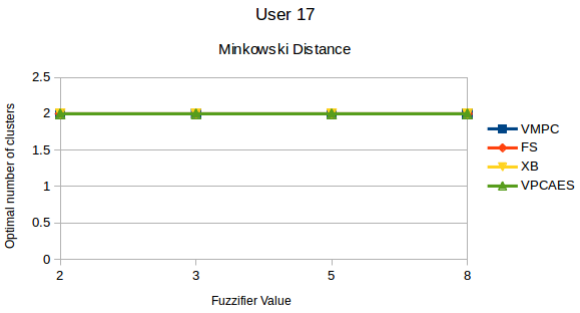
(a) Euclidean distance



(b) Manhattan Distance

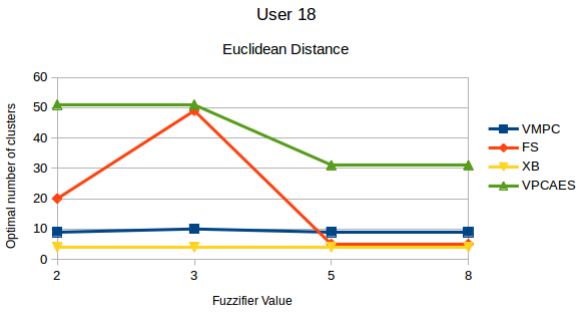


(c) Chebyshev Distance

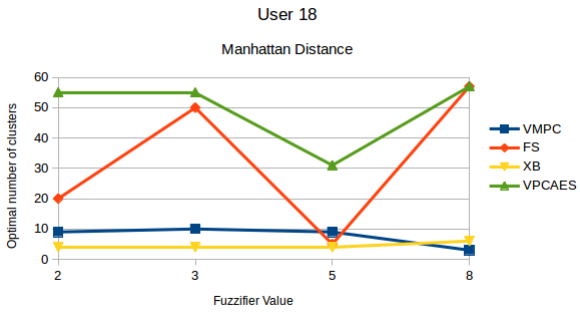


(d) Minkowski Distance

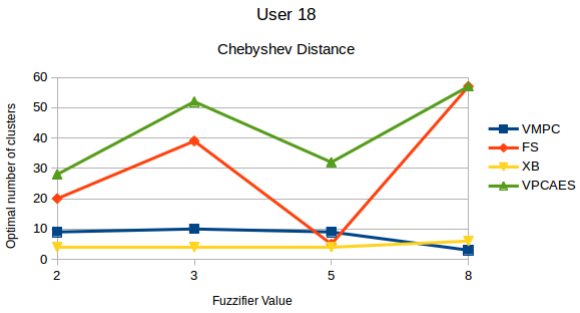
Σχήμα 3.20: Διαγράμματα για τον 17ο χρήστη



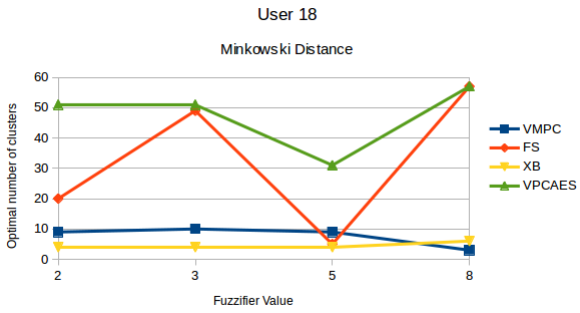
(a) Euclidean distance



(b) Manhattan Distance

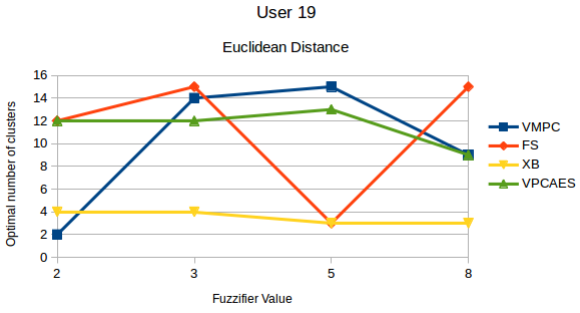


(c) Chebyshev Distance

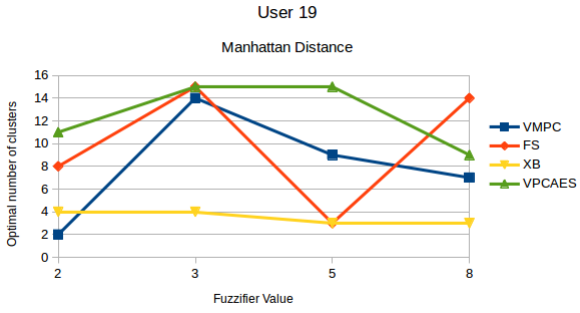


(d) Minkowski Distance

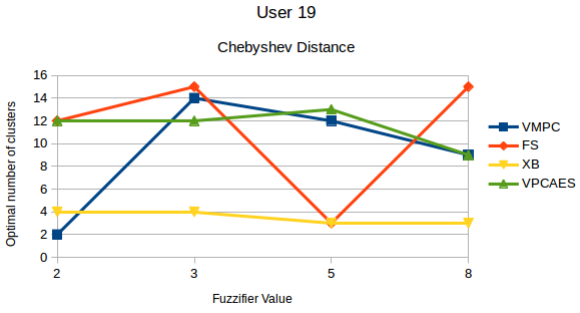
Σχήμα 3.21: Διαγράμματα για τον 18ο χρήστη



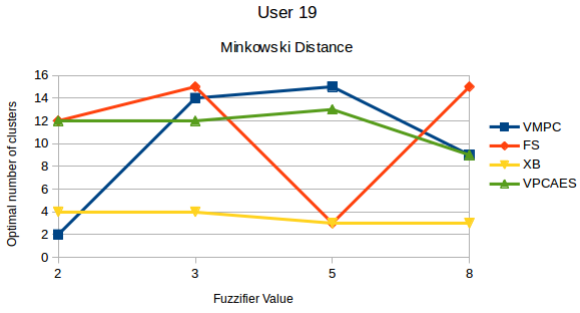
(a) Euclidean distance



(b) Manhattan Distance

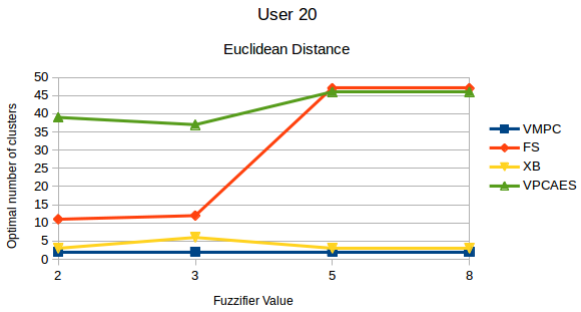


(c) Chebyshev Distance

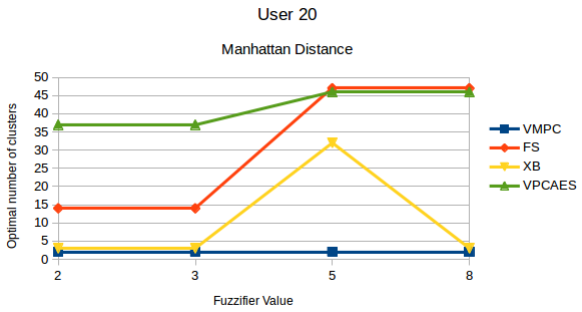


(d) Minkowski Distance

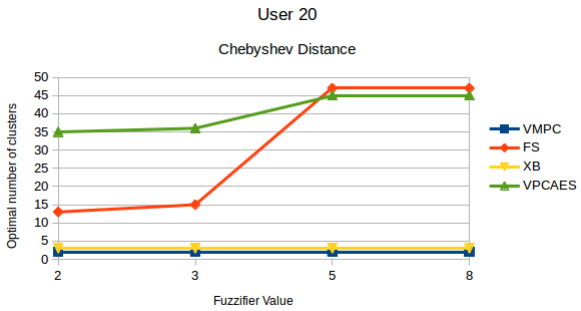
Σχήμα 3.22: Διαγράμματα για τον 19ο χρήστη



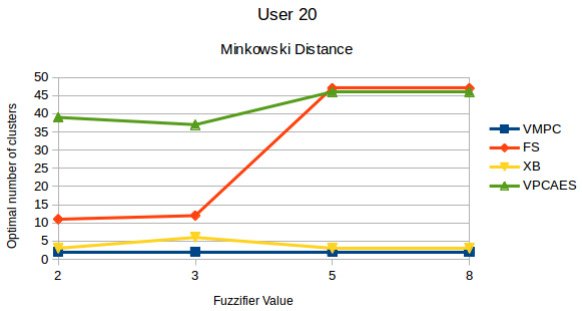
(a) Euclidean distance



(b) Manhattan Distance



(c) Chebyshev Distance



(d) Minkowski Distance

Σχήμα 3.23: Διαγράμματα για τον 20ο χρήστη

3.5.2 Πίνακες αποτελεσμάτων

File	Distance	Fuzzifier	VMPC	FS	XB	VPCAES	VT	VPC	VPE	VK	Sessions by inspection	Candidate Session Id
shmmy1	euclidean	2	2	23	7	22	2	38	38	2	5 - 15	39
shmmy1	euclidean	3	7	32	7	32	2	2	2	2	5 - 15	39
shmmy1	euclidean	5	6	8	8	26	2	2	2	2	5 - 15	39
shmmy1	euclidean	8	4	38	5	38	2	2	2	2	5 - 15	39
shmmy1	manhattan	2	2	23	7	22	2	38	37	2	5 - 15	39
shmmy1	manhattan	3	7	36	7	29	2	2	2	2	5 - 15	39
shmmy1	manhattan	5	6	8	8	18	2	2	2	2	5 - 15	39
shmmy1	manhattan	8	4	38	5	38	2	2	2	2	5 - 15	39
shmmy1	chebyshev	2	2	23	7	22	2	38	38	2	5 - 15	39
shmmy1	chebyshev	3	7	20	7	29	2	2	2	2	5 - 15	39
shmmy1	chebyshev	5	6	8	8	26	2	2	2	2	5 - 15	39
shmmy1	chebyshev	8	4	38	5	38	2	2	2	2	5 - 15	39
shmmy1	minkowski	2	2	23	7	22	2	38	38	2	5 - 15	39
shmmy1	minkowski	3	7	32	7	32	2	2	2	2	5 - 15	39
shmmy1	minkowski	5	6	8	8	26	2	2	2	2	5 - 15	39
shmmy1	minkowski	8	4	38	5	38	2	2	2	2	5 - 15	39

Πίνακας 3.2: Πίνακας αποτελεσμάτων για τον 1ο χρήστη

File	Distance	Fuzzifier	VMPC	FS	XB	VPCAES	VT	VPC	VPE	VK	Sessions by inspection	Candidate Session Id
shmmy2	euclidean	2	8	4	4	9	2	2	2	2	3 - 8	26
shmmy2	euclidean	3	14	17	4	9	2	2	2	4	3 - 8	26
shmmy2	euclidean	5	4	7	4	9	2	2	2	2	3 - 8	26
shmmy2	euclidean	8	2	25	4	16	2	2	2	2	3 - 8	26
shmmy2	manhattan	2	8	4	4	9	2	2	2	2	3 - 8	26
shmmy2	manhattan	3	14	17	4	9	2	2	2	4	3 - 8	26
shmmy2	manhattan	5	17	7	4	9	2	2	2	2	3 - 8	26
shmmy2	manhattan	8	2	25	4	16	2	2	2	2	3 - 8	26
shmmy2	chebyshev	2	8	4	4	9	2	2	2	2	3 - 8	26
shmmy2	chebyshev	3	14	17	4	9	2	2	2	4	3 - 8	26
shmmy2	chebyshev	5	4	7	4	9	2	2	2	2	3 - 8	26
shmmy2	chebyshev	8	2	25	4	16	2	2	2	2	3 - 8	26
shmmy2	minkowski	2	8	4	4	9	2	2	2	2	3 - 8	26
shmmy2	minkowski	3	14	17	4	9	2	2	2	4	3 - 8	26
shmmy2	minkowski	5	4	7	4	9	2	2	2	2	3 - 8	26
shmmy2	minkowski	8	2	25	4	16	2	2	2	2	3 - 8	26

Πίνακας 3.3: Πίνακας αποτελεσμάτων για τον 2ο χρήστη

File	Distance	Fuzzifier	VMPC	FS	XB	VPCAES	VT	VPC	VPE	VK	Sessions by inspection	Candidate Session Id
shmmy3	euclidean	2	17	7	2	32	2	2	2	2	5 - 8	33
shmmy3	euclidean	3	6	2	2	27	2	2	2	2	5 - 8	33
shmmy3	euclidean	5	6	2	2	30	2	2	2	2	5 - 8	33
shmmy3	euclidean	8	6	2	2	32	2	2	2	2	5 - 8	33
shmmy3	manhattan	2	17	7	2	32	2	2	2	2	5 - 8	33
shmmy3	manhattan	3	6	2	2	28	2	2	2	2	5 - 8	33
shmmy3	manhattan	5	6	2	2	32	2	2	2	2	5 - 8	33
shmmy3	manhattan	8	6	2	2	32	2	2	2	2	5 - 8	33
shmmy3	chebyshev	2	17	7	2	32	2	2	2	2	5 - 8	33
shmmy3	chebyshev	3	6	2	2	29	2	2	2	2	5 - 8	33
shmmy3	chebyshev	5	6	2	2	29	2	2	2	2	5 - 8	33
shmmy3	chebyshev	8	6	2	2	32	2	2	2	2	5 - 8	33
shmmy3	minkowski	2	17	7	2	32	2	2	2	2	5 - 8	33
shmmy3	minkowski	3	6	2	2	27	2	2	2	2	5 - 8	33
shmmy3	minkowski	5	6	2	2	30	2	2	2	2	5 - 8	33
shmmy3	minkowski	8	6	2	2	32	2	2	2	2	5 - 8	33

Πίνακας 3.4: Πίνακας αποτελεσμάτων για τον 3ο χρήστη

File	Distance	Fuzzifier	VMPC	FS	XB	VPCAES	VT	VPC	VPE	VK	Sessions by inspection	Candidate Session Id
shmmy4	euclidean	2	11	22	2	18	2	2	2	2	3 - 6	23
shmmy4	euclidean	3	12	2	2	18	2	2	2	2	3 - 6	23
shmmy4	euclidean	5	7	2	3	17	2	2	2	2	3 - 6	23
shmmy4	euclidean	8	11	2	2	22	2	2	2	2	3 - 6	23
shmmy4	manhattan	2	11	22	2	16	2	2	2	2	3 - 6	23
shmmy4	manhattan	3	12	2	2	18	2	2	2	2	3 - 6	23
shmmy4	manhattan	5	7	2	3	16	2	2	2	2	3 - 6	23
shmmy4	manhattan	8	11	2	2	21	2	2	2	2	3 - 6	23
shmmy4	chebyshev	2	11	22	2	18	2	2	2	2	3 - 6	23
shmmy4	chebyshev	3	12	2	2	18	2	2	2	2	3 - 6	23
shmmy4	chebyshev	5	7	2	3	17	2	2	2	2	3 - 6	23
shmmy4	chebyshev	8	11	2	2	22	2	2	2	2	3 - 6	23
shmmy4	minkowski	2	11	22	2	18	2	2	2	2	3 - 6	23
shmmy4	minkowski	3	12	2	2	18	2	2	2	2	3 - 6	23
shmmy4	minkowski	5	7	2	3	17	2	2	2	2	3 - 6	23
shmmy4	minkowski	8	11	2	2	22	2	2	2	2	3 - 6	23

Πίνακας 3.5: Πίνακας αποτελεσμάτων για τον 4ο χρήστη

File	Distance	Fuzzifier	VMPC	FS	XB	VPCAES	VT	VPC	VPE	VK	Sessions by inspection	Candidate Session Id
shmmy5	euclidean	2	2	50	2	2	1	49	49	2	4 - 15	52
shmmy5	euclidean	3	2	30	2	2	2	3	2	2	4 - 15	52
shmmy5	euclidean	5	2	3	2	2	3	2	2	2	4 - 15	52
shmmy5	euclidean	8	2	51	2	2	2	2	2	2	4 - 15	52
shmmy5	manhattan	2	2	50	2	2	2	51	49	2	4 - 15	52
shmmy5	manhattan	3	2	30	2	2	2	3	2	2	4 - 15	52
shmmy5	manhattan	5	2	3	2	2	3	2	2	2	4 - 15	52
shmmy5	manhattan	8	2	51	2	2	2	2	2	2	4 - 15	52
shmmy5	chebyshev	2	2	50	2	2	2	49	49	2	4 - 15	52
shmmy5	chebyshev	3	2	30	2	2	2	3	2	2	4 - 15	52
shmmy5	chebyshev	5	2	3	2	2	3	2	2	2	4 - 15	52
shmmy5	chebyshev	8	2	51	2	2	2	2	2	2	4 - 15	52
shmmy5	minkowski	2	2	50	2	2	1	49	49	2	4 - 15	52
shmmy5	minkowski	3	2	30	2	2	2	3	2	2	4 - 15	52
shmmy5	minkowski	5	2	3	2	2	3	2	2	2	4 - 15	52
shmmy5	minkowski	8	2	51	2	2	2	2	2	2	4 - 15	52

Πίνακας 3.6: Πίνακας αποτελεσμάτων για τον 5ο χρήστη

File	Distance	Fuzzifier	VMPC	FS	XB	VPCAES	VT	VPC	VPE	VK	Sessions by inspection	Candidate Session Id
shmmy6	euclidean	2	4	2	2	5	2	2	2	2	3 - 4	8
shmmy6	euclidean	3	7	2	2	4	2	2	2	2	3 - 4	8
shmmy6	euclidean	5	4	2	2	6	2	2	2	2	3 - 4	8
shmmy6	euclidean	8	6	2	2	7	2	2	2	2	3 - 4	8
shmmy6	manhattan	2	4	2	2	5	2	2	2	2	3 - 4	8
shmmy6	manhattan	3	7	2	2	4	2	2	2	2	3 - 4	8
shmmy6	manhattan	5	4	2	2	6	2	2	2	2	3 - 4	8
shmmy6	manhattan	8	6	2	2	7	2	2	2	2	3 - 4	8
shmmy6	chebyshev	2	4	2	2	5	2	2	2	2	3 - 4	8
shmmy6	chebyshev	3	7	2	2	4	2	2	2	2	3 - 4	8
shmmy6	chebyshev	5	4	2	2	6	2	2	2	2	3 - 4	8
shmmy6	chebyshev	8	6	2	2	7	2	2	2	2	3 - 4	8
shmmy6	minkowski	2	4	2	2	5	2	2	2	2	3 - 4	8
shmmy6	minkowski	3	7	2	2	4	2	2	2	2	3 - 4	8
shmmy6	minkowski	5	4	2	2	6	2	2	2	2	3 - 4	8
shmmy6	minkowski	8	6	2	2	7	2	2	2	2	3 - 4	8

Πίνακας 3.7: Πίνακας αποτελεσμάτων για τον 6ο χρήστη

File	Distance	Fuzzifier	VMPC	FS	XB	VPCAES	VT	VPC	VPE	VK	Sessions by inspection	Candidate Session Id
shmmy7	euclidean	2	6	20	2	14	2	4	2	2	7 - 11	28
shmmy7	euclidean	3	13	2	5	12	2	2	2	2	7 - 11	28
shmmy7	euclidean	5	6	2	2	20	2	2	2	2	7 - 11	28
shmmy7	euclidean	8	6	2	2	20	2	2	2	2	7 - 11	28
shmmy7	manhattan	2	6	20	2	14	2	4	2	2	7 - 11	28
shmmy7	manhattan	3	13	2	5	12	2	2	2	2	7 - 11	28
shmmy7	manhattan	5	11	2	2	19	2	2	2	2	7 - 11	28
shmmy7	manhattan	8	7	2	2	20	2	2	2	2	7 - 11	28
shmmy7	chebyshev	2	6	27	2	14	2	4	2	2	7 - 11	28
shmmy7	chebyshev	3	13	2	5	24	2	2	2	2	7 - 11	28
shmmy7	chebyshev	5	10	2	1	20	2	2	2	2	7 - 11	28
shmmy7	chebyshev	8	6	2	2	20	2	2	2	2	7 - 11	28
shmmy7	minkowski	2	6	20	2	14	2	4	2	2	7 - 11	28
shmmy7	minkowski	3	13	2	5	12	2	2	2	2	7 - 11	28
shmmy7	minkowski	5	6	2	2	20	2	2	2	2	7 - 11	28
shmmy7	minkowski	8	6	2	2	20	2	2	2	2	7 - 11	28

Πίνακας 3.8: Πίνακας αποτελεσμάτων για τον 7ο χρήστη

File	Distance	Fuzzifier	VMPC	FS	XB	VPCAES	VT	VPC	VPE	VK	Sessions by inspection	Candidate Session Id
shmmy8	euclidean	2	6	41	2	38	2	35	2	2	6 - 11	42
shmmy8	euclidean	3	16	35	5	38	2	2	2	2	6 - 11	42
shmmy8	euclidean	5	16	2	3	38	2	2	2	2	6 - 11	42
shmmy8	euclidean	8	8	41	3	35	2	2	2	2	6 - 11	42
shmmy8	manhattan	2	6	41	2	40	2	35	2	2	6 - 11	42
shmmy8	manhattan	3	16	35	5	38	2	2	2	2	6 - 11	42
shmmy8	manhattan	5	16	2	3	38	2	2	2	2	6 - 11	42
shmmy8	manhattan	8	8	41	3	35	2	2	2	2	6 - 11	42
shmmy8	chebyshev	2	6	41	2	38	2	35	2	2	6 - 11	42
shmmy8	chebyshev	3	16	35	5	41	2	2	2	2	6 - 11	42
shmmy8	chebyshev	5	16	2	3	38	2	2	2	2	6 - 11	42
shmmy8	chebyshev	8	8	41	3	35	2	2	2	2	6 - 11	42
shmmy8	minkowski	2	6	41	2	38	2	35	2	2	6 - 11	42
shmmy8	minkowski	3	16	35	5	38	2	2	2	2	6 - 11	42
shmmy8	minkowski	5	16	2	3	38	2	2	2	2	6 - 11	42
shmmy8	minkowski	8	8	41	3	35	2	2	2	2	6 - 11	42

