Διατριβή Μάστερ

ΑΛΓΟΡΙΘΜΟΙ ΕΝΗΜΕΡΩΣΗΣ ΠΡΟΦΙΛ ΣΥΣΤΗΜΑΤΩΝ ΕΞΑΤΟΜΙΚΕΥΣΗΣ ΓΙΑ ΚΙΝΗΤΟΥΣ ΧΡΗΣΤΕΣ

Μαρία Ανδρέου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ιούνιος 2009

Περίληψη

Με την ραγδαία ανάπτυξη του ασύρματου δικτύου και κατ' επέκταση της χρήσης των κινητών συσκευών, οι κινητοί χρήστες μπορούν να έχουν σήμερα πρόσβαση σε όλες τις υπηρεσίες διαδικτύου. Ωστόσο στο ασύρματο δίκτυο το πρόβλημα της πληροφοριακής υπερφόρτωσης είναι ακόμη πιο έντονο και η ανάγκη για λύση στο πρόβλημα αυτό, επιβεβλημένη.

Οι μέθοδοι εξατομίκευσης θεωρήθηκαν σαν η καλύτερη λύση στο πρόβλημα αυτό. Ωστόσο οι περισσότερες έρευνες που έχουν γίνει δεν αναφέρονται σε κινητούς χρήστες, ή αν το κάνουν αγνοούν το γεγονός ότι το ασύρματο δίκτυο επιτρέπει στους χρήστες να έχουν πρόσβαση σε πληροφορίες από οπουδήποτε και οποιονδήποτε στιγμή. Το γεγονός αυτό προσδίδει στο χρήστη κάποια ιδιαίτερα χαρακτηριστικά τα οποία δεν εμφανίζονταν μέχρι τώρα στους χρήστες του σταθερού περιβάλλοντος. Οι ανάγκες των κινητών χρηστών αλλάζουν σε σχέση με τον χρόνο και το χώρο στον οποίο βρίσκονται. Ο χρόνος, ο χώρος και η κατάσταση στην οποία βρίσκεται την στιγμή της αναζήτησης, ο χρήστης, επηρεάζει σημαντικά τόσο τις προτιμήσεις του όσο και τις επιλογές του. Κατ'επέκταση τα συστήματα εξατομίκευσης που αναφέρονται στο κινητό περιβάλλον είναι σημαντικό να λαμβάνουν υπόψη τους τρεις αυτούς άξονες.

Η παρούσα Διπλωματική εργασία εξειδίκευσης (ΔΕΕ), ασχολείται με την ανάλυση αλγορίθμων ενημέρωσης προφίλ χρηστών σε συστήματα εξατομίκευσης που αναφέρονται σε κινητά περιβάλλοντα. Η πρόκληση εδώ είναι ο τρόπος που ενημερώνουμε και επεξεργαζόμαστε το προφίλ του χρήστη ώστε να μπορέσουμε να αναπαραστήσουμε σε αυτό πληροφορίες που μας δίνουν στοιχεία για το πώς αλλάζουν οι προτιμήσεις του χρήστη καθώς κινείται. Σημαντικό είναι το πώς εντοπίζουμε τις αλλαγές στις ανάγκες του χρήστη καθώς αλλάζει δραστηριότητες, χώρο, τόπο και καταστάσεις κατά την διάρκεια της ημέρας. Πώς εντοπίζεται η αλλαγή αυτή στο χρόνο, τόπο και στις δραστηριότητες του χρήστη, πώς καταγράφεται και πως συνδέετε με τις αλλαγές στις προτιμήσεις του.

Μαρία Ανδρέου – Πανεπιστήμιο Κύπρου, 2009

ΑΛΓΟΡΙΘΜΟΙ ΕΝΗΜΕΡΩΣΗΣ ΠΡΟΦΙΛ ΣΥΣΤΗΜΑΤΩΝ ΕΞΑΤΟΜΙΚΕΥΣΗΣ ΓΙΑ ΚΙΝΗΤΟΥΣ ΧΡΗΣΤΕΣ

Μαρία Ανδρέου

Η Διατριβή αυτή

Υποβλήθηκε προς Μερική Εκπλήρωση των

Απαιτήσεων για την Απόκτηση

Τίτλου Σπουδών Master

σε Προηγμένες Τεχνολογίες Πληροφορικής

στο

Πανεπιστήμιο Κύπρου

Συστήνεται προς Αποδοχή

από το Τμήμα Πληροφορικής

Ιούνης, 2009

ΣΕΛΙΔΑ ΕΓΚΡΙΣΗΣ

Διατριβή Master

ΑΛΓΟΡΙΘΜΟΙ ΕΝΗΜΕΡΩΣΗΣ ΠΡΟΦΙΛ ΣΥΣΤΗΜΑΤΩΝ ΕΞΑΤΟΜΙΚΕΥΣΗΣ ΓΙΑ ΚΙΝΗΤΟΥΣ ΧΡΗΣΤΕΣ

Παρουσιάστηκε από

Μαρία Ανδρέου

Ερευνητικός Σύμβουλος _____

Όνομα Ερευνητικού Συμβούλου

Μέλος Επιτροπής _____

Όνομα Μέλους Επιτροπής

Μέλος Επιτροπής _____

Όνομα Μέλους Επιτροπής

Πανεπιστήμιο Κύπρου

Ιούνης, 2009

Ευχαριστίες

Με την ευκαιρία της ολοκλήρωσης της εργασίας αυτής, θα ήθελα να ευχαριστήσω των επιβλέπον καθηγητή μου, Δρ. Γιώργο Σαμάρα για την ευκαιρία που μου έδωσε να ασχοληθώ με το αντικείμενο της εξατομίκευσης καθώς και για τις πολύτιμες ιδέες του που ήταν και η βάση για την εργασία αυτή. Επιπλέον θα ήθελα να ευχαριστήσω τον βοηθό του, Χριστόφορο Παναγιώτου, για τη πολύτιμη και συνεχή καθοδήγηση του κατά την εκπόνηση της εργασίας αυτής.

Τέλος θα ήθελα να ευχαριστήσω την οικογένεια μου και τους φίλους μου για τη συνεχή βοήθεια και συμπαράσταση καθ' όλη την διάρκεια της εργασίας αυτής αλλά και για την ανοχή τους όλο αυτό το διάστημα.

Περιεχόμενα

Κεφάλαιο 1

Εισαγωγή

1.1 Γενικά

Η εργασία αυτή επικεντρώνεται στη μελέτη αλγορίθμων για ενημέρωση προφίλ χρηστών που χρησιμοποιούνται στην παροχή εξατομικευμένων υπηρεσιών στο χρήστη. Ωστόσο η παρούσα μελέτη, περιορίζεται σε κινητούς χρήστες, στους ο οποίους ο χώρος και ο χρόνος στον οποίο βρίσκονται επηρεάζει σημαντικά τις επιλογές τους.

Ο τρόπος διαχείρισης εξατομικευμένων υπηρεσιών που αναφέρονται σε κινητούς χρήστες διαφέρει αρκετά από τον αντίστοιχο για ένα σταθερό χρήστη. Στη περίπτωση ενός κινητού χρήστη υπάρχουν περιορισμοί, οι οποίοι επιβάλλονται κυρίως από το μέγεθος της κινητής συσκευής που χρησιμοποιείται και τις αδυναμίες του ασύρματου δικτύου. Κατ' επέκταση, το περιβάλλον με το οποίο αλληλεπιδρά ο χρήστης περιορίζει σημαντικά τις επιλογές του. Ως αποτέλεσμα, επιβάλλεται η ανάγκη για εμφάνιση στο χρήστη όσο το δυνατό πιο χρήσιμες για αυτόν υπηρεσιών. Η κίνηση γενικά χαρακτηρίζεται από δύο διαστάσεις, το χρόνο και το χώρο. Οι ανάγκες ενός κινητού χρήστη επηρεάζονται άμεσα από τις δύο αυτές διαστάσεις, αφού, καθώς κινείται, αλλάζουν οι πληροφορίες που πιθανόν να χρειάζεται.

Η παρούσα εργασία περιορίζεται στη μελέτη αλγορίθμων που λαμβάνουν υπόψη τα ιδιαίτερα αυτά χαρακτηριστικά του κινητού χρήστη, για να ενημερώσουν όσο πιο έγκαιρα και έγκυρα το προφίλ του χρήστη με απώτερο σκοπό την παρουσίαση όσο πιο χρήσιμων πληροφοριών σε αυτόν κατά την αλληλεπίδρασή του με το σύστημα.

1.2 Υποκίνηση έρευνας

Στις μέρες μας το Παγκόσμιο Πλέγμα Πληροφοριών (ΠΠΠ) παρουσιάζει μια συνεχή αύξηση του πληροφοριακού περιεχομένου που παρέχει, με αποτέλεσμα να παρατηρείται το φαινόμενο της πληροφοριακής υπερφόρτωσης. Αποτέλεσμα της τεράστιας αυτής ποσότητας πληροφοριών που συναντούμε στο ΠΠΠ είναι να κατακλύζεται ο χρήστης τόσο με χρήσιμες, όσο και με άχρηστες για αυτόν πληροφορίες και να εναπόκειται στον ίδιο το χρήστη η ανεύρεση εκείνων των πληροφοριών, οι οποίες του προκαλούν μεγαλύτερο ενδιαφέρον. Το γεγονός αυτό δυσχεραίνει την εύρεση πληροφοριών μέγιστης σημασίας για το χρήστη ή ακόμη, πολλές φορές, τον αποπροσανατολίζει από το στόχο του.

Το πιο πάνω πρόβλημα γίνεται ακόμη μεγαλύτερο, όταν αναφερόμαστε σε κινητούς χρήστες. Στην περίπτωση αυτή, η χρήση και εκμετάλλευση του ΠΠΠ καθίσταται σχεδόν αδύνατη, όχι μόνο λόγω της περιορισμένης υπολογιστικής δύναμης των κινητών συσκευών αλλά και γιατί ο όγκος αυτής της πληροφορίας είναι τόσο μεγάλος, που ένας κινητός χρήστης είναι αδύνατο να διαθέσει το χρόνο για να ψάξει να βρει αυτό που χρειάζεται. Επίσης, ο υπερβολικά μεγάλος αριθμός χρηστών με μη συσχετιζόμενα δεδομένα, επιβάλλει τη δημιουργία νέων υπηρεσιών, με νέους τρόπους δόμησης περιεχομένου, έτσι ώστε να διευκολύνεται η εύρεση εξατομικευμένων αποτελεσμάτων με μέγιστη σημασία για το συγκεκριμένο χρήστη.

Η δημιουργία και διατήρηση ενός συστήματος, το οποίο θα δίνει στο χρήστη αποτελέσματα λαμβάνοντας υπόψη το προσωπικό του προφίλ αλλά και τις ιδιαιτερότητες του κινητού χρήστη αποτελεί πρόκληση. Το γεγονός ότι αναφερόμαστε σε κινητούς χρήστες, των οποίων οι ανάγκες πλέον δεν περιορίζονται σε αυτές που γνωρίζαμε μέχρι τώρα για τους σταθερούς χρήστες αλλά αλλάζουν σε σχέση με το χώρο και τον τόπο στον οποίο βρίσκονται, είναι οι σημαντικότεροι άξονες που επηρεάζουν το κινητό χρήστη.

Η εργασία αυτή συμπληρώνει και εμπλουτίζει σχετική έρευνα για την δημιουργία ενός συστήματος το οποίο διαμορφώνει το προσωπικό προφίλ του κάθε χρήστη χρησιμοποιώντας ένα μηχανισμό με βάρη προτεραιοτήτων στις πληροφορίες που φαίνεται να τον ενδιαφέρουν. Το σύστημα αναφέρεται σε κινητούς χρήστες, των οποίων τα ενδιαφέροντα και οι πληροφορίες που χρειάζονται αλλάζουν ανάλογα με το χρόνο και τόπο στον οποίο βρίσκονται. Έτσι, το προφίλ του χρήστη δημιουργείται συναρτήσει τριών αξόνων· του χρόνου, του τόπου, της παρούσας κατάστασής του (δηλ. αν βρίσκεται στη δουλειά, σε διακοπές κλπ.). Μ’ αυτό τον τρόπο, το προφίλ

αλλάζει προτεραιότητες με βάση τους τρις αυτούς άξονες, επιδιώκοντας να παρέχει στο χρήστη τις πληροφορίες που πιθανότατα χρειάζεται κατά τη στιγμή της αναζήτησης, με όσο το δυνατό μεγαλύτερη ακρίβεια. [1] , [2]

Στην εργασία αυτή, προχωρούμε ένα βήμα πιο βαθιά για να μελετήσουμε το καταλληλότερο τρόπο με τον οποίο μπορούν να διατηρηθούν και να ενημερωθούν τα προφίλ των χρηστών ώστε να αντιπροσωπεύουν σε όσο το δυνατό μεγαλύτερο βαθμό τον χρήστη. Μια ακόμη πρόκληση στην έρευνα αυτή, είναι το πόσο γρήγορα μπορούν να εντοπιστούν αλλαγές στη συμπεριφορά του χρήστη και πόσο γρήγορα μπορεί να διαφοροποιηθεί το προφίλ ώστε να τις αντιπροσωπεύει κατάλληλα.

1.3 Σχετική Έρευνα

Στη συνέχεια μελετούμε μερικές άλλες εργασίες που έγιναν με κύριο άξονά τους την εξατομίκευση στο διαδίκτυο. Σημαντικός είναι ο τρόπους που χρησιμοποιούν για αναπαράσταση των ενδιαφερόντων του χρήστη, την εύρεση και ενημέρωση των ενδιαφερόντων αυτών καθώς και ο τρόπος που εξάγουν τα εξατομικευμένα αποτελέσματα.

Στο [38] διερευνάται η εξατομίκευση πάνω σε επερωτήσεις που αναφέρονται σε βάσεις δεδομένων. Αν και η όλη υλοποίηση δεν αναφέρεται σε κινητό δίκτυο ούτε σε οντολογίες εντούτοις περιγράφει ένα μοντέλο εξατομίκευσης πολύ ενδιαφέρον. Η έρευνα αυτή επικεντρώνεται στο μοντέλο που περιγράφει τις προτιμήσεις του χρήστη, στο πώς αντιστοιχείται και συγκρίνεται το προφίλ του χρήστη με τις επερωτήσεις που ζητά από το σύστημα και στο πώς το σύστημα παρουσιάζει τα αποτελέσματα της εξατομίκευση αυτής.

Το μοντέλο που περιγράφει τις προτιμήσεις του χρήστη ορίζεται σε δύο άξονες. Από την μια περιγράφει τις προτιμήσεις του χρήστη στις τιμές των χαρακτηριστικών των διάφορων οντοτήτων του συστήματος και από την άλλη, περιγράφει προτιμήσεις στις σχέσεις μεταξύ των οντοτήτων αυτών. Οι προτιμήσεις του χρήστη για την τιμή ενός χαρακτηριστικού ορίζεται ως εξής:

$$doi = < d_T(u), d_F(u) >$$

Όπου,

u: Το χαρακτηριστικό

$d_T(u), d_F(u) \in [-1,1]$ και $d_T(u) * d_F(u) \leq 0$

Ο πιο πάνω ορισμός δίνει την προτίμηση του χρήστη στην τιμή ενός χαρακτηριστικού σε τρεις διαστάσεις. Τον βαθμό προτίμησης, τον βαθμό προτίμησης στην παρουσία η απουσία της τιμής του χαρακτηριστικού και στην ελαστικότητα του χαρακτηριστικού εάν αυτό είναι αριθμητικό. Πιο συγκεκριμένα, το $d_T(u)$ δηλώνει τη προτίμηση του χρήστη στη παρουσία του χαρακτηριστικού και το $d_F(u)$ στην απουσία του. Για αριθμητικά χαρακτηριστικά, οι τιμές των $d_T(u)$ και $d_F(u)$ μπορεί να δηλωθούν σαν ελαστικές χρησιμοποιώντας τον όρο $e_{(d)}$.