Πίνακας 3.9: Πίνακας αποτελεσμάτων για τον 8ο χρήστη

File	Distance	Fuzzifier	VMPC	FS	XB	VPCAES	VT	VPC	VPE	VK	Sessions by inspection	Candidate Session Id
shmmy9	euclidean	2	2	16	3	15	3	16	11	3	10 - 15	19
shmmy9	euclidean	3	2	16	4	16	3	2	2	3	10 - 15	19
shmmy9	euclidean	5	2	3	4	18	2	2	2	3	10 - 15	19
shmmy9	euclidean	8	2	3	4	18	2	2	2	2	10 - 15	19
shmmy9	manhattan	2	2	16	3	15	3	16	16	3	10 - 15	19
shmmy9	manhattan	3	2	16	4	16	3	2	2	3	10 - 15	19
shmmy9	manhattan	5	2	3	4	18	2	2	2	3	10 - 15	19
shmmy9	manhattan	8	2	3	4	18	2	2	2	2	10 - 15	19
shmmy9	chebyshev	2	2	16	3	15	3	16	11	3	10 - 15	19
shmmy9	chebyshev	3	2	16	4	16	3	2	2	3	10 - 15	19
shmmy9	chebyshev	5	2	3	4	18	2	2	2	3	10 - 15	19
shmmy9	chebyshev	8	2	3	4	18	2	2	2	2	10 - 15	19
shmmy9	minkowski	2	2	16	3	15	3	16	11	3	10 - 15	19
shmmy9	minkowski	3	2	16	4	16	3	2	2	3	10 - 15	19
shmmy9	minkowski	5	2	3	4	18	2	2	2	3	10 - 15	19
shmmy9	minkowski	8	2	3	4	18	2	2	2	2	10 - 15	19

Πίνακας 3.10: Πίνακας αποτελεσμάτων για τον 9ο χρήστη

File	Distance	Fuzzifier	VMPC	FS	XB	VPCAES	VT	VPC	VPE	VK	Sessions by inspection	Candidate Session Id
shmmy10	euclidean	2	2	9	7	8	3	7	7	4	6 - 7	10
shmmy10	euclidean	3	2	6	4	6	3	6	6	4	6 - 7	10
shmmy10	euclidean	5	2	4	4	6	2	3	2	2	6 - 7	10
shmmy10	euclidean	8	2	3	3	9	2	2	2	2	6 - 7	10
shmmy10	manhattan	2	2	6	7	5	3	7	7	4	6 - 7	10
shmmy10	manhattan	3	2	6	4	6	3	6	6	4	6 - 7	10
shmmy10	manhattan	5	2	4	4	6	2	3	2	2	6 - 7	10
shmmy10	manhattan	8	2	3	3	9	2	2	2	2	6 - 7	10
shmmy10	chebyshev	2	3	7	2	7	2	8	8	2	6 - 7	10
shmmy10	chebyshev	3	3	6	7	6	2	6	6	2	6 - 7	10
shmmy10	chebyshev	5	3	4	8	6	2	2	2	2	6 - 7	10
shmmy10	chebyshev	8	3	4	2	9	2	2	2	2	6 - 7	10
shmmy10	minkowski	2	2	9	7	8	3	7	7	4	6 - 7	10
shmmy10	minkowski	3	2	6	4	6	3	6	6	4	6 - 7	10
shmmy10	minkowski	5	2	4	4	6	2	3	2	2	6 - 7	10
shmmy10	minkowski	8	2	3	3	9	2	2	2	2	6 - 7	10

Πίνακας 3.11: Πίνακας αποτελεσμάτων για τον 10ο χρήστη

File	Distance	Fuzzifier	VMPC	FS	XB	VPCAES	VT	VPC	VPE	VK	Sessions by inspection	Candidate Session Id
shmmy11	euclidean	2	9	8	3	10	2	2	2	2	4 - 5	31
shmmy11	euclidean	3	7	28	14	21	2	2	2	2	4 - 5	31
shmmy11	euclidean	5	3	2	6	30	2	2	2	2	4 - 5	31
shmmy11	euclidean	8	7	2	2	17	2	2	2	2	4 - 5	31
shmmy11	manhattan	2	9	8	3	19	2	30	2	2	4 - 5	31
shmmy11	manhattan	3	7	29	8	30	2	2	2	2	4 - 5	31
shmmy11	manhattan	5	6	2	5	30	2	2	2	2	4 - 5	31
shmmy11	manhattan	8	6	15	2	17	2	2	2	2	4 - 5	31
shmmy11	chebyshev	2	9	8	3	10	2	30	2	2	4 - 5	31
shmmy11	chebyshev	3	7	29	3	19	2	2	2	2	4 - 5	31
shmmy11	chebyshev	5	3	2	6	30	2	2	2	2	4 - 5	31
shmmy11	chebyshev	8	6	2	6	17	2	2	2	2	4 - 5	31
shmmy11	minkowski	2	9	8	3	10	2	2	2	2	4 - 5	31
shmmy11	minkowski	3	7	28	14	21	2	2	2	2	4 - 5	31
shmmy11	minkowski	5	3	2	6	30	2	2	2	2	4 - 5	31
shmmy11	minkowski	8	7	2	2	17	2	2	2	2	4 - 5	31

Πίνακας 3.12: Πίνακας αποτελεσμάτων για τον 11ο χρήστη

File	Distance	Fuzzifier	VMPC	FS	XB	VPCAES	VT	VPC	VPE	VK	Sessions by inspection	Candidate Session Id
shmmy12	euclidean	2	3	4	6	4	2	2	2	2	6 - 8	9
shmmy12	euclidean	3	3	4	8	4	2	2	2	2	6 - 8	9
shmmy12	euclidean	5	8	5	6	8	2	2	2	2	6 - 8	9
shmmy12	euclidean	8	8	4	5	6	2	2	2	2	6 - 8	9
shmmy12	manhattan	2	3	4	6	4	2	2	2	2	6 - 8	9
shmmy12	manhattan	3	3	4	8	4	2	2	2	2	6 - 8	9
shmmy12	manhattan	5	8	5	6	8	2	2	2	2	6 - 8	9
shmmy12	manhattan	8	8	4	6	6	2	2	2	2	6 - 8	9
shmmy12	chebyshev	2	3	4	6	4	2	2	2	2	6 - 8	9
shmmy12	chebyshev	3	3	4	8	4	2	2	2	2	6 - 8	9
shmmy12	chebyshev	5	8	5	6	8	2	2	2	2	6 - 8	9
shmmy12	chebyshev	8	8	4	5	6	2	2	2	2	6 - 8	9
shmmy12	minkowski	2	3	4	6	4	2	2	2	2	6 - 8	9
shmmy12	minkowski	3	3	4	8	4	2	2	2	2	6 - 8	9
shmmy12	minkowski	5	8	5	6	8	2	2	2	2	6 - 8	9
shmmy12	minkowski	8	8	4	5	6	2	2	2	2	6 - 8	9

Πίνακας 3.13: Πίνακας αποτελεσμάτων για τον 12ο χρήστη

File	Distance	Fuzzifier	VMPC	FS	XB	VPCAES	VT	VPC	VPE	VK	Sessions by inspection	Candidate Session Id
shmmy13	euclidean	2	19	3	2	29	2	2	2	2	5 - 9	100
shmmy13	euclidean	3	15	3	2	30	2	2	2	2	5 - 9	100
shmmy13	euclidean	5	28	2	2	99	2	2	2	2	5 - 9	100
shmmy13	euclidean	8	11	99	2	99	2	2	2	2	5 - 9	100
shmmy13	manhattan	2	15	3	2	28	2	2	2	2	5 - 9	100
shmmy13	manhattan	3	15	3	2	27	2	2	2	2	5 - 9	100
shmmy13	manhattan	5	17	2	2	99	2	2	2	2	5 - 9	100
shmmy13	manhattan	8	11	99	2	99	2	2	2	2	5 - 9	100
shmmy13	chebyshev	2	19	3	2	29	2	2	2	2	5 - 9	100
shmmy13	chebyshev	3	11	3	2	30	2	2	2	2	5 - 9	100
shmmy13	chebyshev	5	21	2	2	99	2	2	2	2	5 - 9	100
shmmy13	chebyshev	8	11	99	2	99	2	2	2	2	5 - 9	100
shmmy13	minkowski	2	19	3	2	29	2	2	2	2	5 - 9	100
shmmy13	minkowski	3	15	3	2	30	2	2	2	2	5 - 9	100
shmmy13	minkowski	5	28	2	2	99	2	2	2	2	5 - 9	100
shmmy13	minkowski	8	11	99	2	99	2	2	2	2	5 - 9	100

Πίνακας 3.14: Πίνακας αποτελεσμάτων για τον 13ο χρήστη

File	Distance	Fuzzifier	VMPC	FS	XB	VPCAES	VT	VPC	VPE	VK	Sessions by inspection	Candidate Session Id
shmmy14	euclidean	2	3	17	4	16	2	17	17	2	8 - 15	20
shmmy14	euclidean	3	3	16	4	17	2	19	2	2	8 - 15	20
shmmy14	euclidean	5	3	19	4	19	2	2	2	2	8 - 15	20
shmmy14	euclidean	8	2	19	14	19	2	2	2	2	8 - 15	20
shmmy14	manhattan	2	3	17	4	16	2	17	17	2	8 - 15	20
shmmy14	manhattan	3	3	17	4	19	2	17	2	2	8 - 15	20
shmmy14	manhattan	5	3	19	4	19	2	2	2	2	8 - 15	20
shmmy14	manhattan	8	2	19	14	19	2	2	2	2	8 - 15	20
shmmy14	chebyshev	2	3	17	4	16	2	17	17	2	8 - 15	20
shmmy14	chebyshev	3	3	16	4	17	2	19	2	2	8 - 15	20
shmmy14	chebyshev	5	3	19	4	19	2	2	2	2	8 - 15	20
shmmy14	chebyshev	8	2	19	14	19	2	2	2	2	8 - 15	20
shmmy14	minkowski	2	3	17	4	16	2	17	17	2	8 - 15	20
shmmy14	minkowski	3	3	16	4	17	2	19	2	2	8 - 15	20
shmmy14	minkowski	5	3	19	4	19	2	2	2	2	8 - 15	20
shmmy14	minkowski	8	2	19	14	19	2	2	2	2	8 - 15	20

Πίνακας 3.15: Πίνακας αποτελεσμάτων για τον 14ο χρήστη

File	Distance	Fuzzifier	VMPC	FS	XB	VPCAES	VT	VPC	VPE	VK	Sessions by inspection	Candidate Session Id
shmmy15	euclidean	2	2	35	35	57	2	35	35	2	17 - 30	58
shmmy15	euclidean	3	3	43	31	57	2	39	39	2	17 - 30	58
shmmy15	euclidean	5	4	44	13	57	2	43	2	2	17 - 30	58
shmmy15	euclidean	8	3	52	13	57	2	2	2	2	17 - 30	58
shmmy15	manhattan	2	2	43	5	57	2	32	32	2	17 - 30	58
shmmy15	manhattan	3	3	43	32	57	2	32	40	2	17 - 30	58
shmmy15	manhattan	5	4	44	13	57	2	45	2	2	17 - 30	58
shmmy15	manhattan	8	3	52	13	57	2	2	2	2	17 - 30	58
shmmy15	chebyshev	2	2	35	35	57	2	35	35	2	17 - 30	58
shmmy15	chebyshev	3	3	39	5	57	2	39	39	2	17 - 30	58
shmmy15	chebyshev	5	3	45	25	57	2	45	2	2	17 - 30	58
shmmy15	chebyshev	8	3	55	24	57	2	2	2	2	17 - 30	58
shmmy15	minkowski	2	2	35	35	57	2	35	35	2	17 - 30	58
shmmy15	minkowski	3	3	43	31	57	2	39	39	2	17 - 30	58
shmmy15	minkowski	5	4	44	13	57	2	43	2	2	17 - 30	58
shmmy15	minkowski	8	3	52	13	57	2	2	2	2	17 - 30	58

Πίνακας 3.16: Πίνακας αποτελεσμάτων για τον 15ο χρήστη

File	Distance	Fuzzifier	VMPC	FS	XB	VPCAES	VT	VPC	VPE	VK	Sessions by inspection	Candidate Session Id
shmmy16	euclidean	2	2	6	3	35	3	4	4	3	1	44
shmmy16	euclidean	3	11	43	25	43	3	2	2	3	1	44
shmmy16	euclidean	5	2	43	25	43	2	4	2	3	1	44
shmmy16	euclidean	8	2	43	25	43	2	4	2	2	1	44
shmmy16	manhattan	2	2	10	3	40	3	4	4	3	1	44
shmmy16	manhattan	3	2	43	23	43	3	2	2	3	1	44
shmmy16	manhattan	5	7	43	25	38	2	4	2	3	1	44
shmmy16	manhattan	8	2	43	25	43	2	4	2	2	1	44
shmmy16	chebyshev	2	2	4	3	35	3	4	4	3	1	44
shmmy16	chebyshev	3	9	43	32	43	3	2	2	3	1	44
shmmy16	chebyshev	5	9	43	32	43	2	4	2	3	1	44
shmmy16	chebyshev	8	2	43	32	43	2	4	2	2	1	44
shmmy16	minkowski	2	2	6	3	35	3	4	4	3	1	44
shmmy16	minkowski	3	11	43	25	43	3	2	2	3	1	44
shmmy16	minkowski	5	2	43	25	43	2	4	2	3	1	44
shmmy16	minkowski	8	2	43	25	43	2	4	2	2	1	44

Πίνακας 3.17: Πίνακας αποτελεσμάτων για τον 16ο χρήστη

File	Distance	Fuzzifier	VMPC	FS	XB	VPCAES	VT	VPC	VPE	VK	Sessions by inspection	Candidate Session Id
shmmy17	euclidean	2	2	2	2	2	2	2	2	2	2 - 3	3
shmmy17	euclidean	3	2	2	2	2	2	2	2	2	2 - 3	3
shmmy17	euclidean	5	2	2	2	2	2	2	2	2	2 - 3	3
shmmy17	euclidean	8	2	2	2	2	2	2	2	2	2 - 3	3
shmmy17	manhattan	2	2	2	2	2	2	2	2	2	2 - 3	3
shmmy17	manhattan	3	2	2	2	2	2	2	2	2	2 - 3	3
shmmy17	manhattan	5	2	2	2	2	2	2	2	2	2 - 3	3
shmmy17	manhattan	8	2	2	2	2	2	2	2	2	2 - 3	3
shmmy17	chebyshev	2	2	2	2	2	2	2	2	2	2 - 3	3
shmmy17	chebyshev	3	2	2	2	2	2	2	2	2	2 - 3	3
shmmy17	chebyshev	5	2	2	2	2	2	2	2	2	2 - 3	3
shmmy17	chebyshev	8	2	2	2	2	2	2	2	2	2 - 3	3
shmmy17	minkowski	2	2	2	2	2	2	2	2	2	2 - 3	3
shmmy17	minkowski	3	2	2	2	2	2	2	2	2	2 - 3	3
shmmy17	minkowski	5	2	2	2	2	2	2	2	2	2 - 3	3
shmmy17	minkowski	8	2	2	2	2	2	2	2	2	2 - 3	3

Πίνακας 3.18: Πίνακας αποτελεσμάτων για τον 17ο χρήστη

File	Distance	Fuzzifier	VPMC	FS	XB	VPCAES	VT	VPC	VPE	VK	Sessions by inspection	Candidate Session Id
shmmy18	euclidean	2	9	20	4	51	4	51	51	4	8 - 23	58
shmmy18	euclidean	3	10	49	4	51	4	49	2	4	8 - 23	58
shmmy18	euclidean	5	9	5	4	31	2	2	2	2	8 - 23	58
shmmy18	euclidean	8	9	5	4	31	2	2	2	2	8 - 23	58
shmmy18	manhattan	2	9	20	4	55	4	55	56	4	8 - 23	58
shmmy18	manhattan	3	10	50	4	55	4	51	2	4	8 - 23	58
shmmy18	manhattan	5	9	5	4	31	2	2	2	2	8 - 23	58
shmmy18	manhattan	8	3	57	6	57	2	2	2	2	8 - 23	58
shmmy18	chebyshev	2	9	20	4	28	4	55	55	4	8 - 23	58
shmmy18	chebyshev	3	10	39	4	52	4	49	2	4	8 - 23	58
shmmy18	chebyshev	5	9	5	4	32	2	2	2	2	8 - 23	58
shmmy18	chebyshev	8	3	57	6	57	2	2	2	2	8 - 23	58
shmmy18	minkowski	2	9	20	4	51	4	51	51	4	8 - 23	58
shmmy18	minkowski	3	10	49	4	51	4	49	2	4	8 - 23	58
shmmy18	minkowski	5	9	5	4	31	2	2	2	2	8 - 23	58
shmmy18	minkowski	8	3	57	6	57	2	2	2	2	8 - 23	58

Πίνακας 3.19: Πίνακας αποτελεσμάτων για τον 18ο χρήστη

File	Distance	Fuzzifier	VMPC	FS	XB	VPCAES	VT	VPC	VPE	VK	Sessions by inspection	Candidate Session Id
shmmy19	euclidean	2	2	12	4	12	3	5	4	4	5 - 6	16
shmmy19	euclidean	3	14	15	4	12	3	3	2	4	5 - 6	16
shmmy19	euclidean	5	15	3	3	13	3	2	2	3	5 - 6	16
shmmy19	euclidean	8	9	15	3	9	3	2	2	3	5 - 6	16
shmmy19	manhattan	2	2	8	4	11	3	5	4	4	5 - 6	16
shmmy19	manhattan	3	14	15	4	15	3	3	2	4	5 - 6	16
shmmy19	manhattan	5	9	3	3	15	3	2	2	3	5 - 6	16
shmmy19	manhattan	8	7	14	3	9	3	2	2	3	5 - 6	16
shmmy19	chebyshev	2	2	12	4	12	3	5	4	4	5 - 6	16
shmmy19	chebyshev	3	14	15	4	12	3	3	2	4	5 - 6	16
shmmy19	chebyshev	5	12	3	3	13	3	2	2	3	5 - 6	16
shmmy19	chebyshev	8	9	15	3	9	3	2	2	3	5 - 6	16
shmmy19	minkowski	2	2	12	4	12	3	5	4	4	5 - 6	16
shmmy19	minkowski	3	14	15	4	12	3	3	2	4	5 - 6	16
shmmy19	minkowski	5	15	3	3	13	3	2	2	3	5 - 6	16
shmmy19	minkowski	8	9	15	3	9	3	2	2	3	5 - 6	16

Πίνακας 3.20: Πίνακας αποτελεσμάτων για τον 19ο χρήστη

File	Distance	Fuzzifier	VMPC	FS	XB	VPCAES	VT	VPC	VPE	VK	Sessions by inspection	Candidate Session Id
shmmy20	euclidean	2	2	11	3	39	3	18	18	3	5 - 6	48
shmmy20	euclidean	3	2	12	6	37	3	47	47	3	5 - 6	48
shmmy20	euclidean	5	2	47	3	46	3	47	47	3	5 - 6	48
shmmy20	euclidean	8	2	47	3	46	3	47	47	3	5 - 6	48
shmmy20	manhattan	2	2	14	3	37	3	24	24	3	5 - 6	48
shmmy20	manhattan	3	2	14	3	37	3	47	47	3	5 - 6	48
shmmy20	manhattan	5	2	47	32	46	3	47	47	3	5 - 6	48
shmmy20	manhattan	8	2	47	3	46	3	47	47	3	5 - 6	48
shmmy20	chebyshev	2	2	13	3	35	3	18	18	3	5 - 6	48
shmmy20	chebyshev	3	2	15	3	36	3	47	47	3	5 - 6	48
shmmy20	chebyshev	5	2	47	3	45	3	47	47	3	5 - 6	48
shmmy20	chebyshev	8	2	47	3	45	2	47	47	2	5 - 6	48
shmmy20	minkowski	2	2	11	3	39	3	18	18	3	5 - 6	48
shmmy20	minkowski	3	2	12	6	37	3	47	47	3	5 - 6	48
shmmy20	minkowski	5	2	47	3	46	3	47	47	3	5 - 6	48
shmmy20	minkowski	8	2	47	3	46	3	47	47	3	5 - 6	48

Πίνακας 3.21: Πίνακας αποτελεσμάτων για τον 20ο χρήστη

Κεφάλαιο 4

Ανακάλυψη μοτίβων

4.1 Εισαγωγή

Η ανακάλυψη μοτίβων (pattern discovery) συνδυάζει μεθόδους και αλγορίθμους από αρκετές επιστημονικές κατηγορίες όπως για παράδειγμα τη στατιστική, την εξόρυξη δεδομένων και τη μηχανική μάθηση. Εφαρμόζεται στα προεπεξεργασμένα δεδομένα προκειμένου να βρεθούν τα "μονοπάτια" που ακολουθεί ένας επισκέπτης κατά την πλοήγησή του σε μία ιστοσελίδα. Οι αλγόριθμοι που χρησιμοποιούνται εξαρτώνται εξ ολοκλήρου από το άτομο που θέλει να κάνει την ανάλυση. Σε αυτό το κεφάλαιο της διπλωματικής θα περιγράψουμε κάποιες από τις τεχνικές που χρησιμοποιούνται για την ανακάλυψη μοτίβων και πιο συγκεκριμένα θα εστιάσουμε σε ομάδες ιστοσελίδων που δύνανται να προσπελαστούν συχνά μαζί, όπως αυτό αποτυπώνεται στα αρχεία καταγραφής του εξυπηρετητή. Στη συνέχεια, παρουσιάζονται κάποιες από τις πιο γνωστές τεχνικές που χρησιμοποιούνται για την ανακάλυψη μοτίβων όπως αυτές παρουσιάζονται στο [50].

4.1.1 Συσταδοποίηση

Οι μορφές της συσταδοποίησης (clustering) που χρησιμοποιούνται για την ανακάλυψη μοτίβων στον παγκόσμιο ιστό μπορούν να χωριστούν σε δύο κατηγορίες: σε αυτές που συσταδοποιούν τους χρήστες και σε αυτές που συσταδοποιούν τις σελίδες. Η συσταδοποίηση των σελίδων γίνεται ανάλογα με το περιεχόμενό τους και αυτές με παρόμοιο περιεχόμενο, ανήκουν στην ίδια συστάδα. Η συσταδοποίηση των χρηστών γίνεται ανάλογα με το πόσο όμοια είναι η συμπεριφορά τους κατά την πλοήγηση της ιστοσελίδας. Η συσταδοποίηση μπορεί να γίνει ανάλογα με ένα προκαθορισμένο μοντέλο, το οποίο καθορίζεται από πριν ή με τη χρήση μετρικών απόστασης μεταξύ των ζευγών των αντικειμένων των δεδομένων. Όταν χρησιμοποιείται κάποιο μοντέλο για τη συσταδοποίηση ακολουθούνται τα εξής βήματα:

1. Παρατηρούνται τα χαρακτηριστικά κάποιων αντικειμένων/δεδομένων.
2. Κάθε αντικείμενο περιέχεται σε μία από τις συστάδες αλλά δεν γνωρίζουμε ποια είναι αυτή.
3. Η πιθανότητα του συγκεκριμένου αντικειμένου να ανήκει σε μία συστάδα είναι μεγαλύτερη σε κάποιες συστάδες από κάποιες άλλες.
4. Μέσα σε μία συστάδα, τα χαρακτηριστικά των αντικειμένων παράγονται από την ίδια κατανομή, της οποίας οι παράμετροι είναι ελεύθερες.
5. Το μοντέλο συσχετίζει τις παρατηρήσεις που έγιναν στο πρώτο βήμα σε σχέση με τις συμμετοχή των αντικειμένων σε κάθε συστάδα και τις παραμέτρους.

6. Οι τιμές των παραμέτρων βρίσκονται μεγιστοποιώντας την πιθανοφάνεια (ή τον λογάριθμό της) των παρατηρήσεων.

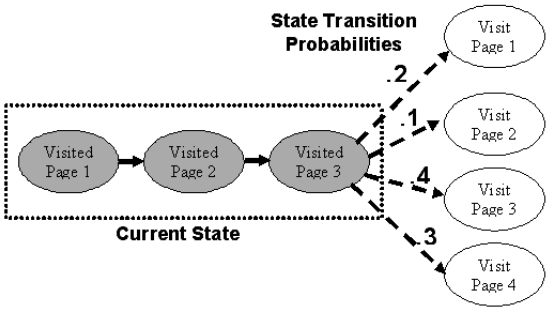
7. Έχοντας υπολογίσει τις παραμέτρους, υπολογίζουμε την πιθανότητα συμμετοχής σε σχέση με τις παρατηρήσεις.

Για τη μεγιστοποίηση της πιθανοφάνειας χρησιμοποιούνται αλγόριθμοι όπως είναι ο Expectation Maximization[51].

Όταν έχουμε καθορίσει μια μετρική απόστασης, τα αντικείμενα που βρίσκονται κοντά μεταξύ τους ανήκουν στην ίδια συστάδα. Οι πιο διαδεδομένοι αλγόριθμοι σε αυτή την κατηγορία είναι η συσταδοποίηση που κάθε αντικείμενο ανήκει σε μία μόνο συστάδα (partitional clustering) ή η ιεραρχική συσταδοποίηση κατά την οποία "χτίζεται" μια ιεραρχία από συστάδες (hierarchical clustering). Αλγόριθμοι συσταδοποίησης για την ανακάλυψη μοτίβων σε δεδομένα καταγραφής εξυπηρετητών διαδικτύου έχουν χρησιμοποιηθεί με πολλούς τρόπους. Χαρακτηριστικά στο [52], οι συγγραφείς συσταδοποιούν συναλλαγές διαδικτύου χρησιμοποιώντας hierarchical clustering αλγορίθμους, με σκοπό τη μεγιστοποίηση μια αντικειμενικής συνάρτησης για την καλύτερη συσταδοποίηση των παρόμοιων συναλλαγών. Επίσης, έχει ληφθεί υπόψιν ο χρόνος και η τοποθεσία των χρηστών [53], προκειμένου να παραχθούν συστάδες που δείχνουν χρήστες με παρόμοια συμπεριφορά πλοήγησης σε μια χρονική περίοδο, διαφοροποιώντας την προτεραιότητα που δίνεται σε μία σελίδα ή στον χρόνο επίσκεψης. Τέλος, έχουν χρησιμοποιηθεί οι αλγόριθμοι Adaptive Resonance Theory1 Neural Network (ART1 NN) [54], Ant Colony Optimization[55], Self-Organizing Feature Maps (SOM) και K-Medoids [56] για την ανακάλυψη παρόμοιων συμπεριφορών χρηστών. Η συσταδοποίηση με βάση κάποιο μοντέλο, έχει αποδειχθεί αποτελεσματική για τη συσταδοποίηση κειμένων που έχουν υψηλή διαστατικότητα ενώ η ιεραρχική συσταδοποίηση θεωρείται ακατάλληλη για τον Παγκόσμιο Ιστό, λόγω της πολύ μεγάλης ποσότητας πληροφορίας που περιέχεται σε αυτόν [57]. Η συσταδοποίηση με μετρικές απόστασης εξαρτάται πολύ από τον τύπο των δεδομένων, κάτι που απαιτεί ειδικές γνώσεις από αυτόν που θα εφαρμόσει την τεχνική, επομένως δεν χρησιμοποιείται τόσο συχνά[58]. Τέλος, η συσταδοποίηση σαν τεχνική μπορεί να χρησιμοποιηθεί για να διαχωρίσουμε τα δεδομένα σε ομοιογενείς ομάδες, αλλά από μόνη της δεν μπορεί να μας βοηθήσει στην πρόβλεψη κάποιας σελίδας που θα ζητήσει ένας χρήστης.

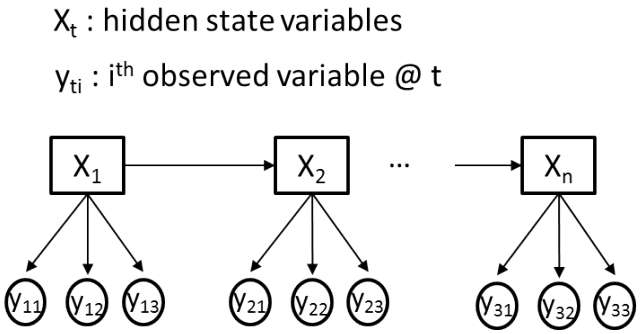
4.1.2 Εξόρυξη ακολουθιακών μοτίβων

Τα ακολουθιακά μοτίβα (sequential patterns) είναι μία ένδειξη των αιτημάτων που κάνουν συχνά οι χρήστες, διατηρώντας όμως τη σειρά με την οποία έγιναν αυτά. Για τη μοντελοποίηση των μοτίβων πλοήγησης έχουν χρησιμοποιηθεί εκτενώς τα Μοντέλα Markov με τον ακόλουθο τρόπο: Κάθε προβολή μιας σελίδας ή αλλιώς ένα αίτημα μοντελοποιείται σαν μία κατάσταση και η πιθανότητα μετάβασης μεταξύ δύο καταστάσεων, αντιπροσωπεύει την πιθανοφάνεια με την οποία ο χρήστης θα μεταβεί μεταξύ των δύο αυτών σελίδων.



Σχήμα 4.1: Αναπαράσταση σελίδων σαν Μαρκοβιανή Διαδικασία [5]

Έχουν προταθεί διάφορα μοντέλα ανάλογα με τις παραμέτρους των μοντέλων Markov και συνοπτικά μπορούμε να αναφέρουμε τα all-*k*th-order, selective, relational, hidden μοντέλα Markov. Από τα προαναφερθέντα, ιδιαίτερο ενδιαφέρον παρουσιάζουν τα κρυφά μαρκοβιανά μοντέλα (Hidden Markov Models - HMM), εξαιτίας της ευρείας χρήσης τους στον τομέα της μηχανικής μάθησης.



Σχήμα 4.2: Hidden Markov Model

Ένα HMM, αρχικά περιέχει ένα πεπερασμένο αριθμό καταστάσεων. Οι μεταβάσεις μεταξύ των καταστάσεων γίνονται με βάση κάποια πιθανότητα, που είναι συγκεκριμένη για κάθε κατάσταση. Από κάθε κατάσταση, μπορεί να εξαχθεί ένα αποτέλεσμα ή μία παρατήρηση, η οποία υπολογίζεται με βάση μια ξεχωριστή κατανομή πιθανότητας που σχετίζεται με αυτή την κατάσταση. Οι καταστάσεις είναι κρυφές και μόνο τα αποτελέσματα είναι ορατά σε έναν εξωτερικό παρατηρητή.

4.1.3 Μοντέλα ανάμιξης

Τα μοντέλα ανάμιξης (mixture models) είναι μια ακόμα περίπτωση ταξινομητών. Προκειμένου να βρεθεί ένα σωστό μοντέλο ταξινόμησης των δεδομένων γίνονται οι ακόλουθες παραδοχές. Κατ' αρχάς η συμπεριφορά του κάθε χρήστη στο σύνολο των δεδομένων θεωρείται ανεξάρτητη από τους υπόλοιπους χρήστες και κατά δεύτερον ότι η συμπεριφορά μπορεί να παραχθεί από ένα μοντέλο ανάμιξης k συνιστωσών. Σε ένα mixture model, ενδιαφερόμαστε για τον αριθμό των συνιστωσών, την κατανομή πιθανότητας που χρησιμοποιείται προκειμένου να χωριστούν οι χρήστες σε συστάδες και τις παραμέτρους κάθε συνιστώσας του μοντέλου. Τα mixture models έχουν χρησιμοποιηθεί προκειμένου να χωριστούν οι χρήστες ανάλογα με τη συμπεριφορά τους στον Παγκόσμιο Ιστό. Στην προτεινόμενη μέθοδο, μοντελοποιήθηκαν τα κοινά ενδιαφέροντα των χρηστών σαν κρυφοί παράγοντες και ανακαλύφθηκαν τμήματα "παρόμοιων" χρηστών, με βάση την κατανομή

των συνιστωσών ενός πεπερασμένου mixture model [59]. Με βάση αυτή τη μέθοδο, συσχετίστηκε πιθανοτικά η παρατηρούμενη συμπεριφορά πλοήγησης των χρηστών με τα ενδιαφέροντά τους που δεν έχουν παρατηρηθεί ακόμα. Άλλα μοντέλα που έχουν χρησιμοποιηθεί μαζί με τα mixture models περιλαμβάνουν κατανομές Pareto και Gauss.

4.1.4 Ταξινόμηση

Ως ταξινόμηση, ορίζουμε τη διαδικασία κατά την οποία ένα αντικείμενο ανατίθεται σε μία ή περισσότερες προκαθορισμένες κλάσεις. Κατά την επιβλεπόμενη μάθηση, αρχικά, ενδιαφερόμαστε να φτιάξουμε ένα μοντέλο το οποίο με βάση τα χαρακτηριστικά των δεδομένων, θα παράγει τις κλάσεις που μπορούν να κατηγοριοποιηθούν αυτά, χρησιμοποιώντας κάποιο σύνολο εκπαίδευσης, και στη συνέχεια θα κατηγοριοποιεί άγνωστα δεδομένα. Υπάρχουν πολλές τεχνικές ταξινόμησης, αλλά μερικές από τις πιο διάσημες και ευρέως χρησιμοποιούμενες σε δεδομένα του Παγκοσμίου Ιστού είναι τα δέντρα αποφάσεων (decision trees), ο απλός μπεϋζιανός ταξινομητής (naive bayesian classifier), ο k πλησιέστερων γειτόνων (k-nearest neighbours) και οι μηχανές υποστήριξης διανυσμάτων ή support vector machines (SVM)[60]. Τα δέντρα αποφάσεων είναι ο πιο εύκολος τρόπος κατανόησης μιας ταξινόμησης. Ο ταξινομητής Naive Bayes χρησιμοποιείται ευρέως σε αλγορίθμους στατιστικής μάθησης, επειδή έχει πολύ μικρό χρόνο εκπαίδευσης. Με αυτόν τον ταξινομητή έχουν χωριστεί επισκέπτες ιστοσελίδων με βάση τη διάρκεια των sessions τους, τις σελίδες που έχουν επισκεφθεί και την ποσότητα των ιστοσελίδων που ζητούν κατά τη διάρκεια μιας επίσκεψης στη σελίδα [61]. Ο ταξινομητής k-Nearest Neighbour (kNN) βασίζεται στην αρχή του ότι τα σημεία ενός συνόλου δεδομένων τα οποία έχουν παρόμοιες ιδιότητες θα έχουν εγγύτητα μεταξύ τους. Οι μηχανές υποστήριξης διανυσμάτων έχουν χρησιμοποιηθεί για να ταξινομήσουν τα αιτήματα του ιστορικού ενός χρήστη, αφού πρώτα έχουν προεπεξεργαστεί με βάση την τεχνική TFIDF[62].

4.1.5 Τεχνικές συλλογικής διήθησης

Οι τεχνικές συνεργατικής διήθησης (Collaborative Filtering - CF) είναι τεχνικές που χρησιμοποιούνται κυρίως για να προβλεφθούν οι προτιμήσεις κάθε χρήστη. Όταν σε μία βάση δεδομένων, συσσωρευθούν προτιμήσεις χρηστών, οι δείκτες ομοιότητας μπορούν να εντοπίσουν άτομα που έχουν παρόμοιες προτιμήσεις, με το χρήστη που πλοηγείται αυτή τη στιγμή στην ιστοσελίδα. Παρόλο που η συγκεκριμένη τεχνική είναι σχετικά εύκολη στην υλοποίηση, απαιτεί πολύ μεγάλο αριθμό δειγμάτων προκειμένου να δώσει συστάσεις, οι οποίες να έχουν νόημα. Αν δεν υπάρχει κοντινή πληροφορία σε προτιμήσεις, δίνονται συστάσεις που δεν έχουν νόημα ή είναι λάθος. Όπως είναι λογικό, όταν το μέγεθος της βάσης δεδομένων αυξάνεται, η παραγωγή των συστάσεων γίνεται χρονοβόρος λόγων των υπολογισμών που απαιτούνται. Επίσης στην παραγωγή των συστάσεων δεν λαμβάνονται υπόψιν αφενός το προφίλ του επισκέπτη και αφετέρου πληροφορίες σχετικά με την συμπεριφορά πλοήγησής του. Ένα ακόμα μειονέκτημα αυτής της μεθόδου είναι ότι αντιμετωπίζει “προβλήματα αραιότητας”. Αυτό προκύπτει από το γεγονός ότι οι χρήστες έχουν δείξει προτιμήσεις για πολύ λίγα αντικείμενα σε σχέση με το σύνολο των αντικειμένων που προσφέρει μια ιστοσελίδα και επομένως είναι δύσκολο να οριστεί μια “γειτονιά” προτιμήσεων ενός χρήστη με μεγάλη ακρίβεια, κάτι που οδηγεί σε ανούσιες συστάσεις [63]. Για τη λύση αυτών των προβλημάτων έχουν χρησιμοποιηθεί τεχνικές ταξινόμησης και πρόβλεψης όπως για παράδειγμα κανόνες συσχέτισης [64], ο ταξινομητής Bayes [65], ο αλγόριθμος k-Nearest Neighbor [66]αλλά

και τα SVMs[67].

4.2 Στατιστική Ανάλυση

4.2.1 Εισαγωγή

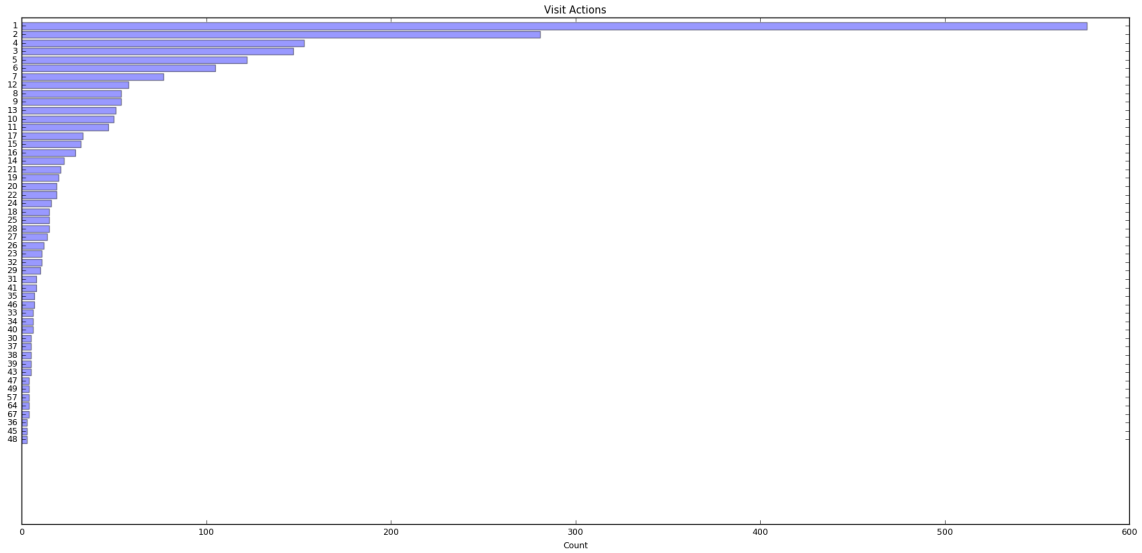
Σε ένα πρώτο βήμα, τα προεπεξεργασμένα δεδομένα αναλύονται στη βάση των τεχνικών της διερευνητικής ανάλυσης δεδομένων (Exploratory Data Analysis - EDA) [68]. Η χρήση της EDA επιτρέπει:

- Την εξέταση με λεπτομέρεια της μορφής των δεδομένων.
- Την εύρεση σχέσεων μεταξύ των γνωρισμάτων.
- Την αναγνώριση υποσυνόλων των παρατηρήσεων που έχουν μεγαλύτερο ενδιαφέρον.
- Την προσέγγιση των συσχετίσεων μεταξύ των δεδομένων, καθώς και των στόχων που περιμένουμε να λάβουμε από την ανάλυση.

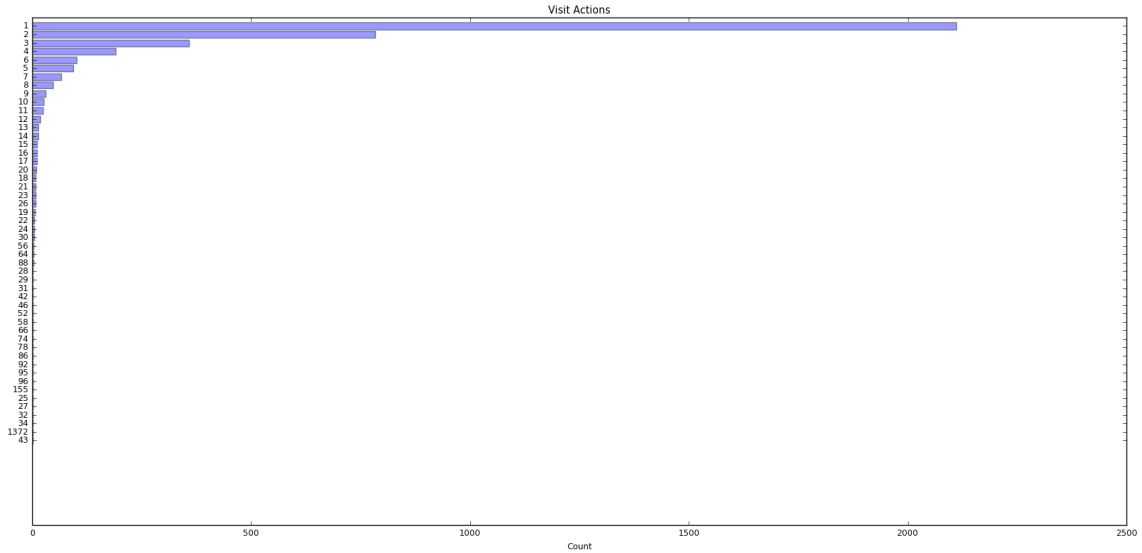
Σε αυτή την παράγραφο θα εξάγουμε απλά στατιστικά συμπεράσματα από αρχεία καταγραφής, τα οποία μπορούν να μας δώσουν χρήσιμα συμπεράσματα σχετικά με μία ιστοσελίδα. Θα χρησιμοποιηθεί ως σύνολο δεδομένων το αρχείο καταγραφής του EPA, που βρίσκεται στα Internet Traffic Archives [69], αλλά και αυτά της Διαδικτυακής Κοινότητας Φοιτητών της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών[22] καθώς και του επίσημου ιστοτόπου της σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών [70]. Να σημειωθεί ότι στα επόμενα, δεν μπορούσαν να παραχθούν όλα τα στατιστικά αποτελέσματα για το αρχείο καταγραφής του [22] λόγω της μορφής του [71].