Οι προτιμήσεις του χρήστη σε ότι αφορά τις σχέσεις δύο οντοτήτων ορίζονται πολύ πιο απλά και δίνονται ως εξής:

$$doi = < d >, \text{όπου } d \in [0,1]$$

Στην όλη υλοποίηση, το προφίλ του χρήστη αναπαρίσταται σαν ένας γράφος με κόμβους τις οντότητες, τα χαρακτηριστικά και τις τιμές που παίρνουν, και με ακμές τους συνδέσμους μεταξύ των τιμών και των χαρακτηριστικών, καθώς και τους συνδέσμους μεταξύ των οντοτήτων που δηλώνουν και την σχέση τους. Στις ακμές αυτές παρουσιάζονται και τα αντίστοιχα ποσοστά προτίμησης σε κάθε περίπτωση. Οι επερωτήσεις, επίσης, αναπαρίστανται σαν γράφος και οι αντιστοίχηση του προφίλ με τις επερωτήσεις του χρήστη γίνεται θεωρώντας ότι ο γράφος των επερωτήσεων είναι ένα κομμάτι του γράφου που αναπαριστά το προφίλ. Το σύστημα βρίσκει τα κοινά μονοπάτια στους δύο γράφους και συνδυάζοντας τις προτιμήσεις στις κοινές ακμές, βρίσκει τις πρώτες *k* προτιμήσεις που θα ληφθούν υπόψη στις απαντήσεις του συστήματος προς το χρήστη. Οι απαντήσεις αυτές εμφανίζονται ταξινομημένες με βάση το βαθμό ενδιαφέροντος (*doi*) και πρέπει να ικανοποιούν τουλάχιστον *l* από τις πρώτες *k* προτιμήσεις.

Στο [39] μελετάται η αυτόματη δημιουργία προφίλ χρηστών στο διαδίκτυο χρησιμοποιώντας οντολογίες και κατηγοριοποίηση κειμένου (text classification). Στην έρευνα αυτή το προφίλ που δημιουργείται είναι μια οντολογία, η οποία για κάθε έννοια (concept) χρησιμοποιεί ένα βάρος που αντιπροσωπεύει το ενδιαφέρον του χρήστη για την έννοια αυτή. Το προφίλ αυτό, δημιουργείται αναλύοντας τις σελίδες που επισκέπτεται σε ότι αφορά το περιεχόμενο, το μέγεθος και την ώρα που πέρασε ο χρήστης στη σελίδα.

Εφόσον το προφίλ του χρήστη είναι μια οντολογία με βάρη προτίμησης, για να είναι εφικτή η σύγκριση μεταξύ του προφίλ και του περιεχομένου πρέπει το περιεχόμενο να μπορεί επίσης να αναπαρασταθεί από μια οντολογία. Στη περίπτωση

αυτή σαν βάση, στις οντολογίες που δημιουργούνται για το προφίλ και το περιεχόμενο, χρησιμοποιήθηκαν οι ήδη υπάρχον θεματικές ιεραρχίες που χρησιμοποιούνται σε πόρταλς όπως το Yahoo.com και About.com. Κάθε σελίδα συνδέεται με μια έννοια, αυτήν στην οποία καταλήγει στην θεματική ιεραρχία. Επιπλέον κάθε έννοια, αναπαρίσταται από ένα διάνυσμα που περιέχει τους όρους του λεξικού τις οντολογίας. Οι όροι αυτοί, εξάγονται με κατηγοριοποίηση κειμένου στους κόμβους περιεχομένου που βρίσκονται κάτω από την ίδια θεματική ιεραρχία. Το βάρος που δίνεται σε ένα συγκεκριμένο όρο (tc_{ij}) είναι ο συντελεστής της συχνότητας του όρου αυτού στους κόμβους που βρίσκονται κάτω από την ίδια θεματική ενότητα (tf_{ij}), και τις σπανιότητας του όρου σε άλλες έννοιες (idc_j) και δίνεται ως εξής:

$$tc_{ij} = tf_{ij} * idf_i$$

Με τον ίδιο ακριβώς τρόπο δημιουργούνται και οι οντολογίες που αναπαριστούν το προφίλ του χρήστη. Ουσιαστικά, δημιουργούνται προφίλ για κάθε ιστιακό χώρο με τον ίδιο τρόπο όπως δημιουργούνται τα προφίλ των χρηστών, με μόνο τρεις διαφορές. Αρχικά, στα προφίλ χρηστών συλλέγονται αντιπροσωπευτικές ιστοσελίδες με τη επεξεργασία σελίδων τις οποίες ο χρήστης επισκέφτηκε πρόσφατα (browsing cache) ενώ για τους ιστιακούς χώρους χρησιμοποιούνται αράχνες (spiders) για να αντιπροσωπευτικές σελίδες που συνδέονται με τον ιστιακό χώρο. Η δεύτερη διαφορά είναι ότι στο προφίλ του χρήστη τα βάρη προσαρμόζονται και λαμβάνουν υπόψη το χρόνο που πέρασε ένας χρήστης σε μια σελίδα του ιστιακού χώρου. Και τέλος, στο προφίλ χρηστών κρατείται μόνο πληροφορία σχετική με τα βάρη για κάθε όρο, ενώ στο προφίλ ιστιακών χώρων καταγράφονται οι βασικές έννοιες στις οποίες κατηγοριοποιείται ο ιστιακός χώρος.

Κατά την σύγκριση της οντολογίας του προφίλ του χρήστη και ενός ιστιακού χώρου, χρησιμοποιείται cosine similarity measure και ένας συντελεστής χρόνου ως εξής:

$$\text{Similarity}(d_k, c_j) = \text{time-length-factor} * \text{cosine-similarity}\ (d_k, c_j)$$

Με το πιο πάνω μέτρο υπολογίζεται η ομοιότητα της οντολογίας του ιστιακού χώρου με την αντίστοιχη οντολογία στο προφίλ του χρήστη και επιστρέφονται οι πρώτες 30 έννοιες. Αφού το σύστημα έχει αντιστοιχήσει τις έννοιες της οντολογίας του ιστιακού χώρου με αυτές της οντολογίας του προφίλ, υπολογίζεται ο «όρος αντιστοιχίας» ο οποίος υπολογίζει πόσο κοντά βρίσκονται οι αντιστοιχίες έννοιες. Ο όρος αντιστοιχίας υπολογίζεται ως εξής:

$$mapping\_factor = \frac{\frac{maching\_weight}{file\_size\_of\_personalized\_concept}}{\frac{wight\_of\_reference\_concept\_queried\_against\_itself}{file\_size\_of\_reference\_concept}}$$

Τέλος, για τις σελίδες που ανήκουν στις έννοιες που αντιστοιχήθηκαν, η ομοιότητα μεταξύ της σελίδας και της έννοιας της οντολογίας του ιστιακού χώρου στην οποία ανήκει, υπολογίζεται με την πιο κάτω συνάρτηση:

*new weight = similarity between page and reference ontology concept * mapping factor*

Στα τελικά αποτελέσματα μπορούν να εφαρμοστούν τεχνικές re-ranking και φιλτραρίσματος. Επιλέγοντας την κατάλληλη τεχνική re-ranking τα αποτελέσματα που αντιπροσωπεύουν καλύτερα τον χρήστη εμφανίζονται ψηλότερα. Επιπλέον καλά φίλτρα μπορούν να απομακρύνουν σελίδες που δεν είναι σχετικές με τον χρήστη.

Το [40] αναπτύσσεται μια έρευνα η οποία μελετά την προσαρμογή της διαδικασίας πλοήγησης στο διαδίκτυο για κινητές συσκευές. Η μελέτη αυτή περιορίζεται στην εισήγηση συντομευμένων συνδέσμων σε πραγματικό χρόνο ώστε να μπορούν οι κινητοί χρήστες να φτάσουν ευκολότερα στον επιθυμητό κόμβο. Η διαδικασία αυτή χρησιμοποιεί ένα μοντέλο εκμάθησης της συμπεριφοράς του χρήστη.

Πιο συγκεκριμένα αναπτύσσεται ο αλγόριθμος MINPATH, ο οποίος είναι υπεύθυνος για να βρίσκει σύντομα μονοπάτια υψηλής ποιότητας σε πραγματικό χρόνο, χρησιμοποιώντας ένα μοντέλο εκμάθησης. Για να το κάνει αυτό, μελετά σε μη πραγματικό χρόνο (offline) την συμπεριφορά του χρήστη και χτίζει το μοντέλο εκμάθησης με βάση τις εγγραφές πρόσβασης του χρήστη (user access logs). Πιο συγκεκριμένα βασίζεται στην συμπεριφορά του χρήστη για να υπολογίσει την πιθανότητα κάθε πιθανής κατάληξης ενός μονοπατιού από συνδέσμους.

Το σημαντικό στοιχείο στον αλγόριθμο MINPATH, είναι το μοντέλο πρόβλεψης που ακολουθεί. Πιο συγκεκριμένα η εργασία αυτή μελετά τρία μοντέλα.

1. Μοντέλο άνευ όρων (unconditional model) : Προβλέπει την επόμενη σελίδα χωρίς να λαμβάνει υπόψη οποιεσδήποτε άλλες συνθήκες. Για να το κάνει αυτό υπολογίζει το ποσοστό των αιτημάτων για κάθε σελίδα *p*.

$$P_{(p_i=q)} = \frac{number-of-times-q-requested}{total-number-of-page-requests}$$

2. Μικτό μοντέλο Naïve Bayes (Naïve Bayes Mixture Mondel): Προβλέπει την επόμενη σελίδα βασισμένος στο σκεπτικό ότι ακόμη κι ο ίδιος χρήστης ακολουθεί διαφορετικά μονοπάτια σε διαφορετικές επισκέψεις. Εναλλακτικά γίνεται η υπόθεση ότι κάθε μονοπάτι ανήκει σε μια από τις K συστάδες (cluster) η κάθε μια από τις οποίες περιγράφεται από ένα μοντέλο. Έτσι υπολογίζεται η πιθανότητα να ζητηθεί η σελίδα *p* προϋποθέτοντας την ταυτότητα της συστάδας C_k :

$$P(p_i = q \mid < p_o, \ldots, p_{i-1} >) = \sum_{k=1}^{K} P(p_i = q \mid C_k) P(C_k \mid < p_o, \ldots, p_{i-1} >)$$

3. Μοντέλο Μαρκόβ (Markov Model): Οι προηγούμενες δύο λύσεις αγνοούν την σειριακή φύση των μονοπατιών. Το μοντέλο αυτό, ενσωματώνει αυτή την πληροφορία, υπολογίζοντας την πιθανότητας της επόμενης σελίδας, με βάση την προηγούμενη:

$$P(p_i = q \mid p_{i-1})$$

1.4 Συνεισφορά έρευνας

Η εργασία αυτή αποτελεί μέρος και εκβάθυνση της αρχιτεκτονικής συστημάτων εξατομίκευσης στο χώρο του ασύρματου δικτύου που προτείνεται στο [4].
Η αρχιτεκτονική αυτή είναι μια ευέλικτη αρχιτεκτονική συστημάτων εξατομίκευσης για το χώρο του ασύρματου διαδικτύου, που υλοποιείται με χρήση κινητών πρακτόρων.

Η αρχιτεκτονική αυτή, τείνει να χρησιμοποιεί συστατικά, τα οποία είναι αυτόνομα και ανεξάρτητα μεταξύ τους. Γι' αυτό το λόγο έχει σαν βασικό δομικό της στοιχείο κινητούς πράκτορες. Παράλληλα, έρχεται να καλύψει το ρόλο του διαμεσολαβητή μεταξύ του πελάτη, ο οποίος ζητά κάποιο πληροφοριακό περιεχόμενο, και του εξυπηρετητή, ο οποίος του το παρέχει. Αυτό γίνεται με χρήση κινητών πρακτόρων, οι οποίοι επιλέγουν, ετοιμάζουν και παραδίνουν το επιθυμητό περιεχόμενο. Για να είναι όμως αυτό δυνατό, χρειαζόμαστε ένα τρόπο για να περιγράψουμε το περιεχόμενο και τη δομή του για κάθε παροχέα, καθώς επίσης και ένα τρόπο διαχείρισης του προφίλ του κάθε χρήστη. Έτσι, έχοντας από τη μία ένα σύστημα που περιγράφει τη δομή του περιεχομένου και από την άλλη ένα σύστημα που διαχειρίζεται το προφίλ του χρήστη, είμαστε σε θέση να αλλάξουμε δυναμικά τη δομή του περιεχομένου και να το διαμορφώσουμε σύμφωνα με τις ανάγκες του χρήστη.

Η πιο πάνω αρχιτεκτονική αποτελείται κυρίως από τα πιο κάτω συστατικά:

- Το σύστημα περιγραφής της δομής περιεχομένου
- Το σύστημα επιλογής περιεχομένου
- Το σύστημα μορφοποίησης του περιεχομένου
- Τη διαχείριση των προσωπικών προφίλ των χρηστών

Πιο αναλυτική περιγραφή της αρχιτεκτονικής αυτής και συγκεκριμένα των υποσυστημάτων επιλογής περιεχομένου και διαχείρισης των προσωπικών προφίλ των χρηστών θα δούμε στο κεφάλαιο 3. [3], [4]

1.5 Σκιαγράφηση έρευνας

Όπως έχει ήδη αναφερθεί, στόχος της έρευνας που πραγματοποιείται μέσα στα πλαίσια αυτής της διπλωματικής εργασίας εξειδίκευσης, είναι η μελέτη αλγορίθμων ενημέρωσης των προφίλ χρηστών που χρησιμοποιούνται για παροχή εξατομικευμένων υπηρεσιών. Η εργασία αυτή είναι εκβάθυνση ήδη υπάρχουσας έρευνας ενός συστήματος που παρέχει εξατομικευμένες υπηρεσίες σε κινητούς και που αναφέρεται στην αρχιτεκτονική συστημάτων εξατομίκευσης στο ασύρματο περιβάλλον. Προς την επίτευξη αυτού το στόχου έχουν τεθεί οι ακόλουθοι επιμέρους στόχοι:

- Παρουσίαση σχετικής έρευνας.
- Μελέτη του συστήματος που παρέχει εξατομικευμένες υπηρεσίες σε κινητούς χρήστες και της αρχιτεκτονικής συστημάτων εξατομίκευσης σε ασύρματο περιβάλλον.
- Παρουσίαση του προβλήματος της εξατομίκευσης.
- Σχεδιασμός και μελέτη αλγορίθμων ενημέρωσης προφίλ χρηστών.
- Πειραματικές μετρήσεις των αλγορίθμων.
- Αναφορά και παρουσίαση των ιδεών μας για μελλοντική εργασία.
- Συμπεράσματα.

Κεφάλαιο 2

Τεχνολογικό και Θεωρητικό υπόβαθρο.

Στο κεφάλαιο αυτό, θα μελετηθούν το τεχνολογικό και θεωρητικό υπόβαθρο της εργασίας. Αναλύονται οι τεχνολογίες που χρησιμοποιήθηκαν αλλά και μαθηματικοί θεωρητικοί όροι στους οποίους στηρίχθηκαν οι αλγόριθμοι.

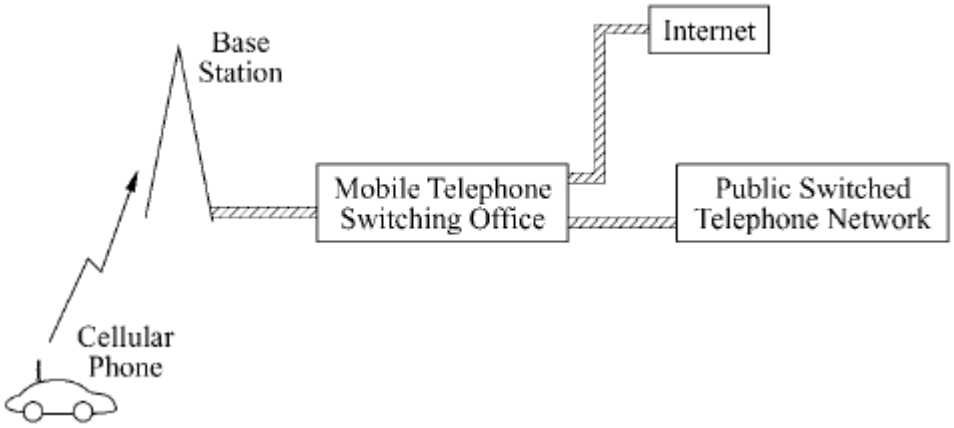
Οι αλγόριθμοι αναπτύχθηκαν στη γλώσσα προγραμματισμού JAVA και με τέτοιο τρόπο ώστε να λαμβάνουν υπόψη τις ιδιαιτερότητες του ασύρματου δικτύου που περιγράφονται. Κατά την ανάπτυξη των αλγορίθμων χρησιμοποιήθηκαν έννοιες και μαθηματικοί μέθοδοι, οι οποίοι και αναλύονται πιο κάτω.