4.2.2 Αριθμός αιτημάτων επισκεπτών

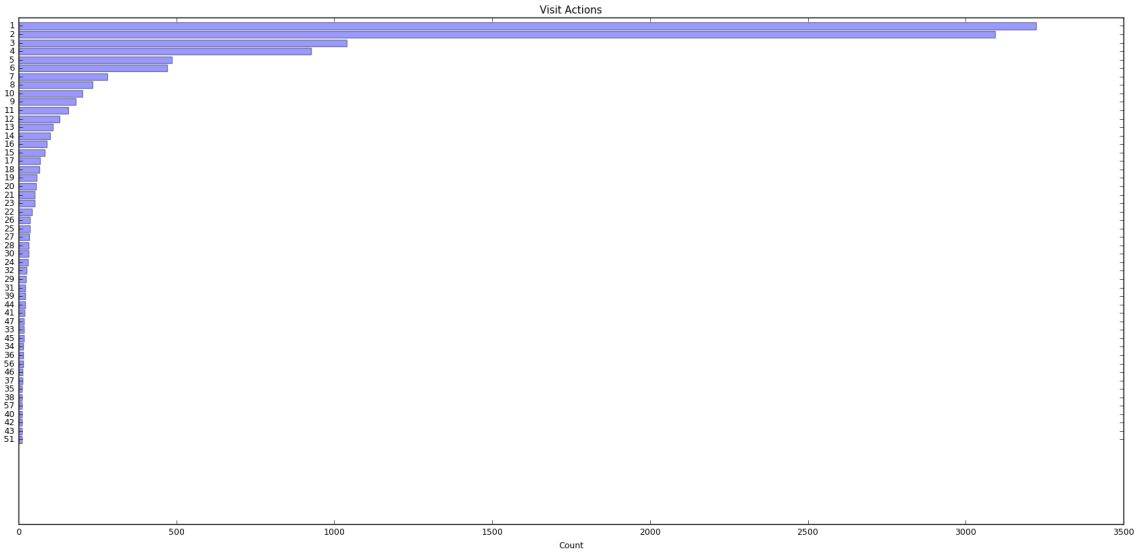
Ένα σημαντικό στατιστικό αποτέλεσμα που μπορεί να εξαχθεί σχετικά εύκολα από τα αρχεία καταγραφής είναι ο αριθμό των αιτημάτων που κάνει κάθε επισκέπτης της ιστοσελίδας. Μελετώντας το μέσο αριθμό των αιτημάτων των επισκεπτών, μπορεί κάποιος να καταλάβει αν η σελίδα παρουσιάζει ενδιαφέρον για τους επισκέπτες της ή αν χρειάζεται κάποια τροποποίηση. Φυσικά, η πλειοψηφία των χρηστών μιας ιστοσελίδας θα “προσπελάσουν” ελάχιστες υποσελίδες και επομένως η κατανομή θα είναι προς τα λίγα αιτήματα, κάτι που επιβεβαιώνεται και από τα διαγράμματα που παράχθηκαν από τα αρχεία καταγραφής EPA, ece, shmmy (Σχήματα 4.3, 4.4, 4.5).



Σχήμα 4.3: Διάγραμμα αριθμού αιτημάτων επισκεπτών από το EPA αρχείο καταγραφής



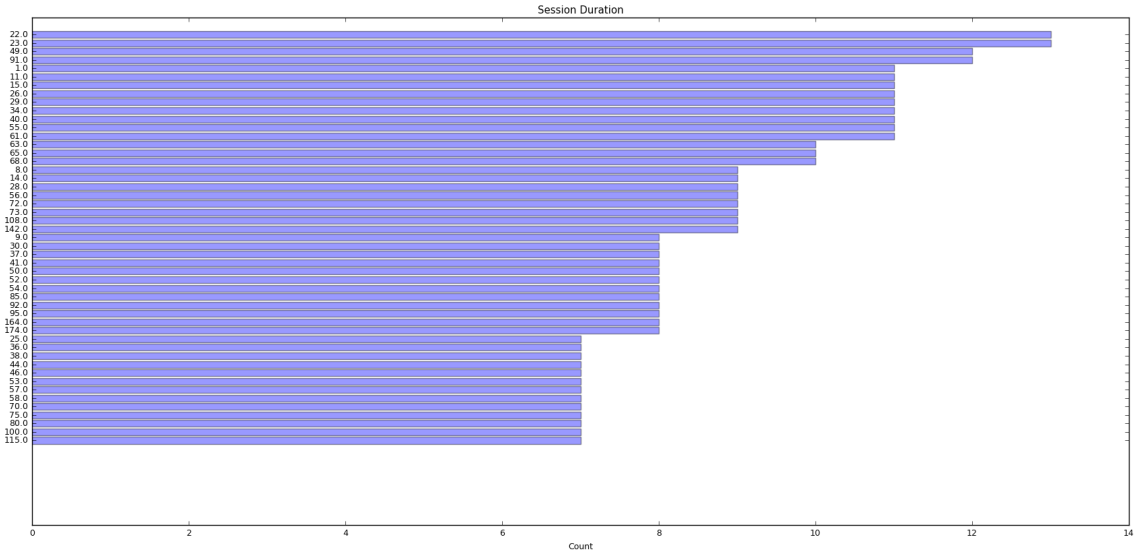
Σχήμα 4.4: Διάγραμμα αριθμού αιτημάτων επισκεπτών από το ECE αρχείο καταγραφής



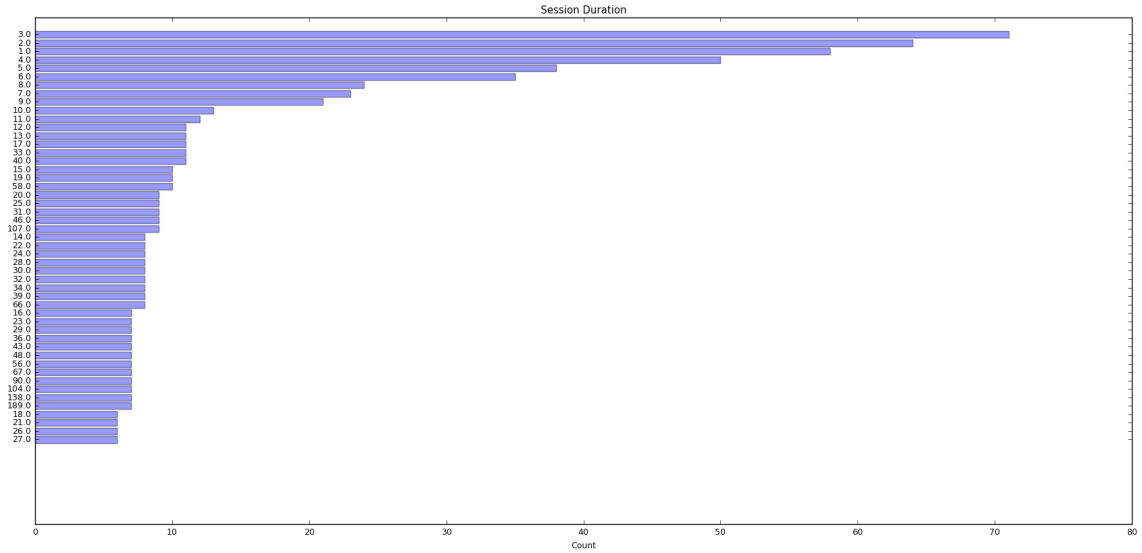
Σχήμα 4.5: Διάγραμμα αριθμού αιτημάτων επισκεπτών από το shmmy αρχείο καταγραφής

4.2.3 Διάρκεια συνεδρίας χρήστη

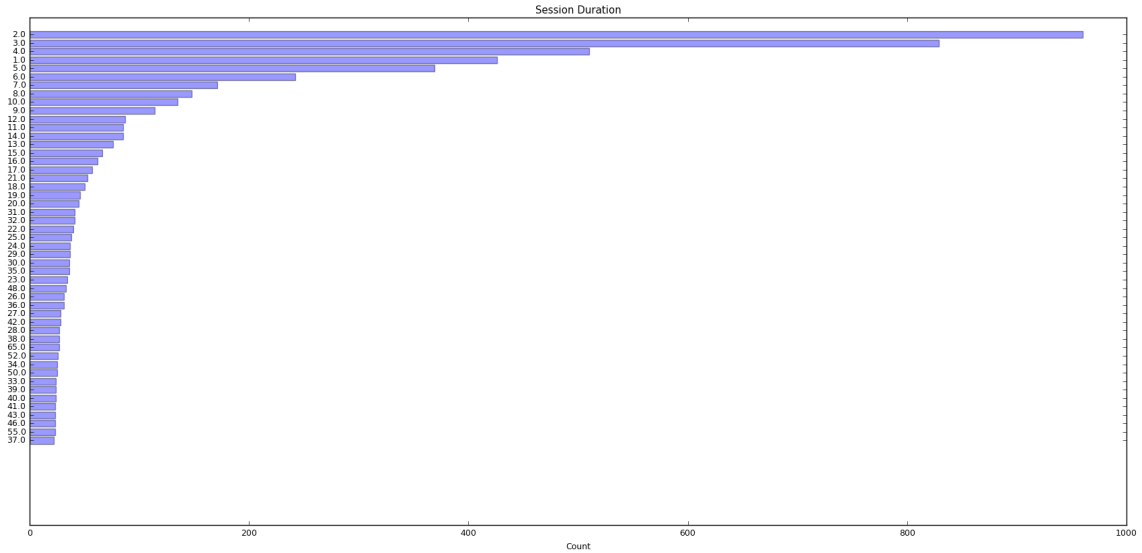
Ακόμα μια σημαντική στατιστική παράμετρος είναι ο αριθμός που διήρκεσε ένα session ενός χρήστη της ιστοσελίδας. Καταρχάς, δεν είναι δυνατόν να γνωρίζουμε από τα αρχεία καταγραφής πόσο χρόνο πέρασε στην τελευταία σελίδα ο χρήστης, κάτι που είναι ιδιαίτερα προβληματικό ειδικά για τους χρήστες που έχουν μόνο ένα αίτημα στην ιστοσελίδα (που είναι και η πλειοψηφία των χρηστών στη γενικότερη περίπτωση). Όπως και στην προηγούμενη περίπτωση, μεγάλος χρόνος session υποδηλώνει μεγαλύτερο ενδιαφέρον επισκεπτών. Επειδή ο χρόνος που περνάει ένας μέσος χρήστης σε μία ιστοσελίδα είναι αρκετά μικρός και πάλι η κατανομή θα είναι προς το μικρότερο χρόνο session. Όλα αυτά επιβεβαιώνονται και από τα διαγράμματα που παράχθηκαν από τα αρχεία καταγραφής EPA, ece, shmmy (Σχήματα 4.6, 4.7, 4.8).



Σχήμα 4.6: Διάγραμμα διάρκειας session σε δευτερόλεπτα από το EPA αρχείο καταγραφής



Σχήμα 4.7: Διάγραμμα διάρκειας session σε δευτερόλεπτα από το ECE αρχείο καταγραφής



Σχήμα 4.8: Διάγραμμα διάρκειας session σε δευτερόλεπτα από το shmmy αρχείο καταγραφής

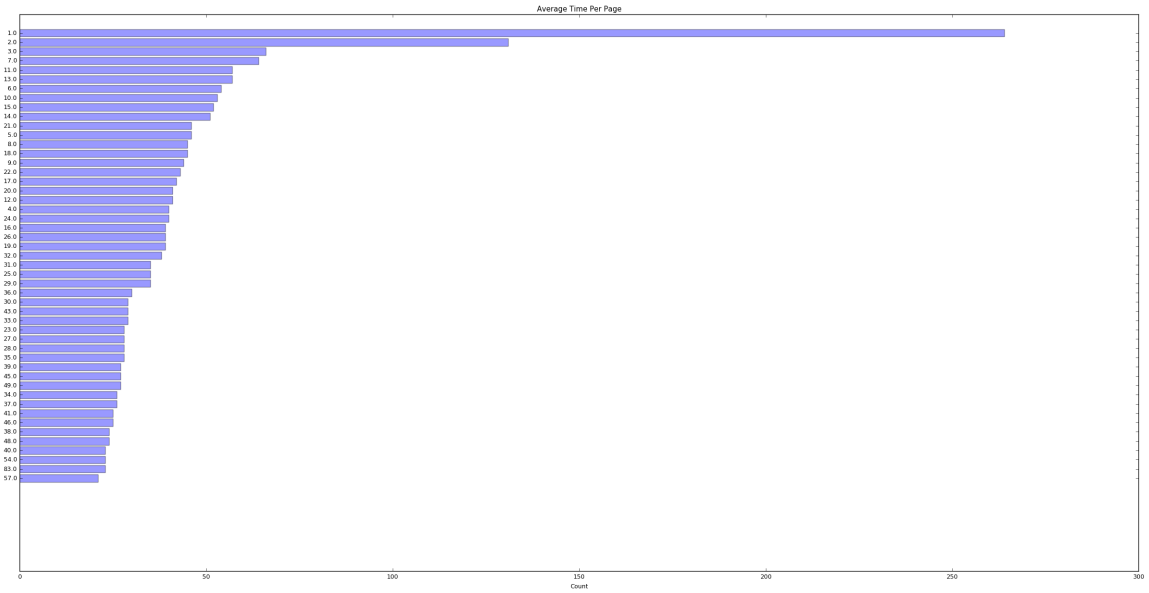
4.2.4 Μέσος χρόνος ανά σελίδα

Ο μέσος χρόνος ανά σελίδα υπολογίζεται για όλα τα sessions ως εξής:

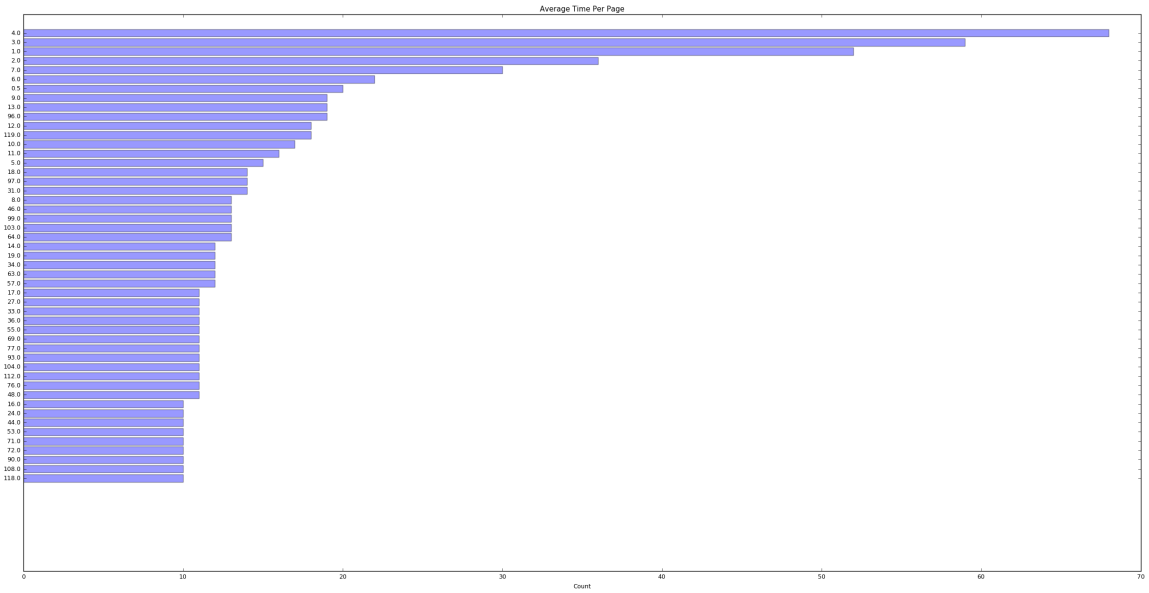
$$\text{μέσος χρόνος ανά σελίδα } = \frac{\text{διάρκεια session}}{\text{αριθμός αιτημάτων χρήστη - 1}} \tag{4.1}$$

Αφαιρούμε ένα από τον αριθμό των αιτημάτων χρήστη γιατί ο χρόνος της τελευταίας σελίδας δεν μπορεί να υπολογιστεί στη διάρκεια του session όπως αναφέραμε και πριν. Όπως και στην προηγούμενη στατιστική ανάλυση, μεγάλος μέσος χρόνος ανά σελίδα υποδηλώνει μεγαλύτερο ενδιαφέρον επισκεπτών. Επειδή ο χρόνος που περνάει ένας μέσος χρήστης σε μία ιστοσελίδα είναι αρκετά μικρός και πάλι η κατανομή θα είναι προς τα πάνω, δηλαδή προς το μικρότερο μέσο χρόνο ανά σελίδα. Όλα αυτά επιβεβαιώνονται και από τα διαγράμματα που παράχθηκαν από τα αρχεία

καταγραφής EPA, ECE (Σχήματα 4.9, 4.10).



Σχήμα 4.9: Διάγραμμα μέσου χρόνου ανά σελίδα σε δευτερόλεπτα από το EPA αρχείο καταγραφής

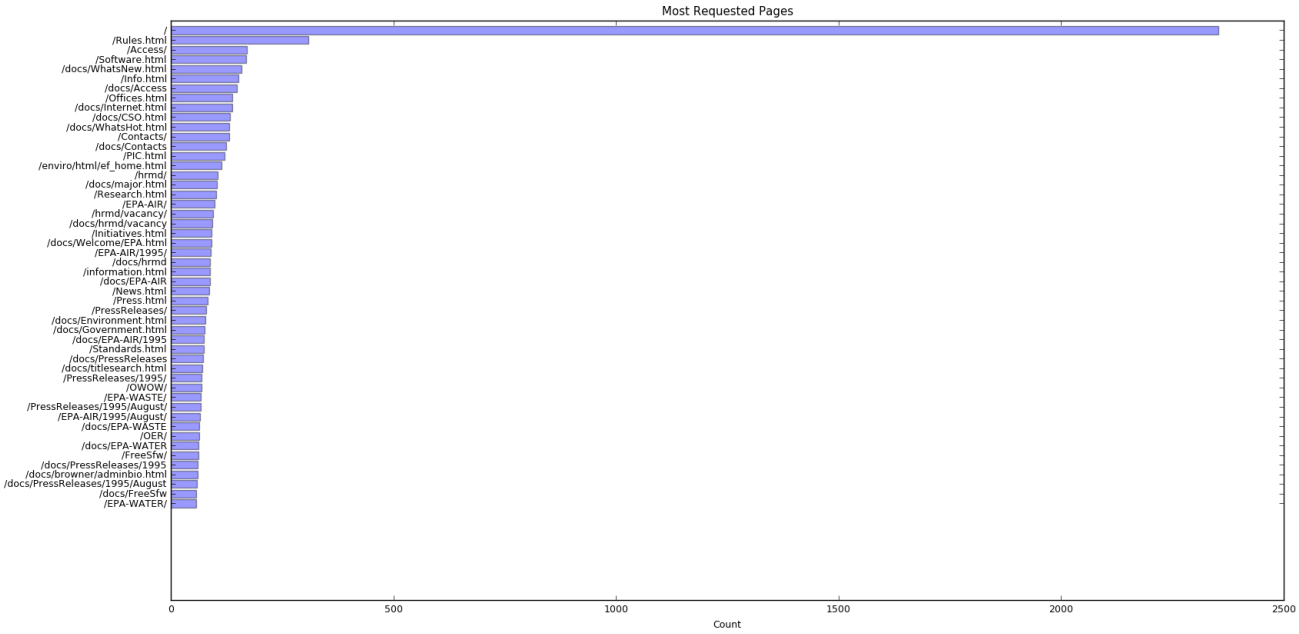


Σχήμα 4.10: Διάγραμμα μέσου χρόνου ανά σελίδα σε δευτερόλεπτα από το ECE αρχείο καταγραφής

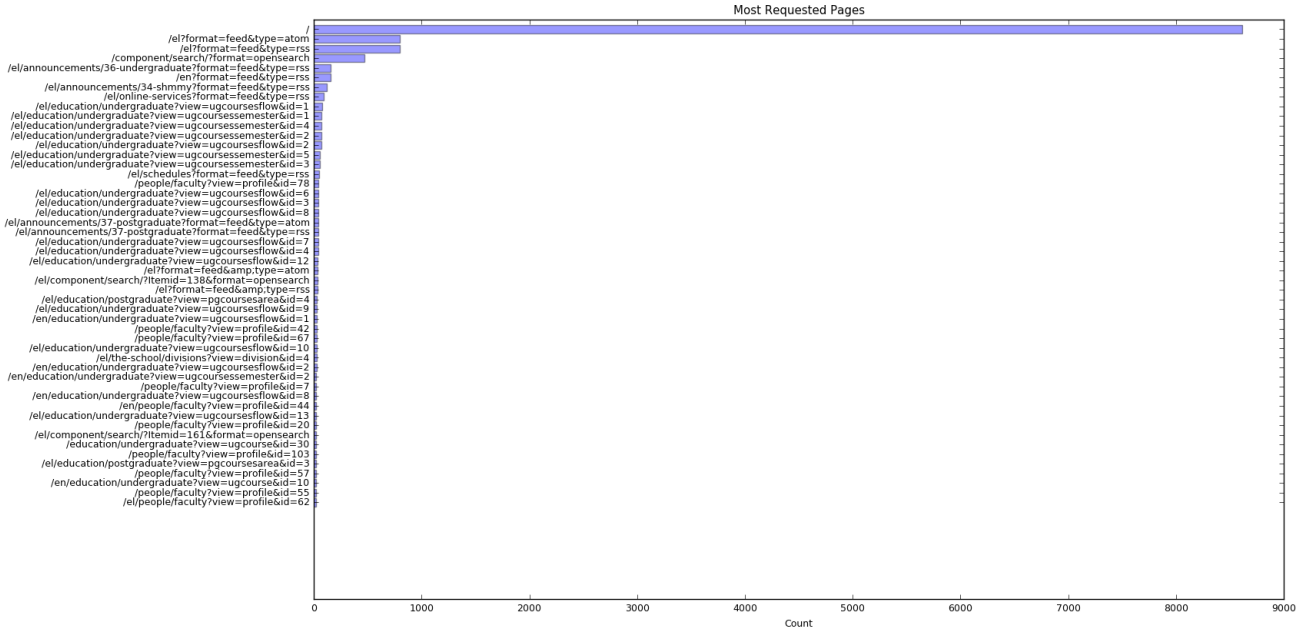
4.2.5 Σελίδες με τη μεγαλύτερη ζήτηση

Το πιο ενδιαφέρον στατιστικό που μπορεί να εξαχθεί σε αυτό το στάδιο είναι οι σελίδες που έχουν τη μεγαλύτερη ζήτηση. Σίγουρα τη μεγαλύτερη ζήτηση θα την έχει η αρχική σελίδα ενός ιστοτόπου καθώς από αυτή εισέρχονται οι περισσότεροι επισκέπτες αλλά και αυτή περιέχει τους περισσότερους συνδέσμους προς τις άλλες σελίδες του ιστοτόπου. Όμως, πέρα από αυτή μπορούμε να δούμε ποιες άλλες σελίδες έχουν αρκετό ενδιαφέρον και ζήτηση, με απώτερο στόχο για παράδειγμα την ανανέωση της σχεδίασης του ιστοτόπου, κάνοντάς τον πιο φιλικό προς τον χρήστη.

Παρακάτω παρουσιάζονται οι σελίδες με τη μεγαλύτερη ζήτηση στα αρχεία καταγραφής των EPA, ECE που επιβεβαιώνουν κάποιες από τις προαναφερθείσες παρατηρήσεις (Σχήματα 4.11, 4.12).



Σχήμα 4.11: Διάγραμμα σελίδων με τη μεγαλύτερη ζήτηση από το EPA αρχείο καταγραφής



Σχήμα 4.12: Διάγραμμα σελίδων με τη μεγαλύτερη ζήτηση από το ECE αρχείο καταγραφής

Κεφάλαιο 5

Κανόνες συσχέτισης

5.1 Εισαγωγή

Η εξαγωγή κανόνων συσχέτισης (association rule mining) είναι ίσως η πιο σημαντική τεχνική ανακάλυψης μοτίβων. Με τον όρο ”κανόνες συσχέτισης”, εννοούμε να βρούμε όλες τις σχέσεις που συνυπάρχουν ανάμεσα σε ένα σύνολο αντικειμένων. Αυτή η τεχνική, που είναι γνωστή και ως εξόρυξη συχνών ομάδων αντικειμένων (frequent itemset mining), αρχικά αναπτύχθηκε για την “ανάλυση του καλαθιού της αγοράς” (market basket analysis), της οποίας ο σκοπός είναι να βρει πώς οι καταναλωτές αγοράζουν προϊόντα. Για παράδειγμα ένας κλασσικός κανόνας συσχέτισης είναι ο εξής:

Πάνες → Μπύρες [υποστήριξη = 20%, εμπιστοσύνη= 80%]

Αυτός ο κανόνας αναφέρει ότι το 20% των πελατών αγοράζουν πάνες και μπύρες μαζί και αυτοί που αγοράζουν πάνες, αγοράζουν επίσης μπύρες το 80% των φορών.

Λόγω της χρησιμότητάς της, η εξαγωγή κανόνων συσχέτισης είναι μια πολύ ενεργή ερευνητική περιοχή και στην περίπτωση του Παγκοσμίου Ιστού χρησιμοποιείται, μεταξύ άλλων για την εύρευση ομάδων σελίδων που προσπελάζονται μαζί. Για παράδειγμα, έχει χρησιμοποιηθεί μαζί με τεχνικές συσταδοποίησης, προκειμένου να γίνεται εξατομίκευση της ιστοσελίδας σε πραγματικό χρόνο[72]. Επίσης, έχουν προταθεί τεχνικές για εξαγωγή έμμεσων κανόνων συσχέτισης, οι οποίοι σχετίζονται με άμεσους κανόνες συσχέτισης σε ένα σύνολο σύνθετων κανόνων συσχέτισης που χρησιμοποιούνται για την σύσταση ιστοσελίδων[73]. Επιπρόσθετα, μία ακόμα προσέγγιση για τους κανόνες συσχέτισης, είναι ο συνδυασμός τους με χρονικούς και ακολουθιακούς περιορισμούς. Όταν αυτή η προσέγγιση δοκιμάστηκε σε πραγματικά αρχεία καταγραφής με τη μέθοδο της ανάλυσης διακύμανσης, αποδείχθηκε ότι χρονικοί και οι ακολουθιακοί περιορισμοί και ο συνδυασμός τους, έχουν μεγάλη επίδραση στην ακρίβεια της πρόβλεψης [74]. Παρατηρήθηκε ακόμη, ότι οι χρονικοί περιορισμοί έχουν μεγαλύτερη επίδραση από τους ακολουθιακούς[74].

Το μεγαλύτερο πρόβλημα με την εξαγωγή κανόνων συσχέτισης και την εύρεση συχνών ομάδων αντικειμένων είναι ότι τα αντικείμενα τα οποία θα βρίσκονται συχνά μαζί θα εμφανίζονται και σε πολλούς από τους εξαχθέντες κανόνες με αποτέλεσμα ανακριβείς προβλέψεις[50]. Έτσι, όσο μεγαλώνει το σύνολο των δεδομένων οι προβλέψεις αντί να βελτιώνονται γίνονται χειρότερες. Τέλος, η εξαγωγή κανόνων συσχέτισης δεν λαμβάνει υπόψιν τη σειρά με την οποία προστίθενται τα αντικείμενα σε ένα σύνολο, κάτι που γίνεται από την εξόρυξη διαδοχικών μοτίβων (sequential pattern mining)[75].

5.2 Βασικές έννοιες των κανόνων συσχέτισης

Η εξόρυξη κανόνων συσχέτισης ορίζεται μαθηματικά ως εξής: Έστω $I = i_1, i_2, \ldots, i_m$ ένα σύνολο δεδομένων και $T = (t_1, t_2, \ldots, t_n)$ ένα σύνολο συναλλαγών, όπου κάθε συναλλαγή t_i είναι ένα σύνολο αντικειμένων για το οποίο ισχύει $t_i \subseteq I$. Ένας κανόνας συσχέτισης μπορεί να οριστεί ως ένας κανόνας της μορφής

$$X \to Y \text{ όπου } X \subseteq I, Y \subseteq I \text{ και } X \cap Y = \emptyset \tag{5.1}$$

Τα X, Y είναι σύνολα αντικειμένων. Στο παράδειγμα της ενότητας 5.1, η συναλλαγή είναι το σύνολο {Πάνες, Μπύρες} και I είναι όλο το σύνολο των δεδομένων που πουλιούνται στο συγκεκριμένο κατάστημα. Στον κανόνα συσχέτισης X είναι το σύνολο {Πάνες} και Y είναι το σύνολο {Μπύρες}. Λέμε ότι μία συναλλαγή $t_i \in T$ περιλαμβάνει το σύνολο αντικειμένων X, αν το X είναι υποσύνολο του t_i. Η καταμέτρηση υποστήριξης του X στο T, που συμβολίζεται ως $X.count$, είναι ο αριθμός των συναλλαγών στο T που περιέχουν το X. Το πόσο καλός είναι ένας κανόνας μετριέται από την υποστήριξη (support) και την εμπιστοσύνη (confidence) του.

5.2.1 Υποστήριξη

Η υποστήριξη ενός κανόνα, $X \to Y$, είναι το ποσοστό των συναλλαγών στο T που περιέχει το $X \cup Y$ και μπορεί να ερμηνευθεί σαν την πιθανότητα $\Pr(X \cup Y)$[6]. Η υποστήριξη ενός κανόνα λοιπόν, καθορίζει πόσο συχνός είναι αυτός στο σύνολο των συναλλαγών T. Αν n είναι ο συνολικός αριθμός συναλλαγών στο T, η υποστήριξη ενός κανόνα υπολογίζεται ως:

$$\text{support} = \frac{(X \cup Y).count}{n} \tag{5.2}$$

Η υποστήριξη είναι ένα πολύ χρήσιμο μέγεθος. Αν είναι πολύ χαμηλή, τότε ο συγκεκριμένος κανόνας μπορεί να είναι απλά τυχαίος. Επίσης, σε περιβάλλον στο οποίο εμπλέκεται κέρδος, η εφαρμογή ενός τέτοιου κανόνα μπορεί να είναι ζημιογόνος.

5.2.2 Εμπιστοσύνη

Η υποστήριξη ενός κανόνα, $X \to Y$, είναι το ποσοστό των συναλλαγών στο T, που όταν περιέχουν το X περιέχουν επίσης το Y[6]. Μπορεί να ερμηνευθεί σαν την δεσμευμένη πιθανότητα, $\Pr(X|Y)$ και υπολογίζεται ως:

$$\text{confidence} = \frac{(X \cup Y).count}{X.count} \tag{5.3}$$

Με βάση τον παραπάνω ορισμό, η εμπιστοσύνη καθορίζει την προβλεψιμότητα ενός κανόνα. Αν η εμπιστοσύνη του κανόνα $X \to Y$ είναι χαμηλή, τότε το Y δεν συμπεραίνεται με αξιοπιστία από το X, και επομένως ο κανόνας έχει χαμηλή προβλεψιμότητα και άρα περιορισμένη χρήση.

Συνοψίζοντας, θα μπορούσαμε να πούμε ότι ο σκοπός των κανόνων συσχέτισης, δοθέντος ενός συνόλου συναλλαγών T, είναι η εύρεση όλων των σχέσεων στο T που έχουν υποστήριξη και εμπι-

στοσύνη μεγαλύτερη ή ίση από κάποιες προκαθορισμένες τιμές ελάχιστης υποστήριξης και ελάχιστης εμπιστοσύνης. Υπάρχει ένας μεγάλος αριθμός αλγορίθμων εξόρυξης κανόνων συσχέτισης, με διαφορετικές δυνατότητες και αποδοτικότητα. Τα αποτελέσματά τους όμως είναι τα ίδια λαμβάνοντας υπόψιν τον ορισμό των κανόνων συσχέτισης, δηλαδή δοθέντος ενός σύνολου συναλλαγών T, μίας ελάχιστης υποστήριξης και μιας ελάχιστης εμπιστοσύνης, να βρεθεί ένα σύνολο σχέσεων στο T. Οι κανόνες που θα βρεθούν θα είναι ίδιοι από όλους τους αλγορίθμους, αλλά οι υπολογισμοί και οι απαιτήσεις τους σε μνήμη και χρόνο είναι διαφορετικές [76]. Στα πλαίσια της διπλωματικής εργασίας θα αναλυθούν οι τρεις δημοφιλέστεροι αλγόριθμοι εξόρυξης κανόνων συσχέτισης ή εύρεσης συχνών αντικειμένων καθώς και ένας σχετικά πρόσφατος και συγκεκριμένα οι Apriori, Eclac, FP-Growth, και SaM.

5.3 Ιδιότητες της υποστήριξης ενός συνόλου αντικειμένων

Η εξαντλητική αναζήτηση (exhaustive search) όλων των πιθανών υποσυνόλων αντικειμένων προκειμένου να βρεθούν αυτά που έχουν παραπάνω από μία ελάχιστη υποστήριξη είναι αδύνατη, διότι για κάθε σύνολο n αντικειμένων, I, υπάρχουν 2^{n-1} πιθανά υποσύνολα που πρέπει να ερευνηθούν. Επομένως ο αριθμός των συνόλων αντικειμένων αυξάνεται εκθετικά με τον αριθμό των αντικειμένων. Από τον ορισμό της υποστήριξης προκύπτει ότι για σύνολα I, J και αν συμβολίσουμε την υποστήριξη ενός συνόλου ως $s_T()$:

$$\forall I : \forall J \supseteq I : \quad s_T(J) \leq s_T(I) \tag{5.4}$$

Η παραπάνω σχέση δηλώνει ότι αν ένα σύνολο αντικειμένων επεκταθεί, τότε η υποστήριξή του δεν μπορεί να αυξηθεί. Αυτή η ιδιότητα, είναι γνωστή ως αντι-μονοτονικότητα ή κλειστότητα προς τα κάτω[6]. Από την παραπάνω σχέση μπορούμε να συμπεράνουμε ότι:

$$\forall s_{min} : \forall I : \forall J \supseteq I : \quad s_T(I) < s_{min} \rightarrow s_T(J) < s_{min} \tag{5.5}$$

δηλαδή ότι κανένα υπερσύνολο ενός συνόλου μη-συχνών αντικειμένων δεν μπορεί να είναι συχνό. Αυτή η ιδιότητα είναι γνωστή ως Apriori και η άρνησή της είναι πολύ χρήσιμη στην εξαγωγή συνόλων συχνών αντικειμένων. Η άρνηση της ιδιότητας Apriori μπορεί να εκφραστεί ως:

$$\forall s_{min} : \forall I : \forall J \subseteq I : \quad s_T(I) \geq s_{min} \rightarrow s_T(J) \geq s_{min} \tag{5.6}$$

δηλαδή κάθε υποσύνολο ενός συνόλου συχνών αντικειμένων είναι και εκείνο συχνό [6].

5.3.1 Μερικώς διατεταγμένα σύνολα

Ως μερική διάταξη ορίζουμε μία δυαδική σχέση[77] σε ένα σύνολο S που $\forall a, b, c \in S$ ικανοποιεί τις εξής τρεις ιδιότητες:

- $a \leq a$ (ανακλαστικότητα)
- $a \leq b \wedge b \leq a \rightarrow a = b$ (αντισυμμετρικότητα)
- $a \leq b \wedge b \leq c \rightarrow a \leq c$ (μεταβατικότητα)

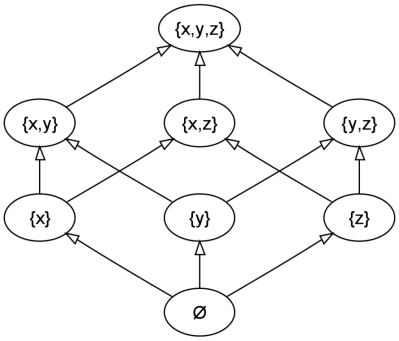
Ένα σύνολο που έχει μερική διάταξη, ονομάζεται μερικώς διατεταγμένο σύνολο. Αν a, b είναι δύο διαφορετικά στοιχεία ενός μερικώς διατεταγμένου συνόλου τότε αν $a \leq b$ ή $b \leq a$ τα στοιχεία αυτά ονομάζονται συγκρίσιμα, ενώ αν δεν ισχύει καμία από τις δύο παραπάνω σχέσεις τότε αυτά τα στοιχεία ονομάζονται μη-συγκρίσιμα. Αν όλα τα στοιχεία ενός συνόλου S είναι συγκρίσιμα μεταξύ τους τότε λέμε ότι το σύνολο έχει ολική διάταξη. Σε ένα σύνολο με ολική διάταξη, η αντισυμμετρικότητα αντικαθίσταται από την εξής ισχυρότερη ιδιότητα:

- $a \leq b \vee b \leq a$ (ολότητα)

Επίσης, χρήσιμοι είναι οι παρακάτω ορισμοί από τη μαθηματική ανάλυση οι οποίοι θα προσαρμοστούν στη διάταξη συνόλων:

- Μία συνάρτηση $f : S \to R$, όπου S και R είναι δύο μερικώς διατεταγμένα σύνολα ονομάζεται μονότονη αν $\forall x, y \in S : \ x \leq_S y \to f(x) \leq_R f(y)$.
- Μία συνάρτηση $f : S \to R$, όπου S και R είναι δύο μερικώς διατεταγμένα σύνολα ονομάζεται μη-μονότονη αν $\forall x, y \in S : \ x \leq_S y \to f(x) \geq_R f(y)$.

Τέλος, κάθε μερικώς διατεταγμένο σύνολο $(S, \leq)$ μπορεί να αναπαρασταθεί ως ένας γράφος G, που δεν περιέχει κύκλους, που ονομάζεται διάγραμμα Hasse.



Σχήμα 5.1: Διάγραμμα Hasse

Ένα διάγραμμα Hasse έχει τα στοιχεία του S ως κορυφές και οι ακμές του μπαίνουν ως εξής: Αν x, y είναι δύο στοιχεία του S με $x < y$ και δεν υπάρχει κάποιο στοιχείο z τέτοιο ώστε $x < z < y$, τότε υπάρχει ακμή από το x στο y.

5.4 Αλγόριθμος Apriori

Η γενική διαδικασία εξερεύνησης για συχνά σύνολα αντικειμένων, ουσιαστικά είναι μια διαδικασία απαρίθμησης, όπου βρίσκονται όλα τα πιθανά σύνολα αντικειμένων και υπολογίζεται η υποστήριξή τους. Ο χώρος καταστάσεων είναι το μερικώς διατεταγμένο σύνολο $(2^B, \subseteq)$ όπου B είναι το σύνολο των αντικειμένων. Ο πιο διαδεδομένος αλγόριθμος αναζήτησης συχνών αντικειμένων είναι ο Apriori. Ο αλγόριθμος αυτός προτάθηκε το 1994 από τους Rakesh Agrawal και Ramakrishnan Srikant[78]. Ακολουθεί ο ψευδοκώδικας του αλγορίθμου Apriori όπως βρίσκεται στο [78].

Αλγόριθμος 3 Ψευδοκώδικας αλγορίθμου Apriori

function APRIORI(T, ϵ)
  $L_1 \leftarrow \{\text{large 1 - itemsets}\}$
  $k \leftarrow 2$
  while $L_{k-1} \neq 0$ **do**
    $C_k \leftarrow \{a \cup \{b\} \mid a \in L_{k-1} \wedge b \notin a\} - \{c \mid \{s \mid s \subseteq c \wedge |s| = k-1\} \nsubseteq L_{k-1}\}$
    for transactions $t \in T$ **do**
      $C_t \leftarrow \{c \mid c \in C_k \wedge c \subseteq t\}$
      for candidates $c \in C_t$ **do**
        $count[c] \leftarrow count[c] + 1$
      end for
    end for
    $L_k \leftarrow \{c \mid c \in C_k \wedge count[c] \geq \epsilon\}$
    $k \leftarrow k + 1$
  end while
return $\bigcup_k L_k$
end function

Αρχικά ο αλγόριθμος, υπολογίζει την υποστήριξη όλων των μονοσυνόλων και απορρίπτει όσα δεν είναι αρκετά συχνά. Στη συνέχεια φτιάχνει υποψήφια σύνολα αντικειμένων που έχουν πληθικότητα δύο, που αποτελούνται από συχνά αντικείμενα, υπολογίζει την υποστήριξή τους και απορρίπτει όσα δεν είναι συχνά. Η διαδικασία επαναλαμβάνεται για σύνολα με αυξανόμενη πληθικότητα που πάντα αποτελούνται από τα σύνολα του προηγούμενου βήματος, τα οποία έχουν υποστήριξη πάνω από την ελάχιστη. Ο αλγόριθμος τερματίζει όταν κανένα υποψήφιο σύνολο δεν έχει την απαιτούμενη υποστήριξη. Συνοψίζοντας, θα μπορούσαμε να πούμε ότι ο αλγόριθμος Apriori βασίζεται στη δημιουργία υποψηφίων συνόλων και στο "κλάδεμα" όσων δεν ικανοποιούν κάποια κριτήρια.