2.1 Ασύρματο Δίκτυο

Η έννοια ασύρματο δίκτυο χρησιμοποιείται για οποιοδήποτε είδος υπολογιστικού δικτύου το οποίο είναι ασύρματο. Το ασύρματο δίκτυο είναι στενά συνδεδεμένο με

τα τηλεπικοινωνιακά δίκτυα, στα οποία η σύνδεση των κόμβων(συσκευή, router, switch, etc) υλοποιείται χωρίς την χρήση συρμάτων. [5]

Στην εργασία αυτή εστιαζόμαστε στα ασύρματα δίκτυα κινητών συσκευών και συγκεκριμένα αναφερόμαστε στο Παγκόσμιο Σύστημα Κινητής Τηλεφωνίας (Global System for Mobile Communications - GSM). Το δίκτυο GSM χωρίζεται σε τρία κυρίως συστήματα τα οποία είναι: το σύστημα εναλλαγής (switching system), το σύστημα του σταθμού – βάση (base station system) και το σύστημα λειτουργίας και υποστήριξης (operation and support system). Η κινητή συσκευή ενώνεται στο σταθμό-βάση η οποία με τη σειρά της ενώνεται στο σταθμό λειτουργία και υποστήριξης. Στη συνέχεια ενώνεται στο σταθμό εναλλαγής όπου η κλήση μεταφέρεται εκεί που χρειάζεται.[6]



Τα ασύρματα δίκτυα είχαν σημαντική επίδραση στο κόσμο τις τελευταίες δεκαετίες. Οι κινητές συσκευές είναι πλέον μέρος ενός τεράστιου ασύρματου δικτύου. Οι άνθρωποι χρησιμοποιούν αυτές τις συσκευές όχι απλά για να επικοινωνήσουν τηλεφωνικά αλλά και για να μοιραστούν γρήγορα και εύκολα πληροφορίες, ανεξάρτητα από του πού βρίσκονται και πότε. Η ανάπτυξη του ασύρματου δικτύου και της κινητής τηλεφωνίας τα τελευταία χρόνια δεν θα μπορούσε να μην επηρεάσει την ανάπτυξη της ασύρματης επικοινωνίας, και κατ' επέκταση του ασύρματου διαδικτύου για παροχή υπηρεσιών και πληροφοριών.

Ωστόσο οι συσκευές αυτές επιβάλλουν κάποιους περιορισμούς, όχι μόνο λόγω της περιορισμένης υπολογιστικής δύναμης που έχουν αλλά και λόγω του μεγέθους τους (συνήθως είναι μικρές για να μεταφέρονται εύκολα). Πιο συγκεκριμένα, μια κινητή συσκευή διαφέρει από ένα desktop PC στα πιο κάτω:

- Μέγεθος: Μία κινητή συσκευή πρέπει να είναι αρκετά μικρή ώστε να μετακινείται εύκολα και ιδανικά, τόσο μικρή ώστε να χωρεί στην παλάμη ενός χεριού.

- Επεξεργαστής: Συνήθως οι επεξεργαστές κινητών συσκευών δεν διαφέρουν μόνο στο γεγονός ότι έχουν λιγότερη υπολογιστική δύναμη από τα επιτραπέζια PC (desktop PC), αλλά επίσης έχουν και βασικές διαφορές στην αρχιτεκτονική τους.

- Μνήμη και χώροι αποθήκευσης: Στις κινητές συσκευές επιβάλλονται σημαντικοί περιορισμοί όσον αφορά την μνήμη, αφού είναι πολύ μικρότερη.

- Οθόνη: Η οθόνη στις συσκευές αυτές συνήθως περιορίζεται σε πολύ μικρό μέγεθος και χαμηλή ανάλυση, ενώ σε πολλές περιπτώσεις έχουμε μόνο μαυρόασπρες οθόνες.

- Είσοδος δεδομένων: Οι περισσότερες κινητές συσκευές είτε δεν έχουν πληκτρολόγιο, είτε έχουν αλλά είναι περιορισμένου μεγέθους. Έτσι, η διαδικασία εισόδου δεδομένων για επεξεργασία γίνεται δυσκολότερη. Ακόμη, στις συσκευές αυτές μπορούμε να εισάγουμε δεδομένα και με ήχο καθώς και με εικόνα.

Επιπρόσθετα σε αυτούς τους περιορισμούς, υπάρχουν και περιορισμοί που επιβάλλονται από το ίδιο το ασύρματο δίκτυο, στο χώρο του οποίου υπάρχουν οι συσκευές αυτές. Τόσο το εύρος ζώνης (Bandwidth), όσο επίσης η αξιοπιστία (Reliability) και ο χρόνος απόκρισης (Latency) του δικτύου είναι παράγοντες που επηρεάζουν τη χρήση των κινητών συσκευών και επιβάλλουν περιορισμούς στην ανάπτυξη του ασύρματου διαδικτύου. Συγκεκριμένα, στα ασύρματα δίκτυα έχουμε πολύ μικρότερο εύρος ζώνης, ενώ παράλληλα το δίκτυο έχει μικρότερη αξιοπιστία και μεγαλύτερο χρόνο απόκρισης.[7]

Όλα τα πιο πάνω επέβαλλαν την ανάγκη για δημιουργίας εφαρμογών, οι οποίες, λαμβάνοντας υπόψη τις αδυναμίες του ασύρματου δικτύου και τα χαρακτηριστικά των κινητών συσκευών που περιγράψαμε πιο πάνω, κάνει εφικτή την επικοινωνία κινητών συσκευών με δίκτυα υπολογιστών και συγκεκριμένα με το διαδίκτυο για ανταλλαγή δεδομένων και παροχή πληροφοριών στο χρήστη.

2.2 Τεχνολογία XML

Ο Παγκόσμιος Ιστός είναι ένα μέσο για ανταλλαγή και διάδοση πληροφοριών. Η συνεχώς αυξανόμενη χρήση του web συνέβαλε στη χρήση βάσεων δεδομένων και στο διαδίκτυο. Το γεγονός ότι ολοένα και περισσότερα δεδομένα Ιστού παράγονται από βάσεις δεδομένων, οδήγησε στην ανάγκη γεφύρωσης του Ιστού με αυτές. Όμως, τα συστήματα βάσεων δεδομένων λειτουργούν πάνω σε δομημένα δεδομένα, όπου το σχήμα των δεδομένων είναι καθορισμένο εκ των προτέρων. Σε αντίθεση, στο διαδίκτυο, τα δεδομένα μπορεί να μην έχουν μια αυστηρά καθορισμένη δομή, με συνέπεια οι βάσεις δεδομένων να μην είναι ό,τι καλύτερο για δόμηση δεδομένων στο web. Αποτέλεσμα αυτού, ήταν η δημιουργία και χρήση της γλώσσας XML. Η γλώσσα αυτή, μπορεί να χειρίζεται ημιδομημένα δεδομένα, δηλαδή δεδομένα που έχουν μη σταθερό σχήμα και παρουσιάζουν αποκλίσεις στη δομή τους. Μπορεί, για παράδειγμα, κάποια πεδία να απουσιάζουν ή να εμφανίζονται διπλά. Επιπλέον, η XML είναι μια γλώσσα σήμανσης, η οποία, σε αντίθεση με την HTML, έχει σαν στόχο να ορίσει ένα τρόπο για καθορισμό συσχετίσεων ανάμεσα στα δεδομένα και όχι τον καθορισμό συσχετίσεων ανάμεσα στα στοιχεία ενός αρχείου. Δίνει έμφαση στην περιγραφή της δομής της πληροφορίας αντί στη δομή της εμφάνισή της. [9]

Η XML προσφέρει ένα σύνολο από κανόνες για καθορισμό της δομής των δεδομένων. Δίνει τη δυνατότητα σχεδιασμού και καθορισμού της διάταξης των δεδομένων σε μορφή κειμένου (text format), ώστε να μπορεί με ευκολία να κατανοηθεί τόσο από έναν υπολογιστή, όσο και από τον ίδιο τον άνθρωπο. Επίσης, με την XML ένας υπολογιστής μπορεί εύκολα να παράγει και να διαβάζει δεδομένα, καθώς και να εξασφαλίζει την ξεκάθαρη δομή των δεδομένων. Έτσι, με αυτή την αυστηρή δομή επιτυγχάνεται η εύκολη μεταφορά δεδομένων σε διαφορετικούς υπολογιστές και συστήματα.

Η XML είναι μια γλώσσα σήμανσης, η οποία δεν έχει προκαθορισμένους σημαντήρες και γραμματικές, κάτι που την καθιστά επεκτάσιμη. Είναι δηλαδή μια μετα-γλώσσα η οποία μας επιτρέπει τον ορισμό και την περιγραφή άλλων γλωσσών σήμανσης. Ακόμη, η XML είναι ανεξάρτητη από την πλατφόρμα στην οποία χρησιμοποιείται. Η δομή της είναι ιεραρχική και υποστηρίζει επικύρωση (validation). Η επικύρωση μπορεί να γίνει με χρήση DTD ή XML Schema. Τα δύο αυτά σχήματα επιτρέπουν τη δημιουργία γραμματικών με χρήση των σημαντήρων της XML και των ιδιοχαρακτηριστικών τους, έτσι ώστε να καθορίζεται η δομή της πληροφορίας. [8]

Περισσότερες πληροφορίες για το πως χρησιμοποιείται η XML μπορεί ο αναγνώστης να βρει στο [9], [10] και [11].

2.2.1 XML Schema Vs DTD

Το DTD είναι μια γραμματική για αρχεία χωρίς συμφραζόμενα (context-free grammar). Μπορεί να ορίσει τα στοιχεία ενός αρχείου, καθώς επίσης και χαρακτηριστικά στα στοιχεία αυτά. Επιπρόσθετα, καθορίζει τη σειρά τους, ενώ επιτρέπει αναδρομή και φώλιασμα στοιχείων. Σε ένα DTD μπορούμε να ορίσουμε στοιχεία τα οποία είτε περιέχουν άλλα στοιχεία, είτε περιέχουν στοιχεία τα οποία αποτελούνται μόνο από κείμενο.

Το XML Schema παρέχει έναν πιο δυναμικό τρόπο για ορισμό των δομών και των περιορισμών ενός XML αρχείου. Παρέχει ένα σύνολο από τύπους δεδομένων που υποστηρίζουν οι περισσότερες προγραμματιστικές γλώσσες (όπως string, integers, float κ.α.), και παράλληλα δίνει την δυνατότητα ορισμού πολύπλοκων στοιχείων χρησιμοποιώντας τους πιο πάνω τύπους δεδομένων.

Τα DTDs παρέχουν μια βασική γραμματική για τον ορισμό ενός αρχείου XML σε σχέση με τα μεταδεδομένα που περιγράφει το αρχείο. Ένα XML αρχείο κάνει ακριβώς το ίδιο πράγμα, αλλά επιπρόσθετα, προσφέρει έναν τρόπο για καθορισμό του τι πρέπει και τι δεν πρέπει να περιέχουν τα δεδομένα. Έτσι προσφέρει μεγαλύτερο έλεγχο στα δεδομένα.

Επίσης, τα DTDs δεν είναι γραμμένα σε XML. Δεν υποστηρίζουν ονοματολογίες (Namespaces) και δεν παρέχουν τρόπους ελέγχου των δεδομένων. Αντίθετα, το XML Schema προσφέρει πλουσιότερη συλλογή τύπων και διευκολύνσεις στη δημιουργία νέων τύπων και αρχέτυπων, ενώ υποστηρίζει ονοματολογίες και ομαδοποίηση χαρακτηριστικών (Attribute grouping). Επιπλέον, έχει το ίδιο συντακτικό με της XML, κι έτσι αρχεία XML Schema μπορούν να τύχουν επεξεργασίας όπως ένα αρχείο XML. [10, 12]

Περισσότερες πληροφορίες για το XML Schema μπορεί ο αναγνώστης να βρει στο [13, 14]

2.3 JAVA

Η Java είναι μια γλώσσα προγραμματισμού η οποία είναι σχεδιασμένη για να μπορεί να αντεπεξέλθει στις προκλήσεις σύγχρονων εφαρμογών που μπορούν να αναπτυχθούν σε ετερογενή και διαδικτυακά κατανεμημένα περιβάλλοντα. Στόχος της είναι η ασφαλής παράδοση εφαρμογών που καταναλώνουν το ελάχιστο των υπολογιστικών πόρων, μπορούν να τρέξουν σε οποιαδήποτε πλατφόρμα λογισμικού και υλικού και που μπορούν να επεκταθούν δυναμικά.

Ο σχεδιασμός της Java στηρίζεται στη φύση των υπολογιστικών περιβαλλόντων στα οποία αναπτύσσονται λογισμικά. Η ραγδαία ανάπτυξη του διαδικτύου και του Παγκόσμιου Πλέγματος Ιστού (ΠΠΠ) μας έχουν οδηγήσει σε ένα παντελώς καινούριο τρόπο ανάπτυξης και διανομής του λογισμικού. Η JAVA είναι η ιδανική γλώσσα που μπορεί να αντεπεξέλθει σε τέτοιου είδους περιβάλλοντα καθώς παρέχει ασφαλή, υψηλής απόδοσης εφαρμογές που μπορούν να τρέξουν σε πολλαπλές πλατφόρμες ετερογενών και κατανεμημένων περιβαλλόντων. Η JAVA είναι μια γλώσσα αντικειμενοστρεφής, μεταφερόμενη (portable) και είναι ιδανική για εφαρμογές που απαιτείται να προσαρμόζονται δυναμικά στο περιβάλλον τους.

Σαν γλώσσα αντικειμενοστρέφειας, η JAVA εκμεταλλεύεται τις σύγχρονες μεθοδολογίες ανάπτυξης συστημάτων και γι’ αυτό το λόγο είναι η καταλληλότερη γλώσσα για κατανεμημένες εφαρμογές και εφαρμογές που στηρίζονται στο μοντέλο πελάτη-εξυπηρετητή. Αυτό γιατί, κατά την ανάπτυξη τέτοιων συστημάτων που απευθύνονται σε πολύπλοκα και διαδικτυακά περιβάλλοντα πρέπει να υιοθετούνται αρχές αντικειμενοστρέφειας. Η JAVA παρέχει μια ξεκάθαρη και αποδοτική πλατφόρμα αντικειμενοστρέφειας.

Είναι γλώσσα που υποστηρίζει multithreading και έτσι μπορεί να εφαρμοστεί σε εφαρμογές υψηλής απόδοσης που χρειάζεται να εκτελούν ταυτόχρονες δραστηριότητες. Επιπλέον σε όλα αυτά η JAVA

Η JAVA είναι σχεδιασμένη ώστε να δημιουργεί εξαιρετικά αξιόπιστα λογισμικά. Παρέχει εκτεταμένο έλεγχο κατά την διάρκεια του compiling που ακολουθείται από ένα δεύτερου επιπέδου έλεγχο κατά την εκτέλεση του προγράμματος. Επιπλέον τα χαρακτηριστικά της γλώσσας οδηγούν τους προγραμματιστές σε αξιόπιστες προγραμματιστικές συνήθειες.

Η διαχείριση της μνήμης στη JAVA είναι εξαιρετικά απλή. Κάτι που περιορίζει στο ελάχιστο τα προγραμματιστριακά λάθη. Η χρήση του αποκομιστή σκυβάλων την κάνει ακόμη ευκολότερη.

Η τεχνολογία της JAVA είναι σχεδιασμένη για να λειτουργεί σε κατανεμημένα συστήματα, κάτι το οποίο σημαίνει ότι η ασφάλεια είναι ύψιστης σημασίας. Με τα χαρακτηριστικά ασφαλείας να είναι σχεδιασμένα μέσα στη γλώσσα, η JAVA, επιτρέπει την δημιουργία εφαρμογών που δεν μπορούν να εισβάλουν σε αυτά.