5.5 Δημιουργία υποψηφίων συνόλων

Ο αλγόριθμος Apriori εξερευνεί ολοένα και μεγαλύτερα σύνολα συχνών αντικειμένων. Όμως, σε κάθε βήμα k κάθε σύνολο μεγέθους $k+1$ μπορεί να δημιουργηθεί με $\frac{k(k+1)}{2}$ τρόπους, κάτι που γίνεται σε κάθε βήμα, ενώ θα αρκούσε η δημιουργία κάθε υποψήφιου συνόλου με ένα τρόπο. Ένας άλλος τρόπος να δούμε το πρόβλημα της δημιουργίας υποψηφίων συνόλων είναι ότι ένα σύνολο μεγέθους k μπορεί να δημιουργηθεί με $k!$ τρόπους, επειδή τα αντικείμενα μπορεί να προστεθούν με οποιαδήποτε σειρά. Ένας τρόπος, μείωσης αυτής της "αχρείαστης" πολυπλοκότητας είναι να αναθέσουμε σε κάθε υποψήφιο σύνολο δεδομένων, μόνο ένα προγενέστερο σύνολο δεδομένων από το οποίο μπορεί να δημιουργηθεί. Για παράδειγμα, ένα σύνολο της μορφής $\{a, b\}$ θα μπορεί να δημιουργηθεί μόνο από το σύνολο $\{a\}$ και όχι από το σύνολο $\{b\}$. Ο αλγόριθμος δημιουργίας υποψηφίων συνόλων με μοναδικούς γονείς[79] μπορεί να περιγραφεί ως εξής:

Αλγόριθμος 4 Αλγόριθμος Δημιουργίας Υποψήφιων Συνόλων

- Αρχικά, παράγονται όλα τα μονοσύνολα διότι οι γονείς τους είναι το κενό σύνολο.
- Στη συνέχεια, επεξεργάζονται αναδρομικά όλα τα μονοσύνολα τα οποία είναι συχνά με τον εξής τρόπο:
  - Για ένα δοθέν συχνό σύνολο αντικειμένων I, δημιουργούνται όλα τα σύνολα J που προκύπτουν από το I με την προσθήκη ενός αντικειμένου και για αυτά το I θεωρείται ότι θα είναι ο μοναδικός γονιός.
  - Για όλα τα δημιουργηθέντα σύνολα αντικειμένων J, ελέγχεται το καθένα ξεχωριστά αν είναι συχνό. Αν δεν είναι, τότε απορρίπτεται. Σε αντίθετη περίπτωση, επεξεργάζεται και αυτό αναδρομικά.

Προκειμένου να γίνει μια πιο αυστηρή προσέγγιση του αλγορίθμου θα πρέπει να οριστούν οι κανονικές μορφές ενός συνόλου αντικειμένων.

5.6 Κανονικές μορφές - Δέντρα προθεμάτων

Ως κανονική μορφή (canonical form) ενός αντικειμένου[80], ορίζουμε μία πρότυπη, μοναδική αναπαράστασή του. Σε αυτή την παράγραφο, θα ορίσουμε μια κανονική μορφή ενός συνόλου αντικειμένων και στη συνέχεια αυτή η κανονική μορφή θα χρησιμοποιηθεί προκειμένου να ανατεθούν μοναδικοί γονείς σε κάθε σύνολο αντικειμένων.

Ένα σύνολο αντικειμένων, αναπαρίσταται από μία κωδική λέξη, όπου κάθε γράμμα του αλφαβήτου B, αναπαριστά ένα αντικείμενο. Υπάρχουν $k!$ κωδικές λέξεις για κάθε σύνολο αντικειμένων μήκους k, επειδή τα αντικείμενα μπορεί να δοθούν σε οποιαδήποτε σειρά. Εισάγοντας μια διάταξη των αντικειμένων και συγκρίνοντας τις κωδικές λέξεις λεξικογραφικά, μπορούμε να ορίσουμε μια διάταξη και στις κωδικές λέξεις. Ορίζουμε ως κανονική κωδική λέξη εκείνη που είναι μικρότερη από όλες τις υπόλοιπες που αποτελούνται από τα ίδια αντικείμενα.

Αν θεωρήσουμε I ένα σύνολο αντικειμένων και $w_c(I)$ την κανονική κωδική λέξη του, τότε ως κανονικός γονέας $p_c(I)$ ενός συνόλου αντικειμένων I, ορίζεται το σύνολο αντικειμένων που περιγράφεται από το μεγαλύτερο κύριο πρόθεμα της κωδικής λέξης $w_c(I)$. Ένας ισοδύναμος μαθηματικός ορισμός του κανονικού γονέα είναι ο

$$p_c(I) = I - \{\max_{a \in I} a\} \tag{5.7}$$

Με αυτούς τους ορισμούς, ο προηγούμενος αναδρομικός αλγόριθμος για ένα συχνό σύνολο αντικειμένων I, πλέον μπορεί να γραφτεί ως εξής:

Αλγόριθμος 5 Βελτιωμένος Αλγόριθμος Δημιουργίας Υποψήφιων Συνόλων

- Αρχικά δημιουργούνται όλα τα σύνολα J που προκύπτουν από το I με την προσθήκη ενός αντικειμένου.
- Σχηματίζεται η κανονική κωδική λέξη $w_c(J)$ κάθε συνόλου J.
- Για κάθε σύνολο J, αν το τελευταίο γράμμα του $w_c(J)$ που είναι το τελευταίο αντικείμενο που προστέθηκε στο I από το J είναι συχνό, γίνεται αναδρομική επεξεργασία του J, αλλιώς το J απορρίπτεται.

Η προαναφερθείσα αναπαράσταση ενός συνόλου δεδομένων έχει την ιδιότητα του προθέματος, η οποία λέει ότι το μεγαλύτερο κύριο πρόθεμα μιας κανονικής κωδικής λέξης ενός συνόλου αντικειμένων είναι μια κανονική κωδική λέξη και εκείνο. Αυτό έχει ως αποτέλεσμα ότι αν γνωρίζουμε το μεγαλύτερο κύριο πρόθεμα μιας κανονικής κωδικής λέξης ενός συνόλου αντικειμένων I, δεν γνωρίζουμε μόνο τον κανονικό γονέα του I, αλλά και την κανονική κωδική του λέξη.

Η ιδιότητα του προθέματος, διευκολύνει την εξερεύνηση του χώρου αναζήτησης, επειδή από αυτή σε συνδυασμό με την κανονική μορφή μπορούμε να δημιουργήσουμε ευκολότερα μεγαλύτερα συχνά σύνολα αντικειμένων[80]. Αυτό προκύπτει, επειδή η προσθήκη ενός αντικειμένου σε ένα σύνολο αντικειμένων που δεν τηρεί τη διάταξη των αντικειμένων του συνόλου, δημιουργεί μη κανονικές μορφές και επομένως μπορεί να απορριφθεί. Στην περίπτωση του Apriori αυτό σημαίνει ότι η δημιουργία συνόλων μεγέθους $k+1$ γίνεται συνδυάζοντας δύο σύνολα μεγέθους k, έστω f_1, f_2 για τα οποία ισχύει:

$$f_1 = \{i_1, \ldots, i_{k-1}, i_k\} \tag{5.8}$$

$$f_2 = \{i_1, \ldots, i_{k-1}, i_k^{'}\} \tag{5.9}$$

$$i_k < i_k^{'} \tag{5.10}$$

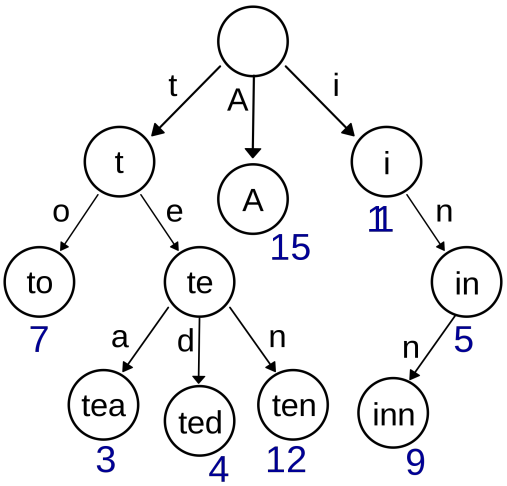
$$\forall j, 1 \leq j < k : i_j < i_{j+1} \tag{5.11}$$

Μετά από αυτό ο αλγόριθμος δημιουργίας υποψήφιων συνόλων με μοναδικούς γονείς διαμορφώνεται ως εξής:

Αλγόριθμος 6 Αλγόριθμος Δημιουργίας Υποψήφιων Συνόλων με Μοναδικούς Γονείς

- Αρχικά, δημιουργούνται όλα τα πιθανά μονοσύνολα, τα οποία βρίσκονται ήδη σε κανονική μορφή.
- Έπειτα, επεξεργάζεται αναδρομικά κάθε κωδική λέξη που περιγράφει ένα συχνό σύνολο αντικειμένων με τον εξής τρόπο:
  - Προστίθεται σε κάθε κωδική λέξη ένα γράμμα, το οποίο με βάση την διάταξη που έχουμε ορίσει βρίσκεται μετά από το τελευταίο γράμμα της λέξης.
  - Αν η λέξη που σχηματίζεται αναπαριστά ένα συχνό σύνολο αντικειμένων τότε την κρατάμε, αλλιώς την απορρίπτουμε.

Με αυτό τον τρόπο κάθε υποψήφιο σύνολο αντικειμένων δημιουργείται μόνο μία φορά. Τέλος, κάτι πολύ σημαντικό σε σχέση με την αντιπαράσταση των αντικειμένων είναι ότι όπως έχουμε πει παραπάνω ένα μερικώς διατεταγμένο σύνολο μπορεί να αναπαρασταθεί με ένα διάγραμμα Hasse. Ένα σύνολο αντικειμένων, που αναπαρίσταται με κανονικές μορφές και έχει μόνο ένα γονιό για κάθε κόμβο, μπορεί να αναπαρασταθεί με ένα δέντρο προθέματος ή αλλιώς trie[81].



Σχήμα 5.2: Αναπαράσταση ενός δέντρου προθέματος ή trie

Η δομή ενός trie σε σχέση με την αναπαράσταση ενός συχνού συνόλου αντικειμένων μπορεί να γίνει με πολλούς τρόπους όπως για παράδειγμα ανάλογα με την υποστήριξή τους. Η εύρεση ενός συχνού συνόλου αντικειμένων είναι ισοδύναμη με τη διάσχιση ενός trie και στη συνέχεια της εργασίας θα προταθούν αλγόριθμοι που προτιμούν αυτή τη τεχνική αλλά και βελτιώσεις του αλγορίθμου Apriori για την καλύτερη δημιουργία συνόλων συχνών αντικειμένων.

5.7 Βελτιώσεις στον αλγόριθμο Apriori

Με τις έννοιες που ορίστηκαν στις προηγούμενες ενότητες πλέον μπορούμε να δώσουμε κάποιες βελτιστοποιήσεις για τον αλγόριθμο Apriori, αφού πρώτα εξετάσουμε κάποιες από τις βασικές ιδέες του πιο αναλυτικά. Πρακτικά, ο αλγόριθμος Apriori εξετάζει σύνολα αντικειμένων αυξανόμενου μεγέθους, ή με μία διαφορετική αναπαράσταση, διασχίζει κατά πλάτος το δέντρο

προθέματος που φτιάχνεται από τις συναλλαγές μιας βάσης δεδομένων. Χρησιμοποιώντας την κανονική μορφή των αντικειμένων και οργανώνοντάς τα σε ένα trie, μπορούμε να εξασφαλίσουμε ότι κάθε πιθανό σύνολο συχνών αντικειμένων θα παραχθεί μόνο μία φορά και δεν θα παραχθούν όλες οι αναδιατάξεις του. Κάθε φορά που παράγεται ένα υποψήφιο υποσύνολο, ελέγχεται αν η υποστήριξή του είναι μεγαλύτερη από την ελάχιστη υποστήριξη, και αν δεν είναι τότε το απορρίπτουμε. Οι συναλλαγές αναπαρίστανται σαν πίνακες αντικειμένων. Έχουν προταθεί διάφορες βελτιστοποιήσεις στον αλγόριθμο Apriori που έχουν να κάνουν όχι με την βασική του ιδέα, αλλά με τις δομές δεδομένων και την αναπαράστασή τους[76]. Για παράδειγμα οι κόμβοι του δέντρου προθέματος έχουν αναπαρασταθεί σαν ταξινομημένα διανύσματα και πίνακες κατακερματισμού. Επίσης έχει γίνει μελέτη του τρόπου που θα κωδικοποιηθούν τα αντικείμενα και του τρόπου που θα ομαδοποιηθούν αυτά (πχ ανάλογα με τη συχνότητά τους). Συνοψίζοντας στον αλγόριθμο, θα μπορούσαμε να πούμε ότι:

- Διασχίζει κατά πλάτος ένα μερικώς διατεταγμένο σύνολο και συγκεκριμένα το $(2^B, \subseteq)$ όπου B το σύνολο των αντικειμένων που περιέχονται στις συναλλαγές.

- Τα υποψήφια συχνά σύνολα δημιουργούνται ενώνοντας σύνολα που διαφέρουν κατά ένα αντικείμενο

- Η μέτρηση της υποστήριξης μπορεί να γίνει με τη χρήση ενός αναδρομικού αλγορίθμου.

Στα πλεονεκτήματά του συγκαταλέγεται, η άμεση απόρριψη των συνόλων αντικειμένων με χαμηλή υποστήριξη και στα μειονεκτήματα του θα μπορούσαμε να συμπεριλάβουμε την υψηλή απαίτησή του σε μνήμη και ότι ο υπολογισμός της υποστήριξης έχει μεγάλο υπολογιστικό κόστος για βάσεις δεδομένων με πολλές συναλλαγές.

5.8 Αναπαράσταση συναλλαγών

Πριν προχωρήσουμε σε αλγορίθμους οι οποίοι είναι υπολογιστικά καλύτεροι από τον Apriori, θα πρέπει να δώσουμε κάποιες απαραίτητες έννοιες για την καλύτερη κατανόηση αυτών των αλγορίθμων.

Η αναπαράσταση των συναλλαγών είναι μία από τις βασικότερες παραμέτρους στους αλγορίθμους που βρίσκουν συχνά σύνολα αντικειμένων [76]. Υπάρχουν δύο τρόποι αναπαράστασης και συγκεκριμένα η οριζόντια αναπαράσταση, που χρησιμοποιείται από αλγορίθμους όπως ο Apriori και η κάθετη αναπαράσταση, που χρησιμοποιείται από αλγορίθμους όπως ο Eclat. Στην οριζόντια αναπαράσταση, κάθε συναλλαγή είναι ένας πίνακας από τα αντικείμενα που την περιέχουν, και όπως έχουμε αναφέρει και σε προηγούμενες παραγράφους, ένα δέντρο προθέματος είναι μία "συμπυκνωμένη" οριζόντια αναπαράσταση, εφόσον οι συναλλαγές που έχουν ίδια προθέματα συγχωνεύονται. Στην κάθετη αναπαράσταση, για κάθε αντικείμενο δημιουργείται μια λίστα συναλλαγών. Ως λίστα συναλλαγών του αντικειμένου a, είναι οι συναλλαγές που το περιέχουν ή αλλιώς η κάλυψη του αντικειμένου a, $K_T(\{a\})$. Με αυτό τον τρόπο αν θέλουμε να βρούμε τις συναλλαγές που περιέχουν δύο αντικείμενα αρκεί να συγχωνεύσουμε τις κάθετες αναπαραστάσεις των δύο αντικειμένων, κρατώντας μόνο τα κοινά τους στοιχεία.

5.9 Αλγόριθμος Eclat

Ο αλγόριθμος Eclat [82], αξιοποιεί όλες τις έννοιες που αναφέραμε προηγουμένως προκειμένου να παράξει γρήγορα συχνά σύνολα αντικειμένων. Κατ' αρχάς τα σύνολα αντικειμένων προσπελάζονται με λεξικογραφική σειρά, δηλαδή έχουμε μία κατά βάθος προσπέλαση του δέντρου προθέματος. Μπορεί να παράγονται περισσότεροι συνδυασμοί από τον αλγόριθμο Apriori, γιατί δεν κρατιούνται οι βαθμοί υποστήριξης κάθε συνόλου αντικειμένων και επομένως το "κλάδεμα" δεν μπορεί να γίνει σε μεταγενέστερο στάδιο. Λόγω της κάθετης αναπαράστασης που χρησιμοποιεί ο αλγόριθμος δεν υπολογίζεται σε κάθε βήμα η υποστήριξη κάτι που μειώνει πολύ το υπολογιστικό κόστος και τη μνήμη που χρησιμοποιείται.

Ουσιαστικά ο αλγόριθμος λειτουργεί ως εξής: Αρχικά για κάθε συχνό μονοσύνολο i δημιουργείται ένα σύνολο D_i. Στη συνέχεια, για κάθε αντικείμενο j που διατηρεί τη λεξικογραφική σειρά, βρίσκεται η υποστήριξη της ένωσης των δύο συνόλων συνενώνοντας την κάλυψη των δύο αντικειμένων. Αν το νέο σύνολο $\{i, j\}$ έχει την απαιτούμενη υποστήριξη, τότε το j εισέρχεται στο i, μαζί με τη συγχωνευμένη υποστήριξη. Ο αλγόριθμος καλείται αναδρομικά για το νέο σύνολο D_i μέχρι να εξαντληθεί η λεξικογραφική σειρά. Ο αλγόριθμος Eclat, παράγει τα υποψήφια σύνολα χρησιμοποιώντας μόνο την τεχνική συνένωσης του Apriori. Η λεξικογραφική σειρά χρησιμοποιείται προκειμένου να μειωθεί ο αριθμός των υποψήφιων συνόλων που δημιουργείται, κάτι που έχει ως αποτέλεσμα τη μείωση των ενώσεων που πρέπει να γίνουν. Ο αλγόριθμος δεν εκμεταλλεύεται την ιδιότητα της μονοτονικότητας, αλλά ενώνει δύο υποσύνολα σε ένα υποψήφιο σύνολο, με αποτέλεσμα να παράγονται σχεδόν όλα τα δυνατά υποσύνολα (αλλά μόνο από ένα γονέα) και επομένως ο χώρος καταστάσεων να είναι πολύ μεγαλύτερος από αυτόν του Apriori.

Ακολουθεί ο ψευδοκώδικας του αλγορίθμου Eclat όπου ορίζουμε ως $\mathcal{D}$ τη βάση συναλλαγών, ως σ την ελάχιστη υποστήριξη, ως $\mathcal{I}$ το σύνολο όλων των αντικειμένων, ως $\mathcal{D}^i$ την i-οστή προβολή της βάσης συναλλαγών και ως $\mathcal{F}[I](\mathcal{D}, \sigma)$ το σύνολο όλων των k-συχνών συνόλων αντικειμένων με το ίδιο $k-1$ πρόθεμα $I \subseteq \mathcal{I}$.

Αλγόριθμος 7 Ψευδοκώδικας Eclat

```
function ECLAT(D, σ, I ⊆ I)
    F[I] = {}
    for all i ∈ I occuring in D do
        F[I] = F[I] ∪ {I ∪ {i}}
        // Create D^i
        D^i = {}
        for all j ∈ I occuring in D such that j > i do
            C = cover({i}) ∩ cover({j})
            if |C| ≥ σ then
                D^i = D^i ∪ {(j, C)}
            end if
        end for
        //Depth-first recursion
        Compute F[I ∪ {i}](D^i, σ)
        F = F[I] ∪ F[I ∪ {i}]
    end for
    return F[I](D, σ)
end function
```

5.10 Αλγόριθμος SaM

Ο αλγόριθμος SaM είναι ένας σχετικά απλός αλγόριθμος που χρησιμοποιεί οριζόντια αναπαράσταση καθώς και μία απλή τακτική "διαίρει και βασίλευε". Προτάθηκε από τον Borgelt [83] και υπολογίζει μια υπό συνθήκη βάση δεδομένων, την επεξεργάζεται αναδρομικά και τέλος αφαιρεί κάποια αντικείμενα από την αρχική υπό συνθήκη βάση. Η διαδικασία του αλγορίθμου είναι η ακόλουθη:

- Αρχικά, από τη βάση των συναλλαγών υπολογίζεται η συχνότητα των αντικειμένων και τα μη-συχνά αντικείμενα εξαλείφονται σε αυτό το βήμα.

- Στο επόμενο βήμα, τα αντικείμενα σε κάθε συναλλαγή ταξινομούνται ανάλογα με τη συχνότητά τους στις συναλλαγές, καθώς είναι γνωστό ότι η επεξεργασία των αντικειμένων σε αύξουσα συχνότητα, οδηγεί σε μειωμένο χρόνο εκτέλεσης.

- Στη συνέχεια οι συναλλαγές ταξινομούνται λεξικογραφικά σε φθίνουσα σειρά.

- Έπειτα "χτίζεται" η υπό συνθήκη βάση δεδομένων του αλγορίθμου στην οποία συνδυάζονται οι ίδιες συναλλαγές και αρχικοποιείται ένας πίνακας στον οποίο κάθε στοιχείο αποτελείται από δύο πεδία: έναν μετρητή των φορών που υπάρχει αυτή η συναλλαγή στη βάση και έναν δείκτη στην ταξινομημένη συναλλαγή. Αυτή η δομή δεδομένων επεξεργάζεται αναδρομικά για την εύρεση των συνόλων συχνών αντικειμένων.

- Εφαρμόζεται διαίρεση του πίνακα με τον εξής τρόπο: Όλα τα στοιχεία του πίνακα που ξεκινούν από το ίδιο αντικείμενο, μεταφέρονται σε ένα νέο πίνακα, στον οποίο το αρχικό αντικείμενο αφαιρείται και προστίθεται ένας δείκτης από τον αρχικό πίνακα προς το νέο πίνακα. Ο νέος πίνακας είναι η υπό συνθήκη βάση δεδομένων ενός νέου υποπροβλήματος και στη συνέχεια επεξεργάζεται αναδρομικά προκειμένου να βρεθούν όλα τα σύνολα συχνών αντικειμένων που θα περιέχουν το αρχικό αντικείμενο.

- Η υπό συνθήκη βάση δεδομένων για τα αντικείμενα που δεν περιέχουν το αρχικό αντικείμενο βρίσκεται με μία απλή συγχώνευση των συναλλαγών που δεν περιέχουν το αρχικό αντικείμενο από την αρχική βάση δεδομένων και του υπό συνθήκη πίνακα που φτιάχτηκε στο προηγούμενο βήμα. Η συγχώνευση είναι σχεδόν πανομοιότυπη με τον αλγόριθμο mergesort, με τη μόνη διαφορά ότι οι ίδιες συναλλαγές συνδυάζονται και απλά αυξάνεται ο δείκτης που δείχνει πόσες φορές υπάρχει αυτή η συναλλαγή.

- Σε κάθε βήμα διαίρεσης ελέγχεται η υποστήριξη, η οποία υπολογίζεται ως το άθροισμα των μετρητών των στοιχείων της "διαίρεσης", προκειμένου να μην συμπεριληφθούν μη συχνά σύνολα. Αν η υποστήριξη είναι πάνω από το κατώφλι προστίθεται το αντικείμενο και ο αλγόριθμος εκτελείται αναδρομικά μέχρι να είναι κενά τόσο το σύνολο που περιέχει το πρώτο στοιχείο και στο οποίο θα γίνει η διαίρεση, όσο και το σύνολο που δεν περιέχει το πρώτο στοιχείο.

Ακολουθεί ο ψευδοκώδικας του αλγορίθμου:

Αλγόριθμος 8 Ψευδοκώδικας αλγορίθμου SaM

```
function SaM(a : array of transactions, p : set of items, s_min)
    n ← 0
    while a ≠ ∅ do
        b ← ∅
        s ← 0
        i ← a[0].items[0]
        while a ≠ ∅ and a[0].items[0] = i do
            s ← s + a[0].wgt
            remove i from a[0].items
            if a[0].items ≠ ∅ then
                remove a[0] from a and append it to b
            else
                remove a[0] from a
            end if
        end while
        c ← b
        d ← ∅
        while a ≠  and b ≠ do
            if a[0].items > b[0].items then
                remove a[0] from a and append it to d
            else if a[0].items < b[0].items then
                remove b[0] from b and append it to d
            else
                b[0].wgt ← b[0].wgt + a[0].wgt
                remove b[0] from b and append it to d
                remove a[0] from a
            end if
        end while
        while a ≠ do
            remove a[0] from a and append it to d
        end while
        while b ≠ do
            remove b[0] from a and append it to d
        end while
        a ← d
        if s ≥ s_min then
            p ← p ∪ {i}
            report p with support s
            n ← n + 1 + SaM(c, p, s_min)
            p ← p − {i}
        end if
    end while
end function
```

5.11 Αλγόριθμος FP-Growth

Σε αυτή την παράγραφο θα παρουσιάσουμε τον αλγόριθμο FP-Growth, ο οποίος κωδικοποιεί τα δεδομένα σε μία δομή που ονομάζεται FP-tree και στη συνέχεια το διασχίζει για να εξάγει τα σύνολα συχνών αντικειμένων.

5.11.1 FP-Tree

Θα μπορούσαμε να πούμε ότι ένα FP-tree είναι ένας συνδυασμός κάθετης και οριζόντιας αναπαράστασης [79]. Αρχικά οι συναλλαγές ταξινομούνται με λεξικογραφική σειρά. Στη συνέχεια φτιάχνεται το FP-tree σε δύο περάσματα της βάσης των συναλλαγών. Κατά το πρώτο πέρασμα κάθε συναλλαγή μετατρέπεται σε ένα μονοπάτι στο FP-tree. Επειδή οι συναλλαγές μπορεί να έχουν κοινά αντικείμενα, τα μονοπάτια έχουν επικάλυψη μεταξύ τους. Όσο περισσότερο επικαλυπτόμενες είναι οι συναλλαγές τόσο, περισσότερο συμπυκνωμένο είναι το FP-tree. Αν το μέγεθος του FP-Tree είναι αρκετά μικρό ώστε να χωράει στη μνήμη, τότε θα μπορέσουμε να εξάγουμε σύνολα συχνών αντικειμένων πολύ γρήγορα.

Στη συνέχεια θα παρουσιάσουμε το πώς κατασκευάζεται ένα FP-Tree από ένα πίνακα συναλλαγών που κάθε γραμμή περιέχει τα αντικείμενα μιας συναλλαγής και ένα μοναδικό αναγνωριστικό (tid). Αρχικά, κάνουμε ένα πέρασμα σε όλα τα δεδομένα για να υπολογίσουμε την υποστήριξη κάθε αντικειμένου. Τα μη-συχνά αντικείμενα αφαιρούνται, ενώ τα συχνά ταξινομούνται σε φθίνουσα σειρά. Στο επόμενο πέρασμα για κάθε συναλλαγή με συχνά αντικείμενα φτιάχνεται ένα μονοπάτι στο FP-Tree. Σε κάθε κόμβο διατηρούμε ένα μετρητή ο οποίος αυξάνεται μόνο όταν δύο συναλλαγές έχουν κοινό πρόθεμα ή είναι ίδιες. Συνεχίζουμε αυτή τη διαδικασία μέχρι να κωδικοποιήσουμε όλες τις συναλλαγές. Σε περίπτωση επικαλυπτόμενων συναλλαγών αυξάνουμε τους μετρητές στα κοινά προθέματα και προσαρτούμε το μη-κοινό μέρος στο τέλος του μονοπατιού.

Το μέγεθος ενός FP-Tree είναι στις περισσότερες περιπτώσεις μικρότερο από το μέγεθος των δεδομένων στην αρχική τους μορφή, επειδή υπάρχουν κοινά αντικείμενα σε κάθε συναλλαγή. Στην καλύτερη περίπτωση, δηλαδή όταν όλες οι συναλλαγές είναι ίδιες, το FP-Tree αποτελείται μόνο από ένα κλαδί. Στη χειρότερη περίπτωση, δηλαδή όταν όλες οι συναλλαγές είναι διαφορετικές και δεν μοιράζονται κοινά προθέματα, το FP-Tree αποτελείται από όσα κλαδιά είναι ο αριθμός των συναλλαγών. Το μέγεθος ενός FP-Tree επίσης, εξαρτάται από πώς έχουν διαταχθεί τα αντικείμενα (σε αύξουσα ή φθίνουσα σειρά σε σχέση με την υποστήριξή τους). Τέλος, τα ίδια αντικείμενα σε διαφορετικά κλαδιά, συνδέονται με δείκτες μεταξύ τους.

5.11.2 Εύρεση συχνών συνόλων αντικειμένων στον αλγόριθμο FP-Growth

Ο αλγόριθμος FP-Growth[84] διασχίζει το FP-Tree από κάτω προς τα πάνω, δηλαδή κοιτάει τις καταλήξεις αντί για τα προθέματα. Αν θέλουμε να βρούμε όλα τα συχνά σύνολα που τελειώνουν σε ένα συγκεκριμένο αντικείμενο αρκεί να βρούμε στο πιο αριστερό κλαδί το αντικείμενο και στη συνέχεια να ακολουθήσουμε τους δείκτες. Η διαδικασία αυτή συνεχίζεται μέχρι να βρούμε για όλα τα αντικείμενα όλα τα συχνά σύνολα τα οποία τελειώνουν σε αυτά. Στη συνέχεια, ο αλγόριθμος εφαρμόζει μια τακτική "διαίρει και βασίλευε" για να σπάσει το πρόβλημα σε μικρότερα υποπροβλήματα.

Για παράδειγμα ας δούμε την διαδικασία που θα ακολουθούσαμε με τον αλγόριθμο FP-Growth για την εύρεση όλων των συχνών συνόλων που τελειώνουν στο αντικείμενο x.

1. Αρχικά βρίσκουμε όλα τα μονοπάτια που περιέχουν το x. Αυτά τα αρχικά μονοπάτια, ονομάζονται μονοπάτια προθεμάτων.

2. Προσθέτοντας, την υποστήριξη όλων των κόμβων που έχουν την ταμπέλα του x, βρίσκουμε την υποστήριξη του x.

3. Σε περίπτωση που το x είναι συχνό ο αλγόριθμος πρέπει να λύσει όλα τα υποπροβλήματα τα οποία τελειώνουν σε x, δηλαδή να βρεθούν όλα τα σύνολα δύο στοιχείων που τελειώνουν σε x. Για να γίνει αυτό όμως θα πρέπει να μετατρέψουμε τα μονοπάτια προθεμάτων σε ένα υπό συνθήκη FP-Tree, το οποίο μοιάζει με ένα FP-Tree, με τη διαφορά ότι χρησιμοποιείται για να βρεθούν όλα τα συχνά σύνολα αντικειμένων που τελειώνουν με μία συγκεκριμένη κατάληξη. Ένα υπό συνθήκη FP-Tree φτιάχνεται ως εξής:

   (a) Αρχικά στα μονοπάτια προθεμάτων ανανεώνουμε τους μετρητές έτσι ώστε να ανακλούν τον πραγματικό αριθμό συναλλαγών που περιέχουν το αντικείμενο x.

   (b) Στη συνέχεια αφαιρούμε όλους τους κόμβους x, γιατί η πληροφορία που χρειαζόμαστε βρίσκεται στους παραπάνω ανανεωμένους κόμβους και μετρητές.

   (c) Στους ανανεωμένους μετρητές, ελέγχουμε ότι κάθε κόμβος είναι συχνός και σε περίπτωση που δεν είναι τότε αφαιρούμε όλο το μονοπάτι από εκείνο τον κόμβο και κάτω.

4. Στη συνέχεια, ο αλγόριθμος φτιάχνει τα υπό συνθήκη FP-Trees προκειμένου να βρει όλα τα σύνολα δύο αντικειμένων που τελειώνουν σε x, επαναλαμβάνοντας τη διαδικασία αναδρομικά από το βήμα 1.

Το παραπάνω παράδειγμα, δείχνει την αρχή του "διαίρει και βασίλευε" που χρησιμοποιεί ο αλγόριθμος FP-Growth. Σε κάθε αναδρομικό βήμα, φτιάχνεται ένα υπό συνθήκη FP-Tree ανανεώνοντας τους μετρητές κατά μήκος των μονοπατιών προθεμάτων και αφαιρώντας όλα τα μη-συχνά αντικείμενα. Επειδή τα υποπροβλήματα είναι ξένα μεταξύ τους, ο αλγόριθμος FP-Growth δεν θα παράξει διπλότυπα σύνολα αντικειμένων. Τέλος, οι μετρητές που υπάρχουν σε κάθε κόμβο, επιτρέπουν στον αλγόριθμο να βρει την υποστήριξη κάθε συνόλου ενώ παράγει σύνολα με κοινά προθέματα.

Ακολουθεί ο ψευδοκώδικας του αλγορίθμου FP-Growth όπου ορίζουμε ως $\mathcal{D}$ τη βάση συνδιαλλαγών, ως σ την ελάχιστη υποστήριξη, ως $\mathcal{I}$ το σύνολο όλων των αντικειμένων, ως $\mathcal{D}^i$ την i-οστή προβολή της βάσης συναλλαγών και ως $\mathcal{F}[I](\mathcal{D}, \sigma)$ το σύνολο όλων των k-συχνών συνόλων αντικειμένων με το ίδιο $k-1$ πρόθεμα $I \subseteq \mathcal{I}$.

Αλγόριθμος 9 Ψευδοκώδικας FP-Growth

function FP-Growth($\mathcal{D}, \sigma, I \subseteq \mathcal{I}$)
  $\mathcal{F}[I] = \{\}$
  for all $i \in \mathcal{I}$ occuring in $\mathcal{D}$ **do**
    $\mathcal{F}[I] = \mathcal{F}[I] \cup \{I \cup \{i\}\}$
    // Create $\mathcal{D}^i$
    $\mathcal{D}^i = \{\}$
    $H = \{\}$
    for all $j \in \mathcal{I}$ occuring in $\mathcal{D}$ such that $j > i$ **do**
      if $support(I \cup \{i, j\}) \geq \sigma$ **then**
        $H = H \cup \{j\}$
      end if
    end for
    for all $(tid, X) \in \mathcal{D}$ with $i \in X$ **do**
      $\mathcal{D}^i = \mathcal{D}^i \cup \{(tid, X \cap H)\}$
    end for
    //Depth-first recursion
    Compute $\mathcal{F}[I \cup \{i\}](\mathcal{D}^i, \sigma)$
    $\mathcal{F} = \mathcal{F}[I] \cup \mathcal{F}[I \cup \{i\}]$
  end for
  return $\mathcal{F}[I](\mathcal{D}, \sigma)$
end function

5.12 Παραγωγή κανόνων συσχέτισης

Ως τώρα σε αυτό το κεφάλαιο, περιγράψαμε τους τρόπους, τις έννοιες και τους αλγορίθμους για την εξαγωγή συνόλων συχνών αντικειμένων. Όμως σε κάποιες εφαρμογές, είναι απαραίτητη η εξαγωγή και κανόνων συσχέτισης. Η εξαγωγή κανόνων συσχέτισης από συχνά σύνολα αντικειμένων είναι σχετικά απλή, και σε αυτή την παράγραφο θα περιγράψουμε τη διαδικασία με την οποία μπορεί να γίνει. Προκειμένου να εξάγουμε κανόνες για κάθε συχνό σύνολο f, χρησιμοποιούμε όλα τα μη-κενά υποσύνολα του f, δηλαδή το δυναμοσύνολό του. Για κάθε υποσύνολο a, εξάγουμε έναν κανόνα της μορφής $(f - a) \rightarrow a$ αν και μόνο αν:

$$\text{εμπιστοσύνη } = \frac{f.count}{(f-a).count} \geq \text{κατώφλι εμπιστοσύνης} \tag{5.12}$$

όπου $f.count$ και $(f-a).count$ είναι η υποστήριξη του f και του $f-a$ αντίστοιχα. Η υποστήριξη του κανόνα είναι $\frac{f.count}{n}$ όπου n είναι ο αριθμός των συναλλαγών στο σύνολο των συναλλαγών T. Επειδή το f είναι ένα συχνό σύνολο αντικειμένων, η υποστήριξη όλων των υποσυνόλων του έχει παραχθεί με κάποιο τρόπο κατά την εξαγωγή συχνών συνόλων αντικειμένων, επομένως δεν χρειάζεται να επεξεργαστούμε ξανά τα δεδομένα κατά την εξαγωγή κανόνων συσχέτισης.

Η παραπάνω εξαντλητική μέθοδος εύρεσης κανόνων είναι ανέφικτη λόγω της πολυπλοκότητάς της. Για να μειωθεί αυτή πρέπει να σκεφτούμε ότι η υποστήριξη του συνόλου f στον παραπάνω υπολογισμό δεν αλλάζει καθώς αλλάζει το σύνολο a. Δηλαδή για κάθε κανόνα $(f - a) \rightarrow a$ που ισχύει, ισχύουν και όλοι οι κανόνες της μορφής $(f - a_{sub}) \rightarrow a_{sub}$, όπου a_{sub} είναι κάθε μη-κενό υποσύνολο του a, επειδή η υποστήριξη του $(f - a_{sub})$ πρέπει να είναι μικρότερη ή ίση από του $(f - a)$.

Επομένως, για κάθε συχνό σύνολο αντικειμένων f, αν ένας κανόνας με συνέπεια το a ισχύει, τότε ισχύουν και όλοι οι κανόνες με συνέπειες υποσύνολα του a. Με βάση αυτή την ιδιότητα, μπορούμε να παράγουμε κανόνες που έχουν ως συνέπεια μόνο ένα στοιχείο αρχικά. Στη συνέχεια, χρησιμοποιούμε τις συνέπειες αυτών των κανόνων για να παράξουμε όλα τις πιθανές συνέπειες με δύο στοιχεία που μπορούν να εμφανιστούν σε έναν κανόνα. Ακολουθεί ο ψευδοκώδικας αυτού του αλγορίθμου([6]). Ως F ορίζουμε το σύνολο όλων των συχνών συνόλων αντικειμένων, ως H_m το σύνολο που αποτελείται από τις συνέπειες με m στοιχεία και ως n τον συνολικό αριθμό συναλλαγών στο T.

Αλγόριθμος 10 Ψευδοκώδικας αλγορίθμου παραγωγής κανόνων συσχέτισης

function genRules(F)
  for each frequent k-itemset f_k in F, $k \geq 2$ **do**
    output every 1-item consequent rule of f_k with conf $\geq$ minconf and support $\leftarrow \frac{f_k.count}{n}$
    $H_1 \leftarrow$ {consequents of all 1-item consequent rules derived from f_k above}
    ap-genRules(f_k, H_1)
  end for
end function
function ap-genRules(f_k, H_m)
  if $(k > m + 1)$ and $H_m \neq \emptyset$ **then**
    for each h_{m+1} in H_{m+1} **do**
      conf $\leftarrow \frac{(f_k).count}{(f_k - h_{m+1}).count}$
      if conf $\geq$ minconf **then**
        output the rule $(f_k - h_{m+1}) \rightarrow h_{m+1}$ with confidence = conf
        and support = $\frac{f_k.count}{n}$
      else
        delete h_{m+1} from H_{m+1}
      end if
    end for
    ap-genRules(f_k, H_{m+1})
  end if
end function

5.13 Μετρήσεις-Αποτελέσματα

Σε αυτό το κομμάτι του κεφαλαίου θα παρουσιάσουμε τα αποτελέσματα της εργασίας μας από την εξόρυξη κανόνων συσχέτισης όπως αυτά εφαρμόστηκαν στα πραγματικά αρχεία καταγραφής του επίσημου ιστοτόπου της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών. Παρότι όλοι οι αλγόριθμοι που αναφέραμε στις προηγούμενες παραγράφους θα παράξουν τα ίδια αποτελέσματα σε ένα σύνολο συναλλαγών για λόγους πληρότητας έγιναν πειράματα με όλους, προκειμένου να επιβεβαιωθεί η παραπάνω υπόθεση[1]. Στη συνέχεια παρουσιάζουμε την διαδικασία που ακολουθήσαμε προκειμένου να καθαρίσουμε το αρχείο καταγραφής του ECE από "άχρηστη" πληροφορία για να βγάλουμε όσο το δυνατόν πιο χρήσιμους κανόνες. Τα αρχεία καταγραφής που λήφθηκαν ήταν από τον Μάρτιο μέχρι και τον Ιούνιο του 2013 καθώς και για κάποιες μέρες του Δεκεμβρίου του 2015.

[1] Οι υλοποιήσεις των αλγορίθμων εξόρυξης κανόνων συσχέτισης και συχνών συνόλων αντικειμένων που χρησιμοποιήθηκαν στα πλαίσια αυτής της διπλωματικής ανήκουν στο [85].

Εξαγωγή Κανόνων Συσχέτισης από τα αρχεία καταγραφής του ECE

1. Αρχικά καθαρίσαμε τα αρχεία καταγραφής από εικόνες, αρχεία μορφοποίησης και scripts κρατώντας μόνο τα "χρήσιμα" links.
2. Παρατηρώντας το πεδίο του User Agent, βρήκαμε τα bots που υπήρχαν και αφαιρέσαμε όλα τα αιτήματά τους.
3. Αναγνωρίσαμε κάθε χρήστη χρησιμοποιώντας το διάνυσμα *(IP, Λειτουργικό Σύστημα, Έκδοση Περιηγητή Ιστού)*.
4. Για κάθε μοναδικό χρήστη χωρίσαμε τα sessions του χρησιμοποιώντας κανόνα μισαώρου, δηλαδή σε περίπτωση που δύο αιτήματα απείχαν χρονικά περισσότερο από 30 λεπτά τότε θεωρούσαμε ότι το δεύτερο αίτημα αποτελεί την αρχή ενός νέου session, αλλά και την προτεινόμενη μέθοδο του πρώτου μέρους.
5. Πέρα από την παραπάνω μέθοδο πειραματιστήκαμε και ως εξής προκειμένου να εξάγουμε ουσιώδη αποτελέσματα:
   - Κρατήσαμε όσα sessions είχαν από κάποια αιτήματα και πάνω (από 3 έως 7 αιτήματα).
   - Χωρίσαμε τα sessions ανάλογα με το πόσες φορές είχαν πρόσβαση στην αρχική σελίδα του ECE.
   - Ενώσαμε όλα τα αρχεία καταγραφής αλλά πήραμε και το καθένα ξεχωριστά.
6. Στη συνέχεια τα sessions που εξάγαμε δόθηκαν ως είσοδος στους αλγορίθμου εξόρυξης κανόνων συσχέτισης με υποστήριξη ίση με 5% και εμπιστοσύνη ίση με 60%.

Από τους παραπάνω πειραματισμούς εξήχθησαν οι εξής κανόνες:

- Όλοι οι επισκέπτες του site της σχολής επισκέπτονται με πιθανότητα 100% την αρχική σελίδα, όπως ήταν φυσικό.
- Σε περιόδους που έχει βγει κάποιο μάθημα το οποίο δίνει μεγάλος αριθμός φοιτητών, αυξάνεται κατά πολύ η επισκεψιμότητα της σελίδας του αντίστοιχου καθηγητή. Ο παραπάνω συλλογισμός επιβεβαιώνεται από την επισκεψιμότητα της σελίδας του κύριου Μαράτου κατά την περίοδο 19-23 Δεκεμβρίου 2015 όπου αυτό γινόταν για όλους τους επισκέπτες της σελίδας.
- Πολύ μεγάλος αριθμός επισκεπτών χρησιμοποιεί την αναζήτηση του site.
- Όταν κάποιος φοιτητής βλέπει τη σελίδα κάποιου μαθήματος ροών βλέπει και κάποια από τα προαπαιτούμενα μαθήματα κορμού για αυτό το μάθημα, παρατήρηση που επιβεβαιώνεται από τα αιτήματα που έγιναν για την επεξεργασία φωνής και φυσικής γλώσσας και για τη θεωρία πιθανοτήτων και στατιστικής.
- Στην αρχή του εξαμήνου υπάρχει ιδιαίτερα αυξημένη κίνηση για τις σελίδες των μαθημάτων κάτι που παρατηρείται ιδιαίτερα για τα μαθήματα κορμού και επιβεβαιώνεται από την ζήτηση των σελίδων των μαθημάτων 2ου και 4ου εξαμήνου τον Μάρτιο του 2013.
- Τέλος, σελίδες διαδοχικών εξαμήνων προσπελάζονται συχνά μαζί, όπως για παράδειγμα του 2ου και του 3ου εξαμήνου τον Μάιο του 2013.