Ένα από τα σημαντικότητα χαρακτηριστικά της γλώσσας αυτής είναι η μεταφερσιμότητά της. Είναι σχεδιασμένη ώστε να υποστηρίζει εφαρμογές που θα αναπτυχθούν σε ετερογενή δυκτιακά περιβάλλοντα. Σε τέτοια μορφής περιβάλλοντα, οι εφαρμογές πρέπει να εφικτό να μπορούν να τρέξουν σε ένα σύνολο από αρχιτεκτονικές υλικού, σε διαφορετικά λογισμικά συστήματα και να αλληλεπιδράσουν με διπροσωπίες κτισμένες με διαφορετικές γλώσσες προγραμματισμού. [15]

Ο μεταγλωττιστής της Java δεν παράγει εκτελέσιμο κώδικα αλλά παράγει μία μορφή αρχείων γνωστή ως Java Bytecodes που δεν είναι εκτελέσιμη απευθείας από μία μηχανή αλλά εκτελείται από την εικονική μηχανή Java (Java Virtual Machine – JVM). Αυτό δίνει στην Java το πλεονέκτημα της ανεξαρτησίας από την συγκεκριμένη μηχανή στην οποία εκτελείται. Αρκεί κανείς να έχει εγκαταστήσει την κατάλληλη έκδοση της Java και το πρόγραμμα θα εκτελεστεί από την εικονική μηχανή.

Όλα τα πιο πάνω, κάνουν την JAVA την καταλληλότερη γλώσσα που θα μπορούσαμε να χρησιμοποιήσουμε για την ανάλυση και ανάπτυξη των αλγορίθμων που παρουσιάζονται.

2.4 Μαθηματικοί Όροι

Στο υποκεφάλαιο αυτό, θα μελετηθούν μερικοί σημαντικοί μαθηματικοί όροι που χρησιμοποιήθηκαν κατά την ανάπτυξη των αλγορίθμων. Εδώ δίνεται μόνο το θεωρητικό υπόβαθρο των αλγορίθμων, το πώς χρησιμοποιήθηκαν αναπτύσσεται εκτεταμένα στο κεφάλαιο 5.

2.4.1 Μαθηματική Μέση Τιμή(mean), Διάμεση Τιμή(median) και το mode

Η Μέση τιμή, η Διάμεση τιμή και το Mode, είναι τρεις διαφορετικοί μέθοδοι εύρεσης της κεντρικής τάσης των δεδομένων.

Μέσος όρος ή αλλιώς μέση τιμή ενός συνόλου ν αριθμών αποτελεί το σπουδαιότερο και χρησιμότερο μέτρο της Στατιστικής και ορίζεται ως το άθροισμα των παρατηρήσεων δια του πλήθους αυτών. Είναι δηλ. η μαθηματική πράξη ανεύρεσης της «μέσης απόστασης» ανάμεσα σε δύο ή περισσότερους αριθμούς. Στη προκειμένη περίπτωση ο μέσος αριθμός είναι το πηλίκο της διαίρεσης του αθροίσματος δοθέντων αριθμών δια του πλήθους αυτών

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{1}{n}(x_1 + \cdots + x_n)$$

Ωστόσο η μέση τιμή δεν είναι πάντα ο καλύτερος τρόπος εύρεσης του κέντρου των δεδομένων. Σε μεγάλα δείγματα η μέση τιμή τείνει να είναι αξιόπιστη ωστόσο σε δείγματα όπου η διασπορά είναι μεγάλη η μέση τιμή δεν είναι τόσο αξιόπιστη. Στη περίπτωση αυτή η μέση τιμή δεν μπορεί να μας δώσει την καλύτερη τιμή που θα μπορούσε να αντιπροσωπεύσει το κέντρο των δεδομένων.

Η Διάμεση τιμή θεωρείται καλύτερη για skewed data, όπου η κατανομή των δεδομένων είναι ασύμμετρη. Η Διάμεση τιμή, μας δίνει την συγκεκριμένη τιμή στο δείγμα για την οποία οι μισές τιμές του συνόλου είναι μικρότερες τιμές και οι άλλες μισές είναι μεγαλύτερες μισές. Για την εύρεση της διάμεσης τιμής παίρνουμε τη μεσαία τιμή ενός ταξινομημένου δείγματος. Εάν αριθμός των τιμών στο δείγμα είναι ζυγός αριθμός, τότε η διάμεση τιμή είναι ο μέσος όρος των μεσαίων τιμών.

Το mode είναι ακόμη ένας τρόπος εύρεσης της κεντρικής τάσης των δεδομένων. Το mode είναι η τιμή ανάμεσα στα δεδομένα η οποία έχει τη μεγαλύτερη συχνότητα. Αυτή η μέθοδος είναι ιδανική για δεδομένα που τείνουν να έχουν την ίδια τιμή. [16, 17]

2.4.2 Κινητός Μέσος Όρος

Στη στατιστική ο κινητός μέσος όρος είναι μια από τις τεχνικές που χρησιμοποιούνται για την ανάλυση δεδομένων. Ο κινητός μέσος όρος χρησιμοποιείται συνήθως σε οικονομικές τεχνικές αναλύσεις.

Ο κύριος στόχος του αλγορίθμου αυτού είναι να "εξομαλύνει" (smooth) τα δεδομένα ώστε η τάση τους να είναι περισσότερο διακριτή. Με την μέθοδο αυτή είναι ευκολότερο να εντοπιστεί η τάση των δεδομένων. Ο κινητός μέσος όρος είναι η μέση τιμή των δεδομένων για ένα συγκεκριμένο αριθμό χρονικών περιόδων. Κινείται γιατί για κάθε υπολογισμό χρησιμοποιεί δεδομένα από τις τελευταίες x χρονικές περιόδους έτσι ώστε να παραμένουν συγχρονισμένα στο παρών. Υπάρχουν δύο είδη moving average το απλό και το εκθετικό.

Η απλή μέθοδος είναι ένας αριθμητικός μέσος όρος. Στη περίπτωση αυτή όλες οι μέρες της χρονικής περιόδου έχουν την ίδια βαρύτητα και έτσι η μέθοδος αυτή κατά κάποιο τρόπο δεν προβλέπει την τάση τον δεδομένων αλλά την ακολουθεί.

Η εκθετική μέθοδος εφαρμόζει μεγαλύτερη βαρύτητα σε πρόσφατα δεδομένα χωρίς να αγνοεί τελείως τα προηγούμενα (όπως γίνεται στην απλή μέθοδο). Με το τρόπο αυτό δεν χάνονται προηγούμενα δεδομένα αλλά μεταφέρονται από την προηγούμενη περίοδο σαν η βάση για τον υπολογισμό. [18]

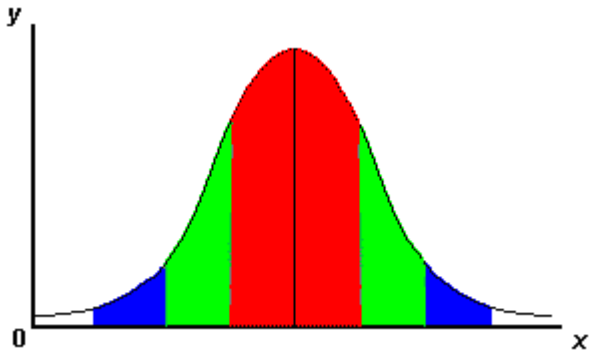
Η εκθετική μέθοδος εφαρμόζεται με το πιο κάτω τύπο:

$$CMA = PPV * (1- SF) + CPV * SF$$

Όπου:
CMA = Current Moving Average
PPV = Previous Period Value
CPV = Current Period Value
SF = Smoothing Factor

2.4.3 Τυπική Απόκληση

Η τυπική απόκληση είναι ένα μέτρο της διασποράς των τιμών ενός συνόλου. Είναι ο πιο κοινός τρόπος εύρεσης της διασποράς ο οποίος υπολογίζει πόσο διασκορπισμένες είναι οι τιμές σε ένα σύνολο τιμών. Εάν ένα μεγάλο ποσοστό των τιμών ενός συνόλου είναι κοντά στο μέσο όρο, τότε η τυπική απόκληση είναι μικρή. Εάν, αντιθέτων, πολλές τιμές του συνόλου απέχουν αρκετά από το μέσο όρο, τότε η τυπική απόκλιση είναι μεγάλη. Εάν όλα τα δεδομένα έχουν ίσες τιμές, τότε η τυπική απόκλιση είναι μηδενική.



Σχήμα 2-1: Τυπική Απόκληση σε μια κανονική κατανομή

Η τυπική απόκλιση N τιμών ενός συνόλου δίνεται από τον τύπο:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2}.$$

Όπου $\overline{x}$ η μέση τιμή των τιμών $x_1 \ldots x_N$. [16]

2.5 Αλγόριθμοι Συσταδοποίησης

Η συσταδοποίηση είναι η διαδικασία ομαδοποίησης ενός συνόλου αντικειμένων σε συστάδες όμοιων αντικειμένων. Μια συστάδα είναι μια συλλογή αντικειμένων που είναι όμοια μεταξύ τους, και ανόμοια με τα αντικείμενα άλλων συστάδων. [17]

Η συστασταδοποίηση είναι ένα εργαλείο για ανάλυση δεδομένων, η οποία λύνει προβλήματα συσταδοποίησης. Σκοπός του είναι να διαμοιράσει τα αντικείμενα σε ομάδες, έτσι ώστε ο βαθμός συσχέτισης του να είναι πολύ δυνατός ανάμεσα στα μέλη της ίδιας ομάδας και αδύνατος με τα μέλη άλλων ομάδων. Με το τρόπο αυτό κάθε συστάδα περιγράφει, σε ότι αφορά τα δεδομένα, την κλάση στην οποία ανήκουν τα μέλη της. Η συσταδοποίηση είναι μια μεθοδολογία, η οποία μπορεί να αποκαλύψει συσχετίσεις μεταξύ των δεδομένων ή ακόμη και την δομή τους, τα οποία μπορεί να μην ήταν εμφανές προηγουμένως παραμένουν ωστόσο σημαντικά ευρήματα που μπορούν να βοηθήσουν σε περαιτέρω έρευνα.

Επιπλέον με την συσταδοποίηση είναι εύκολος ο εντοπισμός της πυκνότητας ή διασποράς των δεδομένων κάτι που είναι ιδιαίτερα σημαντικό στην ανακάλυψη γενικότερων κανόνων και συσχετίσεων ανάμεσα στα δεδομένα.

2.5.1 Το μέτρο της απόστασης

Για να είναι εφικτή η δημιουργία συστάδων είναι σημαντικός ο ορισμός μέτρων αποστάσεων μεταξύ των αντικειμένων. Το μέτρο αυτό είναι το σημαντικότερο ίσως μέρος των αλγορίθμων συσταδοποίησης καθώς είναι αυτό που θα καθορίσει τελικά τις συστάδες και την σχέση μεταξύ τους.

Το μέτρο απόστασης καθορίζει πόσο όμοια είναι δύο αντικείμενα μεταξύ τους. Συνήθη μέτρα απόστασης που χρησιμοποιούνται σε χώρους δύο διαστάσεων είναι τα πιο κάτω:

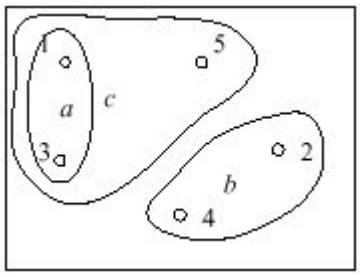
1. Common distance = $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$
2. Manhattan distance = $\sum_{i-1}^{k}|x_i - y_i|$
3. Max of dimensions = $\max_{i=1}^{k}|x_i - y_i|$

*όπου κ ο αριθμός των μεταβλητών που ορίζουν τα δύο σημεία.

Σε περιπτώσεις που τα αντικείμενα μας δεν μπορούν να αναπαρασταθούν σε κάποιον δισδιάστατο χώρο, τότε τα πράγματα είναι δυσκολότερα και χρειάζονται άλλοι τρόποι για τον ορισμό της απόστασης μεταξύ τους.

2.5.2 Ο αλγόριθμος k-means

Ο αλγόριθμος k-means είναι ένας από τους πιο απλούς αλγορίθμους μάθησης ο οποίος χρησιμοποιείται ευρέος σε μεθοδολογίας διαμερισμού. Ο βασικός αλγόριθμος ακολουθεί ένα απλό και εύκολο τρόπο για να κατηγοριοποιήσει τα δεδομένα σε ένα συγκεκριμένο αριθμό συστάδων.

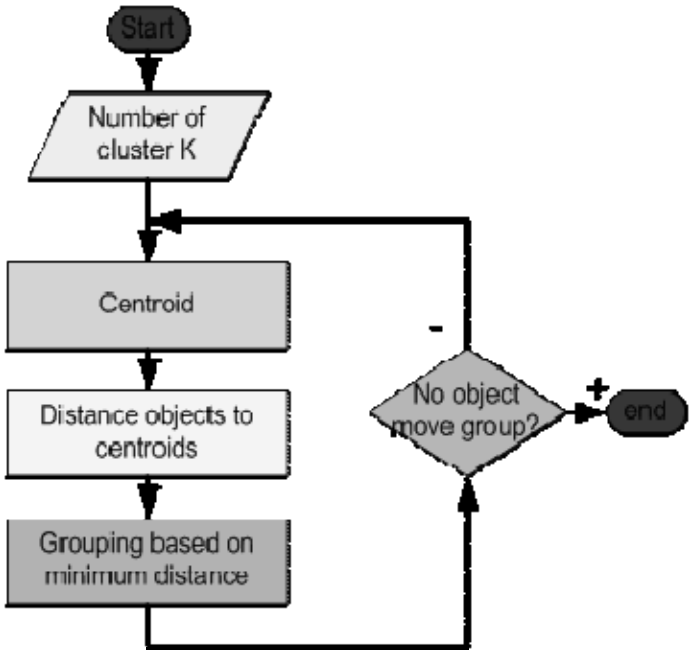


Σχήμα 2-2: Συσταδοποίηση

Στο k-means η κάθε συστάδα αναπαρίσταται από τη μέση τιμή των σημείων που ανήκουν σε αυτή. Η τιμή αυτή αναπαριστά το κέντρο της συστάδας. Το άθροισμα της απόστασης ανάμεσα σε ένα σημείο και το κέντρο της συστάδας χρησιμοποιείται είναι η βασική συνάρτηση που χρησιμοποιείται στο μεθοδολογία αυτή. Η συσταδοποίηση γίνεται ελαχιστοποιώντας το άθροισμα των τετραγώνων των αποστάσεων μεταξύ των δεδομένων και του κέντρου της αντίστοιχης συστάδας.

Αρχικά ο αλγόριθμος ορίζει τα κέντρα των κ συστάδων. Αυτά τα κέντρα πρέπει να τοποθετηθούν όσο το δυνατό πιο μακριά το ένα από το άλλο. Το επόμενο βήμα είναι να συσχετιστεί κάθε σημείο του δεδομένου συνόλου με τη πιο κοντινή συστάδα. Το κέντρο της συστάδας ξανά υπολογίζεται κάθε φορά που ένα νέο σημείο προστίθεται στη συστάδα κι αυτό συνεχίζεται μέχρι όλα τα σημεία να ομαδοποιηθούν στο ζητούμενο αριθμό συστάδων.

Ο αλγόριθμος συνοπτίζεται στα πιο κάτω βήματα και δίνεται από το σχήμα που ακολουθεί:



Σχήμα 2-3: Ο αλγόριθμος k-means

1. Αποφασίζεται ο αριθμός συστάδων k

2. Καθορίζουμε ένα αρχικό διαχωρισμό, ο οποίος κατηγοριοποιεί τα δεδομένα σε k συστάδες. Η ανάθεση αυτή μπορεί να γίνει και τυχαία ή και ακολουθώντας τη πιο κάτω διαδικασία:

2.1. Τα πρώτα k δείγματα θεωρούνται οι αρχικές συστάδες

2.2. Κάθε εναπομείναντα δείγμα ανατίθεται στην πιο κοντινή συστάδα

2.3. Μετά από κάθε ανάθεση υπολογίζεται το κέντρο της κάθε συστάδας

3. Στη συνέχεια υπολογίζεται η απόσταση κάθε σημείου από όλες τις συστάδες. Εάν ένα σημείο δεν βρίσκεται στη πιο κοντινή του συστάδα, τότε μεταφέρουμε το σημείο στη συστάδα αυτή και υπολογίζουμε ξανά τα κέντρα των δύο εμπλεκόμενων συστάδων

4. Το τρίτο βήμα επαναλαμβάνεται μέχρι να επιτευχθεί σύγκλιση. Δηλαδή μέχρι να μην υπάρχουν άλλες νέες αναθέσεις σε συστάδες.