Παρατηρούμε ότι αρκετοί από τους παραπάνω κανόνες είναι προφανείς ή δεν μας δίνουν κάποια χρήσιμη πληροφορία. Κάτι τέτοιο οφείλεται σχεδόν αποκλειστικά στη μορφή των δεδομένων αλλά και στον ιστότοπο της ΣΗΜΜΥ εν γένει. Κατ' αρχάς πολύ μεγάλος αριθμός επισκεπτών επισκέπτεται μία ή δύο ιστοσελίδες και στην συνέχεια αποχωρεί. Είναι πολύ λίγα τα sessions που έχουν πάνω από πέντε επισκέψεις σελίδων. Κάτι που ενισχύει τα μικρά sessions είναι ότι οι περισσότεροι επισκέπτες ξέρουν ακριβώς τι θέλουν να ψάξουν με αποτέλεσμα να μην περιηγούνται στον ιστότοπο πέραν της σελίδας που θέλουν να βρουν.

Γενικότερα ιστοσελίδες κατά τις οποίες οι επισκέπτες γνωρίζουν σε μεγάλο βαθμό από πριν τι θέλουν να προσπελάσουν δεν είναι κατάλληλες για εξόρυξη κανόνων συσχέτισης. Οι ιστότοποι που χρησιμοποιείται κατά κόρον αυτού του είδους η ανάλυση είναι ηλεκτρονικά καταστήματα που ο επισκέπτης ακόμα και να γνωρίζει το προϊόν που θέλει να αγοράσει, μπορεί να περιηγείται για ώρα πριν βρει το μοντέλο καθώς και το είδος του προϊόντος. Σε αυτούς τους ιστοτόπους θα μπορούσαμε για παράδειγμα να βγάλουμε κανόνες της μορφής "Το 30% των χρηστών που ψάχνουν κινητά Samsung στη συνέχεια βλέπουν κινητά LG".

Κεφάλαιο 6

Επίλογος

Σε αυτό το κεφάλαιο θα συνοψιστούν τα συμπεράσματα αυτής της διπλωματικής εργασίας και θα παρουσιαστούν μερικές μελλοντικές επεκτάσεις της σε διάφορες κατευθύνσεις, κυρίως στο δεύτερο μέρος της.

6.1 Συμπεράσματα

Στο πρώτο μέρος της εργασίας έγινε μια εκτενής μελέτη σχετικά με την προεπεξεργασία των αρχείων καταγραφής. Μελετήθηκε η δομή τους, ο τρόπος καθαρισμού τους από web/crawlers/bots έτσι ώστε να έρθουν σε μία πιο "φιλική" μορφή για εφαρμογή αλγορίθμων Εξόρυξης Δεδομένων. Στη συνέχεια, έγινε μια πιο εκτενής μελέτη όσον αφορά την αναγνώριση χρηστών και συνεδριών. Ειδικότερα, μελετήθηκε ένας εξ' ολοκλήρου νέος τρόπος αναγνώρισης των συνεδριών ενός χρήστη. Στα πλαίσια αυτής της μελέτης, αναπτύχθηκε ένα σύστημα μηχανικής μάθησης το οποίο ρυθμίζει διάφορες παραμέτρους του αλγορίθμου ασαφούς συσταδοποίησης c-κέντρων προκειμένου να καταλήξει στον καταλληλότερο αριθμό συνεδριών σε σχέση με τα αιτήματα ενός χρήστη. Στο πλαίσιο της προτεινόμενης μεθοδολογίας, παρουσιάστηκαν εκτενή αποτελέσματα για 20 χρήστες που προήλθαν από αρχεία καταγραφής και συγκεκριμένα αυτά της διαδικτυακής κοινότητας των φοιτητών ΗΜΜΥ [22].

Στο δεύτερο μέρος της εργασίας πραγματοποιήθηκε μια μελέτη σχετικά με τους τρόπους που μπορούμε να εξορύξουμε χρήσιμα μοτίβα από τα προεπεξεργασμένα αρχεία καταγραφής. Στη συνέχεια, εξετάστηκε εκτενώς η εξόρυξη κανόνων συσχέτισης και το πώς αυτή μπορεί να χρησιμοποιηθεί προκειμένου να προβλεφθεί ο επόμενος σύνδεσμος στον οποίο θα μπορούσε να πλοηγηθεί ο χρήστης. Συγκεκριμένα, θεμελιώθηκε μαθηματικά το πρόβλημα των κανόνων συσχέτισης και μετέπειτα αναλύθηκαν τέσσερις αλγόριθμοι, οι Apriori, Eclat, SaM, FP-Growth. Μελετήθηκε το μαθηματικό και προγραμματιστικό υπόβαθρο και των τεσσάρων αλγορίθμων και εφαρμόστηκαν στα αρχεία καταγραφής της επίσημης ιστοσελίδας της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υολογιστών του Εθνικού Μετσόβιου Πολυτεχνείο [70] προκειμένου να εξαχθούν κανόνες συσχέτισης σχετικά με την πλοήγηση των χρηστών στη συγκεκριμένη ιστοσελίδα.

6.2 Μελλοντικές επεκτάσεις

Υπάρχουν αρκετοί τρόποι με τους οποίους θα μπορούσε να επεκταθεί η συγκεκριμένη διπλωματική εργασία. Κατ' αρχάς μπορούν να μελετηθούν και άλλοι τρόποι εξόρυξης χρήσιμων μοτίβων, όπως για παράδειγμα η συσταδοποίηση. Επίσης, στην εξόρυξη κανόνων συσχέτισης θα μπορούσε

να αναπτυχθεί ένα πραγματικό σύστημα κατά το οποίο να γίνεται προανάκληση μιας σελίδας προτού αυτή ζητηθεί σε σχέση με τον κανόνα συσχέτισης, ο οποίος ταιριάζει παραπάνω μέχρι εκείνη τη στιγμή στην πλοήγηση του χρήστη.

Επιπρόσθετα, θα μπορούσαν να μελετηθούν άλλοι αλγόριθμοι εξόρυξης κανόνων συσχέτισης όπως είναι οι RElim και LCM, αλλά και να γίνουν βελτιώσεις στους ήδη υπάρχοντες σε σχέση με τα δεδομένα στα οποία γίνεται η εξόρυξη δεδομένων. Άλλωστε, κάτι τέτοιο ήδη γίνεται τον τελευταίο καιρό[86]. Τέλος, θα ήταν ενδιαφέρουσα μια μελέτη και επέκταση των δομών δεδομένων που χρησιμοποιούν κλασσικοί αλγόριθμοι εξόρυξης κανόνων συσχέτισης, καθώς ήδη έχουν προταθεί αρκετές βελτιώσεις που οδηγούν σε καλύτερες επιδόσεις τόσο από άποψη χρόνου όσο και από άποψη μνήμης.

Βιβλιογραφία

[1] Jaideep Srivastava, Robert Cooley, Mukund Deshpande, and Pang-Ning Tan. Web usage mining: Discovery and applications of usage patterns from web data. *ACM SIGKDD Explorations Newsletter*, 1(2), January 2000.

[2] K-means clustering with apache mahout. http://blog.guillaumeagis.eu/k-means-clustering-apache-mahout/. [Online, accessed March 2016].

[3] Cluster visualization. http://cluviz.twoday.net/stories/687849/. [Online, accessed March 2016].

[4] Spiking neurons - subtractive clustering. http://www.spikingneurons.com/projects_subtractive-clustering. [Online, accessed March 2016].

[5] Data mining and electronic business. http://weigend.com/files/teaching/stanford/2008/stanford2008.wikispaces.com/6.html. [Online, accessed March 2016].

[6] Bing Liu. *Web Data Mining, Exploring Hyperlinks, Contents and Usage Data.* Springer, 2nd edition, 2011.

[7] Ian H. Witten, Eibe Frank, and Mark A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques.* Morgan Kaufmann Publishers Inc., 3rd edition, 2011.

[8] Zdravko Markov and Daniel T. Larose. *Data Mining the Web.* Wiley, 2007.

[9] About w3c. https://www.w3.org/. [Online, accessed February 2016].

[10] Common log format. https://www.w3.org/Daemon/User/Config/Logging.html#common-logfile-format. [Online, accessed March 2016].

[11] List of status codes. https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html. [Online, accessed February 2016].

[12] Hotlinking explanation. https://simple.wikipedia.org/wiki/Hotlinking. [Online, accessed February 2016].

[13] R. Cooley, B. Mobasher, and J. Srivastav. Data preparation for mining world wide web browsing patterns. *Journal of Knowledge and Data Engineering Workshop*, 1:5 – 32, 1999.

[14] Zidrina Pabarskaite and Airstis Raudys. A process of knowledge discovery from web usage data: Systemization and critical review. *Journal of Intelligent Information Systems*, 28(1):79 – 104, 2007.

[15] Mitali Srivastava, Rakhi Garg, and P.K. Mishra. Preprocessing techniques in web usage mining: A survey. *International Journal of Computer Applications*, 97(18), July 2014.

[16] Renata Ivancsy and Sandor Juhasz. Analysis of web user identification methods. *World Academy of Science, Engineering and Technology*, 34, 2007.

[17] M. Spiliopoulou, B. Mobasher, B. Berendt, and M. Nakagawa. Framework for the evaluation of session reconstruction heuristics in web usage analysis. *INFORMS Journal on Computing*, 15, 2003.

[18] Sheetal A. Raiyani, Shailendra Jain, and Ashwin G. Raiyani. Advanced preprocessing using distinct user identification in web log usage data. *International Journal of Advanced Research in Computer and Communication Engineering*, 1(6), August 2012.

[19] Pabarskaite Z. Decision trees for web log mining. *Intelligent Data Analysis Journal*, 7(2):141 – 155, 2003.

[20] Renata Ivancsy and Sandor Juhasz. Analysis of web user identification methods. *World Academy of Science Engineering and Technology*, 34, 2007.

[21] F. Facca and P. Lanzi. Mining interesting knowledge from weblogs: a survey. *Data and Knowledge Engineering*, 53(3):225 – 241, 2005.

[22] Διαδικτυακή Κοινότητα φοιτητών της σχολής HMMY. https://shmmy.ntua.gr/. [Online, accessed March 2016].

[23] L. Catledge and J. Pitkow. Characterizing browsing strategies in the world wide web. *Computer Networks and ISDN Systems*, 27:1065 – 1073, 1995.

[24] Nema Sharma and Pawan Makhija. Web usage mining: A novel approach for web user session construction. *Global Journal of Computer Science and Technology: Network, Web & Security*, 15(3), 2015.

[25] Robert Cooley, Bamshed Mobasher, and Jaideep Srinivastava. Web mining: Information and pattern discovery on the world wide web. *International conference on Tools with Artificial Intelligence*, pages 558 – 567, 1997.

[26] Mitali Srivastava, Rakhi Gang, and P.K. Mishra. Preprocessing techniques in web usage mining: A survey. *International Journal of Computer Applications*, 97(18), July 2014.

[27] He Xinhua and Wang Qiong. Dynamic timeout-based a session identification algorithm. 2011.

[28] M. Chen, A.S. LaPaugh, and J.P. Singh. Predicting category accesses for a user in a structured information space. *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 65 – 72, 2002.

[29] J. Zhang and Ali A. Ghorbani. The reconstruction of user session from a server log using improved time oriented heuristic. *2nd Annual Conference on Communication Networks and Service Research IEEE*, pages 315 – 322, 2004.

[30] V. Chitraa and Dr. Antony Selvdoss Davamani. A survey on preprocessing methods for web usage data. *International Journal of Computer Science and Information Security*, 7(3), 2010.

[31] Jose M. Domenech and Javier Lorenzo. A tool for web usage mining. *8th International Conference on Intelligent Data Engineering and Automated Learning*, 2007.

[32] Mona S. Kamat, Dr. J. W. Bakal, and Madhu Nashipudi. Improved data preparation technique in web usage mining. *International Journal of Computer Networks and Communications Security*, 1(7):284 – 291, 2013.

[33] Zahid Ansari, Syed Abdul Sattar, Waseem Ahmed, and Mohammad Fazle Azeem. Mountain density-based fuzzy approach for discovering web usage clusters from web log data. *Fuzzy Sets and Systems*, 279(15):40 – 63, November 2015.

[34] James C. Bezdek. Pattern recognition with fuzzy objective function algorithms. *Plenum Press, New York*, 1981.

[35] J. C. Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3:32 – 57, 1973.

[36] Timothy J. Ross. *Fuzzy Logic with Engineering Applications.* Wiley, 3rd edition, 2010.

[37] R. Yager and D. Filev. Generation of fuzzy rules by mountain clustering. *Journal of Intelligent & Fuzzy Systems*, 2(3):209 – 219, 1994.

[38] S. Chiu. Model identification based on cluster estimation. *Journal of Intelligent & Fuzzy Systems*, 2(3), September 1994.

[39] Weina Wang and Yunjie Zhang. On fuzzy cluster validity indices. *Fuzzy Sets and Systems*, 158:2095 – 2117, 2007.

[40] J.C. Bezdek. Cluster validity with fuzzy sets. *J. Cybernet*, 3:58 – 73, 1974.

[41] J.C. Bezdek. Numerical taxonomy with fuzzy sets. *J. Math. Biol*, 1:57 – 71, 1974.

[42] R.N. Dave. Validating fuzzy partition obtained through c-shells clustering. *Pattern Recognition Lett. 17*, pages 613 – 623, 1996.

[43] Y. Fukuyama and M. Sugeno. A new method of choosing the number of clusters for the fuzzy c-means method. *Proc. Fifth Fuzzy Systems Symp.*, pages 247 – 250, 1989.

[44] X.L. Xie and G. Beni. A validity measure for fuzzy clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13:841 – 847, 1991.

[45] J.C. Bezdek N.R. Pal. *IEEE Trans. Fuzzy Systems*, 3(3):370 – 379, On cluster validity for fuzzy c-means model.

[46] S.H. Kwon. Cluster validity index for fuzzy clustering. *Electron. Lett. 34*, 22:2176 – 2177, 1998.

[47] Z.Q. Sun Y.G. Tang, F.C. Sun. Improved validation index for fuzzy clustering. *American Control Conf.*, June 2005.

[48] K.L. Wu and M.S. Yang. A cluster validity index for fuzzy clustering. *Pattern Recognition Lett.*, 26:1275 – 1291, 2005.

[49] Khaled Hammouda and Fakhreddine Karray. A comparative study of data clustering techniques. *Fakhreddine Karray University of Waterloo, Ontario, Canada*, 2000.

[50] M. Rekha Sundari, Y. Srinivas, and PVGD. Prasad Reddy. A review on pattern discovery techniques of web usage mining. *Int. Journal of Engineering Research and Applications*, 4(9):131 – 136, September 2014.

[51] Geoffrey McLachlan and Thriyambakam Krishnan. *The EM Algorithm and Extensions*. Wiley, 2nd edition, 2008.

[52] Yinghui Yang and Balaji Padmanabhan. Ghic: A hierarchical pattern-based clustering algorithm for grouping web transactions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(9):1300–1304, 2005.

[53] Sophia G Petridou, Vassiliki A Koutsonikola, Athena I Vakali, and Georgios I Papadimitriou. Time-aware web users' clustering. *Knowledge and Data Engineering, IEEE Transactions on*, 20(5):653–667, 2008.

[54] GT Raju and MV Sudhamani. A novel approach for extraction of cluster patterns from web usage data and its performance analysis. In *Emerging Trends in Electrical and Computer Technology (ICETECT), 2011 International Conference on*, pages 718–723. IEEE, 2011.

[55] Pablo Loyola, Pablo E Román, and Juan D Velásquez. Clustering-based learning approach for ant colony optimization model to simulate web user behavior. In *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Volume 01*, pages 457–464. IEEE Computer Society, 2011.

[56] Sebastián A Ríos, Roberto A Silva, and Felipe Aguilera. A dissimilarity measure for automate moderation in online social networks. In *Proceedings of the 4th International Workshop on Web Intelligence & Communities*, page 3. ACM, 2012.

[57] Shi Zhong and Joydeep Ghosh. A unified framework for model-based clustering. *The Journal of Machine Learning Research*, 4:1001–1037, 2003.

[58] Dong-Ho Kim, Vijayalakshmi Atluri, Michael Bieber, Nabil Adam, and Yelena Yesha. A clickstream-based collaborative filtering personalization model: towards a better performance. In *Proceedings of the 6th annual ACM international workshop on Web information and data management*, pages 88–95. ACM, 2004.

[59] Yanzan Kevin Zhou and Bamshad Mobasher. Web user segmentation based on a mixture of factor analyzers. In *E-Commerce and Web Technologies*, pages 11–20. Springer, 2006.

[60] Aixin Sun, Ee-Peng Lim, and Wee-Keong Ng. Web classification using support vector machine. In *Proceedings of the 4th international workshop on Web information and data management*, pages 96–99. ACM, 2002.

[61] AK Santra and S Jayasudha. Classification of web log data to identify interested users using naïve bayesian classification. *International Journal of Computer Science Issues*, 9(1):381–387, 2012.

[62] Term frequency inverse document frequency. http://www.tfidf.com/. [Online, accessed February 2016].

[63] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186. ACM, 1994.

[64] Weiyang Lin, Sergio A Alvarez, and Carolina Ruiz. Efficient adaptive-support association rule mining for recommender systems. *Data mining and knowledge discovery*, 6(1):83–105, 2002.

[65] Koji Miyahara and Michael J Pazzani. Collaborative filtering with the simple bayesian classifier. In *PRICAI 2000 Topics in Artificial Intelligence*, pages 679–689. Springer, 2000.

[66] Miha Grčar, Blaž Fortuna, Dunja Mladenič, and Marko Grobelnik. knn versus svm in the collaborative filtering framework. In *Data Science and Classification*, pages 251–260. Springer, 2006.

[67] Zhonghang Xia, Yulin Dong, and Guangming Xing. Support vector machines for collaborative filtering. In *Proceedings of the 44th annual Southeast regional conference*, pages 169–174. ACM, 2006.

[68] Daniel T. Larose and Chantal D.Larose. *Discovering Knowledge in Data: An introduction to Data Mining*. Wiley, 2nd edition, 2014.

[69] Internet traffic archives. http://ita.ee.lbl.gov/html/traces.html. [Online, accessed February 2016].

[70] Επίσημος Ιστότοπος Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών. www.ece.ntua.gr. [Online, accessed March 2016].

[71] Phpbb. https://www.phpbb.com/. [Online, accessed March 2016].

[72] Bamshad Mobasher, Robert Cooley, and Jaideep Srivastava. Creating adaptive web sites through usage-based clustering of urls. In *Knowledge and Data Engineering Exchange, 1999.(KDEX'99) Proceedings. 1999 Workshop on*, pages 19–25. IEEE, 1999.

[73] Przemysław Kazienko. Mining indirect association rules for web recommendation. *International Journal of Applied Mathematics and Computer Science*, 19(1):165–186, 2009.

[74] Wang Yong, Li Zhanhuai, and Zhang Yang. Mining sequential association-rule for improving web document prediction. In *Computational Intelligence and Multimedia Applications, 2005. Sixth International Conference on*, pages 146–151. IEEE, 2005.

[75] Florent Masseglia, Maguelonne Teisseire, and Pascal Poncelet. Sequential pattern mining., 2009.

[76] Bart Goethals. Survey on frequent pattern mining. *Univ. of Helsinki*, 2003.

[77] Dan A Simovici and Chabane Djeraba. Mathematical tools for data mining. *Advanced Information and Knowledge Processing*, pages 129–172, 2008.

[78] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. *Proceedings of the 20th International Conference on Very Large Data Bases, VLDB*, pages 487–499, September 1994.

[79] Frequent pattern mining. http://www.borgelt.net/slides/fpm.pdf. [Online, accessed March 2016].

[80] Christian Borgelt. Canonical forms for frequent graph mining. In *Advances in Data Analysis*, pages 337–349. Springer, 2007.

[81] Amihood Amir, Ronen Feldman, and Reuven Kashi. A new and versatile method for association generation. In *Principles of Data Mining and Knowledge Discovery*, pages 221–231. Springer, 1997.

[82] M. J. Jaki, S.Parthasarathy, M. Ogihara, and W. Li. New algorithms for fast discovery of association rules. *KDD Proceedings*, 1997.

[83] C. Borgelt. Simple algorithms for frequent item set mining. *Advances in machine learning II - Springer*, 2010.

[84] Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. In *ACM Sigmod Record*, volume 29, pages 1–12. ACM, 2000.

[85] Software for frequent pattern mining. http://www.borgelt.net/software.html. [Online, accessed March 2016].

[86] Muhammad Shaheena, Muhammad Shahbazb, and Aziz Guergachic. Context based positive and negative spatio-temporal association rule mining. *Knowledge-Based Systems*, 37:261 – 273, January 2013.

[87] E. Trauwaert. On the meaning of dunn's partition coefficient for fuzzy clusters. *Fuzzy Sets and Systems*, 25:217 – 242, 1988.

[88] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining*. Addison-Wesley, 2005.