Εάν ο αριθμός των δεδομένων είναι μικρότερος των συστάδων τότε αναθέτουμε κάθε σημείο ως το κέντρο της συστάδας. Εάν ο αριθμός των δεδομένων είναι μεγαλύτερος από τον αριθμό των συστάδων, για κάθε σημείο υπολογίζουμε την απόστασή του από το κέντρων όλων των συστάδων και αναθέτουμε το σημείο στη συστάδα με την οποία έχει την μικρότερη απόσταση.

Εφόσον δεν είμαστε σίγουροι για την τοποθεσία του κέντρου, χρειάζεται να προσαρμόζουμε το κέντρο κάθε συστάδας με βάση τα τρέχον δεδομένα της. Στη συνέχεια γίνεται ανάθεση όλων των δεδομένων της συστάδας στο κέντρο της. Αυτή η διαδικασία επαναλαμβάνεται έως ότου κανένα σημείο δεν αλλάζει συστάδα. Τέλος, ο αλγόριθμος αυτός επιδιώκει να ελαχιστοποιήσει μια αντικειμενική συνάρτηση. Συνήθως η συνάρτηση που επιλέγεται είναι η συνάρτηση τετραγωνικού σφάλματος(squared error function) . Η συνάρτηση αυτή, υπολογίζει ποσοτικά την διαφορά της τιμής που υπολογίστηκε από την πραγματική τιμή. Με τον τρόποαυτό υπολογίζει το πιθανό σφάλμα. Η συνάρτηση αυτή δίνεται πιο κάτω :

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2$$

Όπου: $\left\| x_i^{(j)} - c_j \right\|^2$ είναι το μέτρο απόστασης μεταξύ του σημείου $x_i^{(j)}$ και του κέντρου της συστάδας c_j και μας δίνει την απόσταση των σημείων από τα αντίστοιχα κέντρα των συστάδων. [17]

2.6 Αλγορίθμοι Κατηγοριοποίησης

Η κατηγοριοποίηση δεδομένων είναι μια διαδικασία η οποία γίνεται σε δύο βήματα. Στο πρώτο βήμα, κτίζεται ένα μοντέλο το οποίο περιγράφει κλάσεις ενός συνόλου δεδομένων. Στο δεύτερο βήμα, το μοντέλο χρησιμοποιείται για κατηγοριοποίηση.

Οι αλγόριθμοι κατηγοριοποίησης μπορούν αν ορισθούν ως εξής:

Δεδομένου ενός δείγματος εκπαίδευσης $\{(x_1, y_1),...,(x_n, y_n)\}$ παράγεται ένας ταξινομητής $h : X \to Y$ ο οποίος αντιστοιχεί ένα αντικείμενο $x \in X$ στην κατηγορία $y \in Y$.

2.6.1 Ο αλγόριθμος C4.5

Ο αλγόριθμος αυτός ανήκει στους αλγορίθμους ταξινόμησης με δέντρα αποφάσεων. Τα δέντρα αποφάσεων είναι δεντρικές δομές τα οποία έχουν την μορφή ενός διαγράμματος ροής. Στα δένδρα αυτά, οι εσωτερικοί κόμβοι δηλώνουν έλεγχο (test) πάνω σε ένα χαρακτηριστικό και τα φύλλα αναπαριστούν τάξεις ή κατανομές τάξεων. Με τον αλγόριθμο αυτό, άγνωστα δείγματα μπορούν να ταξινομηθούν, ελέγχοντας τις τιμές των χαρακτηριστικών του δείγματος σε σχέση με το δένδρο αποφάσεως. Ένα μονοπάτι αρχίζει από τη ρίζα του δέντρου και φτάνει στο φύλο που κρατά την κλάση η οποία προβλέπεται για ένα συγκεκριμένο δείγμα. Τα δέντρα αποφάσεων μπορούν εύκολα να μετατραπούν σε κανόνες ταξινόμησης.

Ο C4.5 είναι ένας αλγόριθμος κατηγοριοποίησης ο οποίος λαμβάνει υπόψη μη διαθέσιμες τιμές, χαρακτηριστικά με συνεχόμενες τιμές εμβέλειας, το κλάδεμα δέντρων αποφάσεων, παραγωγή κανόνων και άλλα. Πιο συγκεκριμένα ο αλγόριθμος αυτός επιτυγχάνει τα πιο κάτω:

- Αποτρέπει την περίπτωση να μάθει περισσότερο από ότι πρέπει (overfitting the data) καθορίζοντας πόσο βαθιά θα πρέπει να προχωρήσει μέσα στο δέντρο.
- Μειώνει το σφάλμα που πιθανότατα να έχουμε κατά το κλάδεμα του δέντρου.
- Δημιουργεί τους κανόνες και μετά κλαδεύει το δέντρο.
- Χειρίζεται συνεχόμενα χαρακτηριστικά (πχ. θερμοκρασία).
- Επιλέγει ένα κατάλληλο μέτρο επιλογής χαρακτηριστικών.

- Χειρίζεται δεδομένα εκπαίδευσης με ελλιπής τιμές.
- Χειρίζεται χαρακτηριστικά με διαφορετικό κόστος.
- Βελτιώνει την υπολογιστική αποδοτικότητα.

Μια γενική προσέγγιση του αλγορίθμου είναι η πιο κάτω:

- Το δέντρο αρχίζει με ένα κόμβο ο οποίος αναπαριστά το σύνολο εκπαίδευσης

- Εάν όλα τα δείγματα ανήκουν στην ίδια κλάση , τότε ο κόμβος γίνεται φύλλο και χαρακτηρίζεται από αυτή την τάξη.
  Αλλιώς:

  1. Επέλεξε ένα χαρακτηριστικό το οποίο έχει την μεγαλύτερη διαχωριστική ικανότητα στις τιμές εξόδου. Το χαρακτηριστικό αυτό λέγεται και χαρακτηριστικό ελέγχου και επιλέγεται με βάση το κριτήριο κέρδους πληροφορίας (information gain).

  2. Δημιούργησε μια διαφορετική διακλάδωση για κάθε τιμή του επιλεγμένου χαρακτηριστικού.

  3. Διαχώρισε τα δείγματα (instances) σε υποσύνολα έτσι ώστε να ανταποκρίνονται στις τιμές του επιλεγμένου κόμβου.
  4. Για κάθε υποσύνολο, τερμάτισε την διαδικασία επιλογής εάν:
     a. Όλα τα μέλη ενός υποσυνόλου έχουν την ίδια τιμή για το χαρακτηριστικό εξόδου, τερματίζουν την διαδικασία επιλογής χαρακτηριστικού για το τρέχον μονοπάτι με την καθορισμένη τιμή.
     b. Το υποσύνολο περιέχει ένα κόμβο ή δεν μπορούν να καθοριστούν άλλα χαρακτηριστικά διαχωρισμού. Όπως και στο (α) χαρακτηρίζουμε μια διακλάδωση με την τιμή εξόδου η οποία χαρακτηρίζει την πλειοψηφία των δειγμάτων που παραμένουν.

  5. Για κάθε υποσύνολο το οποίο δημιουργείται στο (3) το οποίο δεν έχει χαρακτηριστεί σαν τερματικό, επαναλαμβάνει την πιο πάνω διαδικασία. Δεν χρειάζεται να λαμβάνονται υπόψη τα χαρακτηριστικά τα οποία βρίσκονται σε ένα κόμβο.

Ο πιο πάνω αλγόριθμος εφαρμόζεται στα δεδομένα εκπαίδευσης και το παραγόμενο δέντρο ελέγχεται πάνω σε ένα σύνολο δεδομένων για έλεγχο.

2.6.2 Κριτήριο Επιλογής Χαρακτηριστικού Ελέγχου

Το πιο σημαντικό μέρος του πιο πάνω αλγορίθμου είναι ο τρόπος με τον οποίο επιλέγουμε το χαρακτηριστικό που θα διαχωρίσει τα δείγματα. Το κριτήριο με το οποίο επιλέγουμε το χαρακτηριστικό αυτό λέγεται κριτήριο κέρδους πληροφορίας.

Το κριτήριο κέρδους πληροφορίας (information gain) χρησιμοποιείται για επιλογή του χαρακτηριστικού ελέγχου για κάθε κόμβο στο δέντρο. Ένα τέτοιο κριτήριο αναφέρεται και σαν *κριτήριο επιλογής χαρακτηριστικού*. Με βάση το κριτήριο αυτό, το χαρακτηριστικό με το μεγαλύτερο information gain (ή μεγαλύτερη μείωση εντροπίας) επιλέγεται σαν το χαρακτηριστικό ελέγχου (διαχωρισμού) για τον τρέχον κόμβο. Αυτό το χαρακτηριστικό ελαχιστοποιεί την πληροφορία που χρειάζεται για να ταξινομήσει τα δείγματα στα μέρη που προκύπτουν. Μια τέτοια θεωρητική προσέγγιση ελαχιστοποιεί τον αναμενόμενο αριθμό των ελέγχων που χρειάζονται για να ταξινομήσουν ένα αντικείμενο και εγγυάται ότι ένα απλό (αλλά όχι απαραίτητα το πιο απλό) δέντρο θα βρεθεί.

Έστω ότι έχουμε ένα σύνολο S το οποίο αποτελείται από s δείγματα δεδομένων. Υποθέτουμε ότι το χαρακτηριστικό απεικόνισης τάξης έχει m διακριτές τιμές οι οποίες καθορίζουν m διακριτές τάξεις, C_i (for i=1, … ,m). Έστω ότι s_i είναι ο αριθμός των δειγμάτων του S στην τάξη C_i. Η αναμενόμενη πληροφορία (expected information) η οποία χρειάζεται για να ταξινομήσει ένα δοθέν δείγμα δίνεται από τον τύπο:

$$I(S_1,...,S_m) = -\sum_{i=1}^{m} p_{ij} \log_2 p_{ij}$$

όπου p_i είναι η πιθανότητα ένα αυθαίρετο χαρακτηριστικό να ανήκει στην τάξη Ci και υπολογίζεται από το s_i.

Έστω ότι το χαρακτηριστικό *A* έχει *v* τιμές, $\{a_1, a_2, \dots, a_v\}$. Το χαρακτηριστικό *A* μπορεί να χρησιμοποιηθεί για να διαχωρίσει το S σε v υποσύνολα, $\{S_1, S_2, \dots, S_v\}$, όπου S_j περιέχει αυτά τα δείγματα στο S το οποίο έχει τιμή a_j για το A. Εάν το A επιλέγεται σαν το χαρακτηριστικό ελέγχου (i.e, το καλύτερο χαρακτηριστικό για διαχωρισμό), τότε αυτά τα υποσύνολα ανταποκρίνονται στις διακλαδώσεις οι οποίες αρχίζουν από τον κόμβο που περιέχει το σύνολο S. Έστω ότι το S_{ij} είναι ο αριθμός

των δειγμάτων της τάξης C_i σε ένα υποσύνολο S_j. Η εντροπία ή η αναμενόμενη πληροφορία που βασίζεται στο διαχωρισμό υποσυνόλων βάση του A δίνεται από

$$E(A) = \sum_{j=1}^{\nu} \frac{S_{1j} + \ldots + S_{mj}}{S} I\left(s_{1j}, \ldots, s_{mj}\right)$$

Ο όρος $\frac{S_{1j} + \ldots + S_{mj}}{S}$ ενεργεί σαν το βάρος για το j_{th} υποσύνολο και είναι ο αριθμός των δειγμάτων στο υποσύνολο το οποίο διαιρείται από των αριθμό των δειγμάτων στο S. Επίσης ισχύει ότι:

$$\mathrm{I}(\mathrm{S}_1, \ldots, S_m) = -\sum_{i=1}^{m} p_{ij} \log_2 p_{ij}$$

όπου p_{ij} είναι η πιθανότητα ένα δείγμα στο S_j να ανήκει στην τάξη C*i*.

Η κωδικοποίηση πληροφορίας η οποία μπορεί να κερδίσουμε με διακλάδωση στο A είναι:

$$Gain(A) = I\left(s_1, s_2, \ldots, s_m\right) - E(A)$$

Με άλλα λόγια το Gain(A) είναι η αναμενόμενη μείωση της εντροπίας η οποία προκαλείται με το να γνωρίζουμε την τιμή του χαρακτηριστικού A.

Έτσι ο αλγόριθμος υπολογίζει την τιμή του κέρδους πληροφορίας (information gain) για κάθε χαρακτηριστικό. Το χαρακτηριστικό με τον μεγαλύτερο κέρδος πληροφορίας επιλέγεται σαν το χαρακτηριστικό ελέγχου (test) για το σύνολο S. Στη συνέχεια ένας νέος κόμβος δημιουργείται στο δέντρο με αυτό το χαρακτηριστικό ελέγχου. Κάθε τιμή που μπορεί να πάρει το χαρακτηριστικό αυτό, δημιουργεί και μια διακλάδωση που αρχίζει από αυτό το νέο κόμβο. Τα δείγματα κατανέμονται ανάλογα, σε αυτές τις διακλαδώσεις. [17, 19]

Κεφάλαιο 3

Το πρόβλημα της εξατομίκευσης.

3.1 Το πρόβλημα

3.1.1 Υπερφόρτωση Πληροφοριών

Τα τελευταία χρόνια το διαδίκτυο γνώρισε τρομακτική ανάπτυξη και έδωσε έτσι πρόσβαση σε ένα τεράστιο όγκο πληροφοριών. Η συσσώρευση όλων αυτών των πληροφοριών στο διαδίκτυο είχε σαν αποτέλεσμα την εύκολη πρόσβαση σε σχεδόν οποιαδήποτε πληροφορία ανά πάσα στιγμή. Ωστόσο ο όγκος αυτός είναι πλέον τόσό μεγάλος και αυξάνεται τόσο σημαντικά καθημερινά, που καθιστά την διαδικασίας εύρεσης της πληροφορία που πραγματικά χρειάζεται ο χρήστης αφάνταστα δύσκολη και χρονοβόρα. Αυτό, γιατί η εύρεση της κατάλληλης πληροφορίας πρέπει να αναζητηθεί από ένα τεράστιο όγκο πληροφοριών και να φιλτραριστεί από τον ίδιο τον χρήστη. Η όλη διαδικασία είναι ιδιαίτερα χρονοβόρα και χαώδης κάτι που ο σύγχρονος χρήστης, ο οποίος γίνεται όλο και πιο απαιτητικός σε ότι αφορά την ποιότητα των υπηρεσιών που του παρέχονται, είναι δύσκολο να ανεχθεί.

Για να μπορέσουν λοιπόν, εφαρμογές και υπηρεσίες διαδικτύου να ικανοποιήσουν τον απαιτητικό αυτό χρήστη και να επιβιώσουν στο χώρο του διαδικτύου, πρέπει να ακολουθήσουν διαδικασίες που θα απαλείψουν το πρόβλημα, ή τουλάχιστον θα το απαμβλύνουν.

Πολλαπλές προτάσεις έχουν γίνει για να μετριάσουν το πρόβλημα της υπερφόρτωσης του διαδικτύου με όλη αυτή την πληροφορία. Οι μηχανές αναζήτησης, είναι ίσως ο πιο γνωστός και ο πιο ευρύτερα διαδεδομένος τρόπος που χρησιμοποιήθηκε για την άμβλυνση του προβλήματος αυτού. Ωστόσο, οι μηχανές αναζήτησης είναι αποδοτικές στο να φιλτράρουν σελίδες που ικανοποιούν συγκεκριμένες επερωτήσεις. Δυστυχώς οι χρήστες βρίσκουν την χρήση επερωτήσεων για αναζήτηση όχι ιδιαίτερα εύκολη κι ακόμη περισσότερο όταν θα πρέπει να περιορίσουν την αναζήτησή τους σε συγκεκριμένες λέξεις-κλειδιά. Επιπλέον συνήθως το αποτέλεσμα της αναζήτησης είναι η επιστροφή ενός τεράστιου συνόλου από πληροφορίες από της οποίες ελάχιστες αντιπροσωπεύουν αυτό που αναζητά ο χρήστης κάτι που εισάγει την ανάγκη φιλτραρίσματος των αποτελεσμάτων που παρουσιάζονται στο χρήστη.

Η επόμενη λύση που προτάθηκε για την εξομάλυνση του προβλήματος της υπερφόρτωσης πληροφοριών, είναι το semantic web. Εδώ κάθε πληροφορία χαρακτηρίζεται από μεταδεδομένα τα οποία μας δίνουν σημαντικές πληροφορίες για το περιεχόμενό της. Έτσι το semantic web επιτρέπει πιο έξυπνες αναζητήσεις πάνω στα μεταδεδομένα που περιγράφουν τις πληροφορίες. Ούτε όμως αυτή η πρόταση είναι ικανή να δώσει ουσιαστική λύση στο πιο πάνω πρόβλημα. Αυτό γιατί η τεχνολογία του semantic web εξαρτάται άμεσα από το κατά πόσο ο παροχέας της πληροφορίας θα μπει στη διαδικασία να την περιγράψει χρησιμοποιώντας μεταδεδομένα. Ακόμη κι αν υπήρχε τρόπος να αναγκάσουμε την χρήση μεταδεδομένων για την περιγραφή νέων κόμβων πληροφοριών,που δεν υπάρχει, υπάρχει ήδη ένας τεράστιος αριθμός πληροφοριών που δεν έχουν περιγραφεί με μεταδεδομένα και που είναι αδύνατο να περιγραφούν. Έτσι το semantic web μπορεί να εφαρμοστεί μόνο σε ένα υποσύνολο πληροφοριών.

3.1.2 Υπερφόρτωση Πληροφοριών στο Ασύρματο Δίκτυο

Όπως έχουμε ήδη αναφέρει, με την ανάπτυξη του ασύρματου δικτύου το διαδίκτυο έχει ήδη επεκταθεί και χρησιμοποιηθεί σε αυτό, και πλέον οι χρήστες του ασύρματου δικτύου μπορούν να έχουν πρόσβαση σε οποιεσδήποτε πληροφορίες του παγκόσμιου ιστού. Ωστόσο οι περιορισμοί που επιβάλλει τόσο το ασύρματο δίκτυο όσο και οι κινητές συσκευές κάνουν την αναγκαιότητα για λύση στο πρόβλημα της υπερφόρτωσης πληροφοριών επιβεβλημένη.

Στο ασύρματο διαδίκτυο, η ανάγκη για εμφάνιση πληροφοριών σε κινητές συσκευές, δεν είναι κάτι που μπορεί να αμβλυνθεί σε σημαντικό βαθμό από μηχανές αναζήτησης ή από την τεχνολογία του semantic web. Εδώ χρειαζόμαστε κάτι ποιο δραστικό το οποίο θα περιορίσει τις επιλογές που θα εμφανιστούν στο χρήστη στο ελάχιστο. Οι επιλογές αυτές πρέπει να πληρούν τα κριτήρια του χρήστη, ωστόσο αυτό δεν είναι αρκετό. Το τι επιθυμεί ο χρήστης δεν μπορεί πάντα να εκφραστεί με μερικά κριτήρια αναζήτησης, και είναι σχεδόν αδύνατο για το χρήστη να εισάγει όλα τα κριτήρια για αυτό που ζητά. Άρα για να μπορέσουμε να φιλτράρουμε σωστά τα αποτελέσματα που θα εμφανίσουμε στο χρήστη, θα πρέπει να έχουμε πλήρη περιγραφή από αυτόν, της πληροφορίας που ψάχνει.

Οι παράμετροι που επηρεάζουν την παροχή πληροφοριών σε ένα χρήστη του ασύρματου διαδικτύου δεν περιορίζονται στην εμφάνιση πληροφοριών στο χρήστη με βάση τα κριτήρια που επιβάλει. Στο ασύρματο δίκτυο οι χρήστες μπορούν να έχουν πρόσβαση σε πληροφορίες και υπηρεσίες, ανεξάρτητα από την τοποθεσία στην οποία βρίσκονται. Η κινητικότητα του χρήστη στην περίπτωση αυτή, υποδηλώνει την ικανότητά του να μπορεί να συνδεθεί σε κάποιο παροχέα πληροφοριών και να παραμείνει συνδεδεμένος, καθώς κινείται. Αυτό απαιτεί την ανάπτυξη νέων μηχανισμών, οι οποίοι μπορούν να υποστηρίζουν επαρκώς την κινητικότητα του χρήστη και να διασφαλίζουν την πρόσβασή του σε πληροφορίες στο διαδίκτυο, οι οποίες λαμβάνουν υπόψη τόσο την αλλαγή στην τοποθεσία του χρήστη, όσο και την αλλαγή στο χρόνο κατά την διάρκεια της κίνησης του.

Είναι σημαντικό να διαχωρίσουμε τις ανάγκες ενός σταθερού χρήστη από το χρήστη σε επιτραπέζιο περιβάλλον (desktop environment). Στη περίπτωση του κινητού χρήστη τα δυναμικά χαρακτηριστικά του κινητού περιβάλλοντος, ο χρόνος, η τοποθεσία και η κατάσταση του χρήστη τη δεδομένη στιγμή που αλληλεπιδρά με ένα σύστημα, επηρεάζουν τις πληροφορίες που χρειάζεται απ' αυτό. Είναι φανερό, πως οι πιο πάνω παράγοντες δεν ίσχυαν μέχρι τώρα για τους σταθερούς χρήστες αλλά προστίθενται τώρα με την παρουσία κινητών χρηστών. Κι είναι ακόμη πιο έντονη λοιπόν τώρα η ανάγκη για εύρεση μεθοδολογιών που θα λάβουν υπόψη τις ιδιαιτερότητες του ασύρματου δικτύου και του κινητού χρήστη και θα παρέχουν σε αυτόν τις πληροφορίες που χρειάζεται και μόνο.

3.2 Η εξατομίκευση

Τη λύση στο πρόβλημα της υπερφόρτωσης, έρχεται να δώσει η έννοια της εξατομίκευσης των δεδομένων. Η εξατομίκευση περιλαμβάνει τη διαδικασία της

συγκέντρωσης πληροφοριών για τα ενδιαφέροντα του χρήστη, κατά την αλληλεπίδραση του με κάποιο σύστημα, και στη συνέχεια την εμφάνιση της κατάλληλης πληροφορίας σε αυτόν, με βάση τα ενδιαφέροντά του.

3.2.1 Μέθοδοι Εξατομίκευσης

Η τεχνική της εξατομίκευσης επικεντρώνεται στην ιδέα ότι κάθε χρήστης είναι διαφορετικός, και άρα χρειάζεται διαφορετικές πληροφορίες. Η πιο απλή αλλά και η πιο αποδοτική, ίσως, μέθοδος για την πρόβλεψη του τι χρειάζεται ο κάθε χρήστης βασίζεται στο γεγονός ότι οι χρήστες συχνά "επαναλαμβάνουν τον εαυτό τους" [20]. Αυτό σημαίνει ότι τις περισσότερες φορές οι χρήστες ζητούν πληροφορίες για παρόμοια, αν όχι τα ίδια, δεδομένα. Έτσι, η λύση της εξατομίκευσης στηρίζεται στη μελέτη προηγούμενων ενεργειών του χρήστη, για να προβλέψει την επόμενη.

Υπάρχουν δύο βασικές τεχνικές εξατομίκευσης: Η πρώτη μελετά το περιεχόμενο που ζητά ο χρήστης (Content-based method) και δημιουργεί αναπαραστάσεις που περιλαμβάνουν συγκεκριμένα ενδιαφέροντα του χρήστη. Η δεύτερη μελετά τις προτιμήσεις των χρηστών σε συγκεκριμένα προϊόντα και υπηρεσίες. κατηγοριοποιώντας τους σε ομάδες με κοινά ενδιαφέροντα (Collaborative method).

3.2.1.1 Content Based Methods

Στη μεθοδολογία αυτή, η πρόβλεψη στηρίζεται στη μελέτη του περιεχομένου. Συγκεκριμένα, αναλύει το περιεχόμενο της πληροφορίας για να σχηματίσει μια αναπαράσταση των ενδιαφερόντων του χρήστη. Στηρίζεται στη συσχέτιση μεταξύ του περιεχομένου της πληροφορίας που ζήτησε ο χρήστης και της αναπαράστασης των ενδιαφερόντων του. Συγκεκριμένα οι μέθοδοι αυτοί, μελετούν την συμπεριφορά του χρήστη εξετάζοντας το περιεχόμενο προηγούμενων ενεργειών του και ακολούθως, τη συσχετίζει με πληροφοριακούς κόμβους που θα μπορούσαν να εμφανιστούν στο χρήστη. Ανάλογα με αυτή τη συσχέτιση θα κριθεί κατά πόσο ο κόμβος είναι σημαντικός για τον χρήστη ή όχι. [21]

Για την υλοποίηση της μεθόδου αυτής χρησιμοποιούμε συνήθως προφίλ χρηστών, όπου κρατούμε πληροφορίες για τα ενδιαφέροντα τους. Στη συνέχεια, γίνεται μια προσπάθεια πρόβλεψης των αναγκών τους. Για τη δημιουργία ενός προφίλ μπορούμε είτε να πάρουμε δεδομένα κατευθείαν από το χρήστη ζητώντας του να συμπληρώσει κάποια ενδιαφέροντά του, είτε με τη μελέτη των κινήσεών του και τροποποίηση του προφίλ του βάση αυτών. Στη πρώτη περίπτωση, η διαδικασία είναι

αρκετά απλή αλλά απαιτείται κάποια επιπλέον προσπάθεια και χρόνος από το χρήστη, για να μας δώσει της πληροφορίες που χρειαζόμαστε. Στη δεύτερη περίπτωση, για την συλλογή των δεδομένων που χρειαζόμαστε, ώστε να μπορούμε να προβλέψουμε μελλοντικές κινήσεις του χρήστη, απαιτείται η ανάλυση των κινήσεων του. Ανάλογα με την εφαρμογή που αναπτύσσεται και το είδος των πληροφοριών που χρειάζεται να παρέχει, υπάρχουν διάφοροι τρόποι ανάλυσης των κινήσεων αυτών. Οι τρόποι αυτοί μπορεί να περιλαμβάνουν αναζήτηση του λογικού περιεχομένου που ζητά ο χρήστης, καταγραφή των υπερσυνδέσμων (links) που επισκέφτηκε, καταγραφή της διαδρομής που ακολούθησε για να φτάσει σε ένα πληροφοριακό κόμβο, και τον τελευταίο προορισμό του πριν εγκαταλείψει την ιστοσελίδα. Στην όλη διαδικασία μπορεί να ληφθούν υπόψη ο χρόνος που διάνυσε μέσα σε μια ιστοσελίδα ή και ακόμη οι θέσεις των υπερσυνδέσμων και κουμπιών (button) στη σελίδα. Η δεύτερη αυτή περίπτωση απαιτεί πολύ λιγότερη προσπάθεια από το χρήστη, αφού όλα σχεδόν πρέπει να καταγράφονται αυτόματα από το σύστημα[18]. Και οι δύο πιο πάνω περιπτώσεις έχουν όμως ένα σκοπό· να κτίζουν μοντέλα, τα οποία συσχετίζουν πληροφορίες για το περιεχόμενο των πληροφοριών που βρίσκεται στις προτιμήσεις του χρήστη σε σχέση με τις πληροφορίες αυτές.

Το φιλτράρισμα της πληροφορίας με βάση το περιεχόμενο, είναι πιο κατάλληλο όταν το είδος της πληροφορίας που αναλύεται, μπορεί εύκολα να αναλυθεί από μια υπολογιστική μηχανή και όταν η άποψη του χρήστη για την καταλληλότητα της πληροφορίας δεν είναι υποκειμενική. Επιπλέον, η μέθοδος αυτή, μειονεκτεί στο γεγονός ότι μέθοδοι για φιλτράρισμα περιεχομένου με αυτό τον τρόπο μπορούν να εφαρμοστούν σε περιορισμένα είδη περιεχομένου (κείμενο και εικόνες). Επιπλέον η χρήση προφίλ χρηστών για φιλτράρισμα δεδομένων καθώς και η πρόβλεψη που βασίζεται στο ιστορικό του χρήστη, αιχμαλωτίζει μόνο μια διάσταση της συμπεριφοράς του. Συγκεκριμένα, δεν δίνεται η δυνατότητα στους χρήστες να πάρουν πληροφορίες για κάτι το οποίο δεν είναι παρόμοιο με ό,τι περιλαμβάνει το προφίλ του[21, 22].

3.2.1.2 Collaborative Methods

Οι μέθοδοι αυτοί συγκρίνουν τις προτιμήσεις ενός χρήστη με τις προτιμήσεις άλλων χρηστών. Μ’ αυτό τον τρόπο επιδιώκουν να κατηγοριοποιήσουν τους χρήστες με βάση τα κοινά τους ενδιαφέροντα, και κατ’ επέκταση να προβλέψουν τις κινήσεις τους λαμβάνοντας υπόψη τις κινήσεις άλλων χρηστών που ανήκουν στην ίδια κατηγορία με αυτούς. Η μέθοδος αυτή στηρίζεται στην ιδέα ότι αρκετές φορές παίρνουμε κάποιες αποφάσεις επηρεαζόμενοι από γνώμες και αξιολογήσεις άλλων

ατόμων, των οποίων εκτιμούμε τις απόψεις [20]. Τα περισσότερα συστήματα που χρησιμοποιούν τη μέθοδο αυτή ακολουθούν τα επόμενα τρία βήματα:

- Καταγράφουν τις προτιμήσεις ενός μεγάλου συνόλου χρηστών
- Επιλέγουν την ομάδα των χρηστών των οποία τα ενδιαφέροντα είναι όμοια με του χρήστη στον οποίο προσπαθούν να προτείνουν υπηρεσίες
- Εισηγούνται επιλογές στο χρήστη, τις οποίες φαίνεται να προτιμούν άλλοι χρήστες που ανήκουν στην ίδια ομάδα με τον χρήστη.

Υπάρχουν δύο είδη collaborative filtering, ο ένας τρόπος βασίζεται στο χρήστη και συνήθως αναφερόμαστε σε αυτόν σαν user-based και ο άλλος βασίζεται στα αντικείμενα που προτείνονται και ονομάζεται item-based. Ο πρώτος τρόπος, εισηγείται υπηρεσίες και αντικείμενα στο χρήστη λαμβάνοντας υπόψη μόνο της βαθμολογίες των χρηστών σε συγκεκριμένα αντικείμενα/υπηρεσίες. Για να το κάνει αυτό χρησιμοποιεί προφίλ χρηστών και βρίσκει ομοιότητες μεταξύ των χρηστών για να μπορέσει να παρουσιάσει αποτελέσματα στον χρήστη που θα τον ενδιαφέρουν. Η δεύτερη μέθοδος επεξεργάζεται δεδομένα που αφορούν τα αντικείμενα και τις υπηρεσίες που προτείνει. Στη συνέχεια βρίσκει συσχετίσεις μεταξύ των αντικειμένων αυτών τις οποίες χρησιμοποιεί για να τις εισηγήσεις του στο χρήστη.

Συνήθως, η συλλογή των δεδομένων στην περίπτωση αυτή γίνεται με αξιολόγηση (rating) δεδομένων από τον χρήστη. Δηλαδή, οι χρήστες αξιολογούν κατά πόσο μια πληροφορία που τους δόθηκε ήταν αυτό που αναζητούσαν. Υπάρχουν όμως περιπτώσεις που μπορεί η συλλογή πληροφοριών για το χρήστη να γίνεται και πάλι μελετώντας προηγούμενες ενέργειές τους. Κοινός στόχος και των δύο αυτών τρόπων συλλογής των δεδομένων είναι η δημιουργία μοντέλων, τα οποία συσχετίζουν πληροφορίες για τις προτιμήσεις άλλων χρηστών με τις προτιμήσεις του ζητούμενου χρήστη.

Αυτός ο τρόπος, είναι ιδανικός για συγκεκριμένου τύπου πληροφορίες, όπως για παράδειγμα βιβλία, ταινίες, μουσική κλπ. Ωστόσο δεν φαίνεται να είναι αποδοτικός με άλλες κατηγορίες. Αυτό γιατί βασίζεται στις κινήσεις των χρηστών για να προβλέψει τις προτιμήσεις τους. Με αυτό το τρόπο ο χρήστης περιορίζεται στο να δείξει στο σύστημα τι τύπο προϊόντων η υπηρεσιών αναζητεί, χωρίς όμως να δίνει καμία πληροφορία για το γιατί. Επιπλέον μειονέκτημα της μεθόδου αυτής είναι το γεγονός ότι, στην περίπτωση που ζητούμε αξιολόγηση των αποτελεσμάτων από το χρήστη, η άμεση αυτή είσοδος δεδομένων από αυτόν οδηγεί στη σπατάλη χρόνου εκ μέρους του, στην προσπάθειά του να εξατομικεύσει τα αποτελέσματα που του

δόθηκαν [22]. Εκτός αυτού, η κατηγοριοποίηση των χρηστών μπορεί να δημιουργήσει προβλήματα κατά το φιλτράρισμα των δεδομένων. Κι αυτό γιατί ένα μικρό δείγμα χρηστών μπορεί να επιφέρει χαμηλής ποιότητας εισηγήσεις για τον χρήστη. Η ποιότητα των εισηγήσεων αυτών αυξάνεται όσο αυξάνονται και οι χρήστες. Ακόμη ένα αρνητικό της μεθόδου αυτής είναι η ανικανότητά της να κάνει εισηγήσεις σε ασυνήθιστους χρήστες, τους οποίους δεν μπορεί να κατατάξει σε κάποια από τις υπάρχουσες κατηγορίες χρηστών. Τέλος, το collaborating filtering μπορεί να είναι λιγότερο σημαντικό σαν τεχνική, όταν οι κατηγορίες των χρηστών και οι προτιμήσεις τους είναι γνωστές και καλά ορισμένες [23].

3.2.2 Χρήση Προφίλ Χρηστών για Εξατομίκευση

Τα προφίλ χρηστών είναι το κυριότερο συστατικό των περισσότερων εξατομικευμένων υπηρεσιών διαδικτύου. Συνήθως περιλαμβάνουν ένα σύνολο θεματικών επιλογών για το χρήστη, με τις οποίες μπορεί ο χρήστης να περιγράψει τα θέματα που τον ενδιαφέρουν. Έτσι, με τα προφίλ μπορούμε να εξάγουμε όλες τις προτιμήσεις και πληροφορίες για μια τυπική συμπεριφορά του χρήστη.

Τα δεδομένα που χρησιμοποιούνται για την κατασκευή των προφίλ χωρίζονται σε δύο κατηγορίες [26]:

- Δημογραφικά (demographic) : Τα δεδομένα αυτά καθορίζουν ποιος είναι ο χρήστης
- Δεδομένα Συναλλαγών (Transactional) : Τα δεδομένα αυτά καθορίζουν τι κάνει ο χρήστης και δίνουν στοιχεία της συμπεριφοράς του.

Είναι σημαντικό το προφίλ ενός χρήστη να μπορεί να μας δώσει πληροφορίες και για τους δύο πιο πάνω τύπους δεδομένων. Τέτοια δεδομένα αυτά μπορεί να παραχθούν είτε απ' ευθείας από το χρήστη, είτε με παρακολούθηση προηγούμενων συμπεριφορών του.

Τα προφίλ των χρηστών συνήθως είτε βασίζονται στη γνώση (knowledge based) είτε στη συμπεριφορά (behaviour based). Η πρώτη κατηγορία περιλαμβάνει τη δημιουργία στατικών μοντέλων χρηστών και δυναμικά αντιστοιχεί τους χρήστες στο πιο κοντινό μοντέλο. Για να παρθεί η γνώση αυτή συνήθως χρησιμοποιούνται ερωτηματολόγια και συνεντεύξεις. Η δεύτερη κατηγορία περιλαμβάνει προσεγγίσεις οι οποίες χρησιμοποιούν τη συμπεριφορά του χρήστη σαν μοντέλο.

3.2.3 Τρόποι Αναπαράστασης των προφίλ

3.2.3.1 Με βαθμίδες αξιολόγησης (Ratings – Based)

Ο τρόπος αυτός χρησιμοποιείτε σε Collaborative methods, όπου οι χρήστες αφού πάρουν εισηγήσεις από το σύστημα ζητείται να παρέχουν πληροφορίες στο σύστημα για το πόσο ενδιαφέρον ήταν η πληροφορία που τους παρουσιάστηκε. Αυτή η πληροφορία που δίνεται στο σύστημα λέγεται relevance feedback. Το relevance feedback δίνεται από τον χρήστη, με το να προσφέρει σε αυτόν το σύστημα μια κλίμακα βαθμολόγησης για κάθε πληροφορία που εισηγείται στο χρήστη. Η επιλογή είναι συνήθως μια κλίμακα με 3 έως 5 επιλογές από «Καθόλου Ενδιαφέρον» μέχρι «Πολύ Ενδιαφέρον».

Το relevance feedback αναπαρίσταται σαν ένα σύνολο από τα αντικείμενα που εισηγείται το σύστημα στον χρήστη και οι αντίστοιχες τιμές που δίνει ο χρήστης στο σύστημα. Συνήθως όμως το relevance feedback είναι μη συμπληρωμένο καθότι οι χρήστες είναι απρόθυμοι να σπαταλήσουν χρόνο για να παρέχουν το feedback αυτό.

3.2.3.2 Αναπαράσταση με διανύσματα όρου-συχνότητας (Term-Frequency)

Στη μέθοδο αυτή το προφίλ αναπαρίσταται σαν ένα σύνολο από αντικείμενα-όρους με τα αντίστοιχα ποσοστά προτίμησης, $pr = \{\langle o_1, w_{o_1} \rangle, \langle o_2, w_{o_2} \rangle, \cdots, \langle o_n, w_{o_n} \rangle\}$, όπου o_i ο όρος ή το αντικείμενο στο οποίο αναφερόμαστε και w_{o_i} το βάρος προτίμησης ή συχνότητα για τον όρο αυτό. Οι όροι αντιπροσωπεύουν λέξεις ή και προτάσεις και το βάρος προτίμησης ή συχνότητα είναι συνήθως ο ρυθμός που μας δίνει όσες φορές επαναλαμβάνεται ο όρος στη πληροφορία που εξετάζουμε.

3.2.3.3 Δυαδική Αναπαράσταση

Στην αναπαράσταση αυτή, αναπαριστούμε τα ενδιαφέροντα του χρήστη χρησιμοποιώντας δυαδικά βάρη προτίμησης για να ορίσουμε το τι αρέσει και τι δεν αρέσει στο χρήστη. Τα χαρακτηριστικά που ενδιαφέρουν τον χρήστη παρουσιάζονται σαν διανύσματα όρων-συχνότητας τα οποία ο χρήστης βαθμολόγησε σαν «ενδιαφέρον». Αντίστοιχα οι όροι που δεν ενδιαφέρουν τον χρήστη,

βαθμολογούνται σαν «Μη ενδιαφέρον». Έτσι στην αναπαράσταση αυτή του προφίλ έχουμε δυαδικά βάρη προτίμησης τα οποία μπορεί να πάρουν μόνο δύο τιμές .

3.2.3.4 Αναπαράσταση προφίλ με χρήση οντολογιών.

Περιγραφές των δεδομένων μπορούν να αφομοιωθούν και να πρωτοτυποποιηθούν σαν μια δομημένη συλλογή, όπως ένα λεξιλόγιο ενός προφίλ. Η κατηγοριοποίηση αυτή γίνεται συνήθως με οντολογίες. Μια οντολογία είναι η δημιουργία ενός τομέα (domain) σε μια μορφή που είναι κατανοητή στον άνθρωπο και αναγνώσιμη από τις μηχανές. Η μορφή αυτή αποτελείται από οντότητες, χαρακτηριστικά, σχέσεις και αξιώματα [24]. Εφαρμόζοντας την τεχνική αυτή, πληροφορίες και υπηρεσίες διαδικτύου μπορούν να χαρακτηρισθούν χρησιμοποιώντας μετά-δεδομένα, τα οποία αντιπροσωπεύουν τις πληροφορίες που χρειάζεται ο χρήστης από το συγκεκριμένο πληροφοριακό κόμβο και που περιλαμβάνονται στο προφίλ του. Έτσι επιτυγχάνουμε αποδοτικότερη αναζήτηση σε ό,τι αφορά τις πληροφορίες και τις υπηρεσίες που προκαλούν μεγαλύτερο ενδιαφέρον στο χρήστη [25].

3.2.4 Τρόποι Δημιουργίας των προφίλ

Για τη δημιουργία ενός προφίλ μπορούν να χρησιμοποιηθούν διάφορες μέθοδοι. Η πιο απλή στηρίζεται στην απλή εισαγωγή πληροφοριών από το χρήστη. Οι περισσότερες βασίζονται στην ανάλυση του περιεχομένου που φαίνεται να προτιμά ο χρήστης. Εντούτοις, υπάρχουν και τεχνικές οι οποίες στηρίζονται σε κανόνες φιλτραρίσματος. Μερικές από τις μεθόδους αυτές δίνονται πιο κάτω [26]:

- User-Created Profile: Η μέθοδος αυτή είναι η πιο απλή. Εδώ ο χρήστης καλείται να επιλέξει τα ενδιαφέροντά του, πιθανότατα από κάποια λίστα με πληροφορίες. Οι πληροφορίες που καθορίζονται από το χρήστη σαν πιο ενδιαφέρουσες χρησιμοποιούνται κατά τη διαδικασία του φιλτραρίσματος.

- System-Created profile by Automatic Indexing: Στην τεχνική αυτή δεδομένα που έδωσε ο χρήστης αναλύονται από κάποιο λογισμικό, το οποίο, με βάση την συχνότητα που εμφανίζονται κάποιες πληροφορίες, θέτει βάρη προτίμησης για τον συγκεκριμένο χρήστη.

- System-plus User Created Profile: Η μεθοδολογία αυτή συνδυάζει τις δύο πιο πάνω μεθόδους. Συγκεκριμένα, αρχικά δημιουργείται αυτόματα ένα προφίλ . Στη

συνέχεια, ο χρήστης μπορεί να αλλάξει το προφίλ αυτό με βάση τα δικά του ενδιαφέροντα.

- System-Created Profile based on Learning by Artificial Neutral-Network (ANN): Βασική ιδέα στη μέθοδο αυτή είναι η εκπαίδευση ενός δικτύου τεχνητής νοημοσύνης, το οποίο βασίζεται σε δεδομένα που έχει ήδη επιλέξει ο χρήστης.

- User-Profile Inherited from a User-Stereotype: Στην περίπτωση αυτή το σύστημα έχει κάποια προκαθορισμένα προφίλ. Εδώ, χρήστες με κοινά ενδιαφέροντα και κοινή συμπεριφορά φιλτραρίσματος έχουν κοινά προφίλ.
- Rule-Based Filtering: Ένα προφίλ που βασίζεται στη τεχνική αυτή αποτελείται από ένα σύνολο κανόνων φιλτραρίσματος. Για να το πετύχει αυτό η μέθοδος αυτή ζήτα από το χρήστη πληροφορίες όχι μόνο για τα ενδιαφέροντά του αλλά και για τη συμπεριφορά του.

3.2.5 Αλγόριθμοι για την Δημιουργία, Ενημέρωση και Διατήρηση του προφίλ

Στη συνέχεια θα δούμε μερικές τεχνικές που έχουν χρησιμοποιηθεί για την διατήρηση και ενημέρωση προφίλ χρηστών. Οι τεχνικές αυτές χωρίζονται σε δύο κυρίως κατηγορίες, αυτές που χρησιμοποιούν αλγορίθμους συσχέτισης για να ενημερώσουν κάποιο ποσοστό προτίμησης, και αυτές που χρησιμοποιούν αλγορίθμους εξόρυξης δεδομένων για το σκοπό αυτό. Βασίζονται σε μεθόδους collaborative filtering, Contend-based filtering αλλά και υβριδικές μεθόδους που συνδυάζουν και τα δύο.

Επιπλέον, Οι μέθοδοι αυτοί χωρίζονται σε στατικές και δυναμικές. Στις στατικές το προφίλ του χρήστη και τα δεδομένα που αντιπροσωπεύουν τις προτιμήσεις του, αλλάζουν σπάνια ή καθόλου. Στις δυναμικές το προφίλ των χρηστών ενημερώνεται συχνά. Στη περίπτωση των δυναμικών προφίλ, υπάρχουν ακόμη δύο περιπτώσεις. Στη μία, χειριζόμαστε τους χρήστες σαν ομάδες και κρατάμε για αυτούς ομαδικά προφίλ, στη δεύτερη χειριζόμαστε οποιαδήποτε αλλαγές στη συμπεριφορά ενός χρήστη ατομικά. Οι τρόποι που χειριζόμαστε το προφίλ εξαρτάται από ποια μέθοδο εξατομίκευσης ακολουθούμε. Εάν ακολουθούμε collaborative filtering τότε χρησιμοποιούμε ομαδικά προφίλ. Εάν ακολουθούμε content-based τότε χρησιμοποιούμε ομαδικά προφίλ.

Time-Decay

Η τεχνική αυτή χρησιμοποιείται σε προφίλ που χρησιμοποιούν μεθόδους βαθμολόγησης και content-based. Χρησιμοποιεί μια συνάρτηση με τη οποία δίνουμε λιγότερη βαρύτητα σε προηγούμενα στοιχεία που είχαμε για έναν όρο, χωρίς όμως να τα αγνοεί τελείως. Η συνάρτηση αυτή εφαρμόζεται κατά την βαθμολόγηση των όρων που υπάρχουν στο προφίλ ενός χρήστη και έτσι οι όροι αυτοί ενημερώνονται με το νέο ποσοστό προτίμησης, το οποίο είναι συνάρτηση του παλιού. Η συνάρτηση αυτή δίνεται πιο κάτω:

$$w(t_i) = \sum_{j=1}^{N} \frac{tf(t_i,d_j)}{age(d_j)}$$

$w(t_i)$	weight of term t_i after time decay
t_i	i^{th} term
N	number of documents
$tf(t_i,d_j)$	number of times term t_i appears in document d_j
d_j	j^{th} document
$age(d_j)$	age of document d_j

TF-IDF

Η συνάρτηση αυτή χρησιμοποιείται συνήθως σε εφαρμογές ανάκτησης πληροφορίας (informational retrieval) και εξόρυξης κειμένων. Αυτός ο όρος είναι ένα στατιστικό μέτρο που χρησιμοποιείται για να αξιολογήσει πόσο σημαντική είναι μια λέξη σε ένα κείμενο. Η σημαντικότητα του όρου αυξάνεται σε σχέση με το πόσες φορές εμφανίζεται η λέξη αυτή στο κείμενο σε σχέση με την αναλογία της λέξης αυτής μέσα στη γραμματική όλου του κειμένου. Η συνάρτηση αυτή καθορίζει με το τρόπο αυτό, πόσο σχετική είναι μια δεδομένη λέξη με το συγκεκριμένο κείμενο. Ο αλγόριθμος αυτός όπως είναι φανερό, χρησιμοποιείται σε περιπτώσεις εξατομίκευσης κυρίως σε κείμενα.

$$w(t_i,d_j) = tf(t_i,d_j) * \log \frac{N}{df(t_i)}$$

$w(t_i,d_j)$	tf-idf weight of term t_i in document d_j
t_i	i^{th} term
d_j	j^{th} document
$tf(t_i,d_j)$	number of times term t_i appears in document d_j
N	number of documents
$df(t_i)$	number of documents containing term t_i

Τεχνικές Εξόρυξης Δεδομένων

Όταν χρησιμοποιούμε ομαδικά προφίλ για την αναπαράσταση της συμπεριφοράς και των ενδιαφερόντων του χρήστη, η δημιουργία και ενημέρωση των προφίλ στηρίζεται σε κανόνες και πρότυπα τα οποία εξάγουμε εφαρμόζοντας τεχνικές εξόρυξης δεδομένων στις εγγραφές των εξυπηρετητών. Οι τεχνικές αυτές μπορεί να συμπεριλαμβάνουν, ανάλυση των εγγραφών (Log Analysis), κανόνες συσχέτισης (Association rules), ανακάλυψη διαδοχικών προτύπων (sequential pattern discovery), συσταδοποίηση (clustering) και κατηγοριοποίηση (classification).

Οι τεχνικές ανάλυσης των εγγραφών ενός εξυπηρετητή (Logs Analysis), παίρνουν σαν είσοδο δεδομένα διαδικτύου και τα επεξεργάζονται ώστε να εξάγουν στατιστικές πληροφορίες. Τέτοιες πληροφορίες συμπεριλαμβάνουν την δραστηριότητα σε μια ιστοσελίδα (αριθμός επισκέψεων, χρόνος παραμονής στην σελίδα, επιτυχής/ αποτυχής επισκέψεις κλπ), διαγνωστικά στατιστικά (πόσα λάθη του εξυπηρετητή και πόσες μη ανευρεθείς σελίδες εμφανίστηκαν στους χρήστες), στατιστικές εξυπηρετητή (σελίδες που επισκέπτονται πιο συχνά οι χρήστες, μηχανές αναζήτησης και λέξεις κλειδιά), δημογραφικά δεδομένα του χρήστη, στατιστικά πελάτη (λειτουργικό πελάτη, είδος φυλλομετρητή, μπισκοτάκια - cookies)

Οι κανόνες συσχετίσεων, είναι μια τεχνική που βρίσκει συχνά πρότυπα, σχέσεις και συσχετισμούς ανάμεσα στα δεδομένα. Οι κανόνες συσχέτισης, χρησιμοποιούνται ώστε να φανερώσουν συσχετισμούς μεταξύ ιστοσελίδων τις οποίες επισκέπτεται ο χρήστης σε ένα session. Τέτοιοι κανόνες εμφανίζουν τις πιθανές σχέσεις μεταξύ των ιστοσελίδων που συνήθως ο χρήστης επισκέπτεται σε ένα session. Οι σχέσεις αυτές, μπορούν να φανερώσουν και σχέσεις μεταξύ ομάδων χρηστών με συγκεκριμένα ενδιαφέροντα.

Η ανακάλυψη διαδοχικών πρότυπων είναι μια επέκταση της συσχέτισης κανόνων εξόρυξης. Αυτό γιατί φανερώνουν πρότυπα τα οποία επανεμφανίζονται, ενσωματώνοντας με το τρόπο αυτό την έννοια της διαδοχής του χρόνου (time sequence). Στο διαδίκτυο, ένα τέτοιο πρότυπο μπορεί να είναι μια ιστοσελίδα, ή ένα σύνολο ιστοσελίδων τις οποίες ο χρήστης επισκέπτεται διαδοχικά. Με την μέθοδο αυτή μπορούν να ανακαλυφθούν τάσεις των χρηστών και μπορούν να εξαχθούν πρότυπα που έχουν να κάνουν με την πρόβλεψη των επισκέψεων σε μια σελίδα.

Η συσταδοποίηση χρησιμοποιείται για να ομαδοποιήσουμε αντικείμενα που έχουν κοινά χαρακτηριστικά. Στη περίπτωση του διαδικτύου έχουμε δύο περιπτώσεις

συσταδοποίησης, συσταδοποίηση χρηστών και συσταδοποίηση σελίδων. Με την συσταδοποίηση ιστοσελίδων αναγνωρίζουμε σελίδες που φαίνεται να είναι εννοιολογικά συσχετιζόμενες με βάση την αντίληψη του χρήστη. Η συσταδοποίηση χρηστών, έχει σαν αποτέλεσμα την δημιουργία ομάδων χρηστών που φαίνεται να συμπεριφέρονται παρόμοια όταν επισκέπτονται ένα ιστιακό χώρο.

Η κατηγοριοποίηση είναι η διαδικασία η οποία αντιστοιχεί δεδομένα σε προκαθορισμένες κλάσεις. Στη περίπτωση του διαδικτύου, οι κλάσεις αυτές αντιπροσωπεύουν διαφορετικά προφίλ χρηστών και η κατηγοριοποίηση γίνεται επιλέγοντας χαρακτηριστικά που περιγράφουν κάθε μία κατηγορία χρηστών. Οι πιο γνωστοί αλγόριθμοι κατηγοριοποίησης είναι τα δέντρα αποφάσεων (Decision Trees), ο κατηγοριοποιητή Bayessian (Bayesian classifier), τα νευρωνικά δίκτυα και άλλα. [30, 31]

3.3 Μειονεκτήματα Λύσεων

Όλες οι πιο πάνω μέθοδοι που έχουν αναφερθεί μπορούν να χρησιμοποιηθούν αποδοτικά για τη εξόρυξη δεδομένων και την δημιουργία και ενημέρωση των προφίλ χρηστών. Ωστόσο, όλοι οι πιο πάνω αλγόριθμοι, περιορίζονται στο να μελετήσουμε μόνο το περιεχόμενο των πληροφοριακών κόμβων και να τα συσχετίσουν είτε μεταξύ τους, είτε με τα ενδιαφέροντα των χρηστών.

Ακόμη και σε έρευνες που έχουν γίνει συγκεκριμένα για κινητούς χρήστες [31, 32, 33, 34], τα χαρακτηριστικά που εμφανίζονται στο ασύρματο διαδίκτυο για ένα κινητό χρήστη αγνοούνται παντελώς. Σε μερικές περιπτώσεις έχουμε αναφορές για εξατομίκευση που λαμβάνει υπόψη την τοποθεσία του χρήστη και προσπαθούν να εμφανίσουν σε αυτό υπηρεσίες που βρίσκονται τοπικά κοντά του [34, 36], ωστόσο ο χρόνος και η κατάσταση του χρήστη και πάλι αγνοούνται.

Σχεδόν καμία από τις λύσεις που προτάθηκαν έως τώρα δεν μελετά τα χαρακτηριστικά του κινητού χρήστη. Οι λύσεις που έχουν δοθεί στο πρόβλημα εξατομίκευσης γενικά ασθενούν να εφαρμοστούν, ωστόσο, στο ασύρματο δίκτυο. Αυτό γιατί οι παράγοντες του χρόνου και της δραστηριότητας του χρήστη μπορεί να επηρεάσουν στο μέγιστο βαθμό το τι χρειάζεται ο χρήστης. Για να γίνει αυτό περισσότερο κατανοητό περιγράφουμε το πιο κάτω σενάριο:

Ο χρήστης Α, βρίσκεται στο γραφείο του, είναι μεσημέρι και μόλις έχει τελειώσει μια εργασία που θα παρουσιάσει σε λιγότερο από μια ώρα στους συναδέλφους τους. Επίσης γνωρίζουμε ότι ο χρήστης έχει ανάγκη από φαγητό και ότι στο

χρήστη αυτό αρέσουν ιδιαίτερα τα κινέζικα εστιατόρια. Το ερώτημα είναι, ένα σύστημα εξατομίκευσης θα πρέπει να παρουσιάσει στο χρήστη τα καλύτερα κινέζικα εστιατόρια της πόλης ή το πιο κοντινό φαστφουταδικο.

Στο πιο πάνω σενάριο, ένα σύστημα εξατομίκευσης το οποίο αγνοεί τους παράγοντες χρόνος και δραστηριότητα του χρήστη, θα του παρουσιάσει μια λίστα από τα καλύτερα κινέζικα εστιατόρια της πόλης. Ωστόσο, η κατάσταση που βρίσκεται ο συγκεκριμένος χρήστης και ο χρόνος στον οποίο βρίσκεται στην κατάσταση αυτή, δεν του επιτρέπουν να αναζητήσει για μεσημεριανό ένα καλό κινέζικο εστιατόριο. Ο χρήστης μας βιάζεται, και δεν θα δώσει ιδιαίτερη σημασία στο είδος του φαγητού απλά θα αναζητήσει κάτι για γρήγορο, και όσο το δυνατό πιο κοντινό μεσημεριανό. Άρα μια λίστα με τα καλύτερα κινέζικα της πόλης του είναι ουσιαστικά άχρηστη.

Κεφάλαιο 4

Εξατομίκευση στο Ασύρματο Δίκτυο

Η ανάπτυξη του ασύρματου δικτύου τα τελευταία χρόνια έχει δώσει την ευκαιρία στους κινητούς χρήστες να έχουν πρόσβαση σε ένα τεράστιο δίκτυο πληροφοριών ανά πάσα στιγμή. Ωστόσο οι ήδη υπάρχον πληροφορίες στο διαδίκτυο δεν είναι σχεδιασμένες για να ανταποκρίνονται στους περιορισμούς που επιβάλλει το ασύρματο δίκτυο και οι κινητές συσκευές. Το δε πρόβλημα υπερφόρτωσης που υπάρχει στο διαδίκτυο, στους κινητούς χρήστες είναι ακόμη πιο εμφανές σε βαθμό που μπορεί ακόμη και να αποτρέψει τους κινητούς χρήστες από το να το χρησιμοποιήσουν εκτεταμένα. Οι έως τώρα προτάσεις για λύσεις του προβλήματος υπερφόρτωσης αποτυγχάνουν πλήρως ή μερικώς να εφαρμοστούν και στο ασύρματο δίκτυο. Αυτό γιατί αγνοούν και κατ' επέκταση δεν λαμβάνουν υπόψη τα νέα χαρακτηριστικά που παρουσιάζουν οι χρήστες του κινητού δικτύου. Στη παρόν κεφάλαιο θα μελετηθούν οι ανάγκες του ασύρματου δικτύου και πώς αυτές μπορούν να ικανοποιηθούν εάν ληφθούν υπόψη τα χαρακτηριστικά που αποδίδει στους κινητούς χρήστες.

4.1 Οι ανάγκες του ασύρματου Δικτύου

Με την ανάπτυξη του ασύρματου δικτύου, οι ανάγκες για παροχή εξατομικευμένων υπηρεσιών αλλάζουν. Πιο συγκεκριμένα ο χρήστης γίνεται πιο απαιτητικός στις υπηρεσίες που αναζητά. Αυτό γιατί το ασύρματο δίκτυο του δίνει πλέον την δυνατότητα να έχει πρόσβαση σε πληροφορίες και υπηρεσίες, οποιαδήποτε στιγμή, σε οποιοδήποτε χώρο και ότι κι αν κάνει.

Έτσι είναι πλέον αναγκαιότητα η δημιουργία υπηρεσιών και συστημάτων οι οποίες έχουν επίγνωση και λαμβάνουν υπόψη τόσο το χώρο όσο και τον χρόνο στην οποίο βρίσκεται ο χρήστης κατά την αναζήτηση. Οι χρήστες συστημάτων και συγκεκριμένα εφαρμογών διαδικτύου δεν περιορίζονται πλέον στο παραδοσιακό "επιτραπέζιο" περιβάλλον (desktop environment), αλλά έχουν μετακινηθεί στο χώρο όπου οι άνθρωποι πραγματοποιούν τις εργασίες τους. Οι υπηρεσίες αυτές ονομάζονται situation aware [27] και time aware και επεκτείνονται έτσι ώστε να λαμβάνουν υπόψη τους το άμεσο περιβάλλον του χρήστη.

Όπως αναφέρθηκε πιο πάνω, η εξατομίκευση των πληροφοριών που παρέχονται βασίζεται στο περιεχόμενο. Η έννοια περιεχόμενο, αναφέρεται στην πληροφορία που μπορεί να χρησιμοποιηθεί για να χαρακτηρίσει την κατάσταση μιας οντότητας [27]. Η κατάσταση αυτή αναφέρεται τόσο στο φυσικό περιβάλλον του χρήστη, όσο και στο κοινωνικό περιβάλλον του. Σχετίζεται με στοιχεία αλληλεπίδρασής του με το σύστημα, με το ιστορικό προηγούμενων αλληλεπιδράσεων του, προσωπικές πληροφορίες και προτιμήσεις, καθώς και με τα ενδιαφέροντά του. Στην περίπτωση όμως των κινητών χρηστών, η κατάσταση αυτή επηρεάζεται από κάποια δυναμικά χαρακτηριστικά, του χρήστη όπως ο χρόνος, η τοποθεσία του χρήστη και η κατάστασή του σε μια δεδομένη στιγμή.

Το γεγονός ότι ο χρήστης μας κινείται συνεπάγεται αυτόματα την αλλαγή τοποθεσίας και του χώρου στον οποίο βρίσκεται. Είναι άρα αναμενόμενο ο χρήστης όταν ζητά μια υπηρεσία να ψάχνει για κάτι που βρίσκεται τοπικά κοντά του. Η παροχή εξατομικευμένων υπηρεσιών που βρίσκονται στο άμεσο περιβάλλον του χρήστη είναι μια από τις προκλήσεις του ασύρματου δικτύου. Έτσι είναι εφικτή η εισήγηση στο χρήστη υπηρεσιών που φιλτράρονται με βάση την τοποθεσία του, και προσαρμόζονται καλύτερα στις ανάγκες του. Με το τρόπο αυτό οι χρήστες μπορούν να εκμεταλλεύονται πλήρως τοπικές υπηρεσίες που πιθανόν να χρειάζονται.

Ο χρόνος είναι ένας ακόμη παράγοντας που επηρεάζει τις ανάγκες ενός κινητού χρήστη, αφού καθώς κινείται, με την πάροδο του χρόνου, διαφοροποιούνται οι ανάγκες του. Για παράδειγμα, εάν είναι μεσημέρι, το πιθανότερο είναι να ψάχνει ένα εστιατόριο ή ένα φαστφουντάδικο· αντιθέτως, το απόγευμα μπορεί να αναζητά ένα café. Είναι σημαντικό, άρα, οι εξατομικευμένες πληροφορίες που παρέχονται στο χρήστη να λαμβάνουν υπόψη το χρόνο της αναζήτησης και με τον τρόπο αυτό να προσαρμόζουν τις υπηρεσίες, που πιθανότατα να έχει περισσότερο ανάγκη ο χρήστης τη συγκεκριμένη στιγμή της αναζήτησης.

Ένας ακόμη άξονας που επηρεάζει εξατομικευμένες υπηρεσίες, που αναφέρονται σε κινητούς χρήστες, είναι η κατάστασή τους τη δεδομένη στιγμή της αναζήτησης. Εάν ένας χρήστης βρίσκεται σε διακοπές ή εάν βρίσκεται στο γραφείο, επηρεάζονται και αλλάζουν αντίστοιχα οι ανάγκες του. Έτσι, ένα σύστημα εξατομικευμένων υπηρεσιών είναι σημαντικό να εντοπίζει τις καταστάσεις αυτές του χρήστη και μέσα από την εμπειρία του να προσαρμόζει τα αποτελέσματά του στην παρούσα κατάσταση του χρήστη.

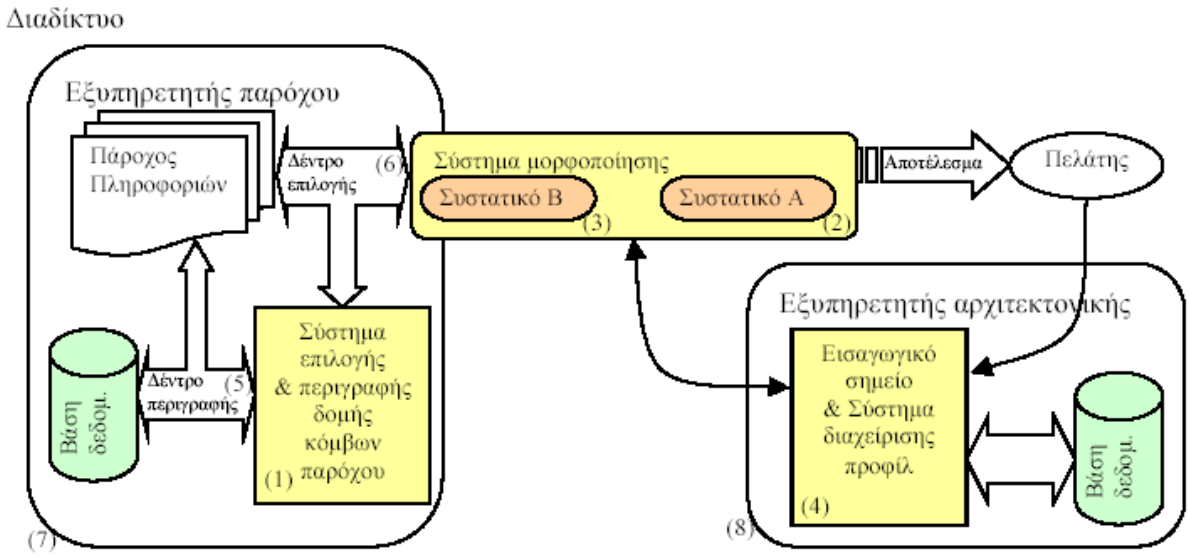
Λαμβάνοντας υπόψη όλα τα πιο πάνω, είναι φανερή η αναγκαιότητα να ευρύνουμε την έννοια της εξατομίκευσης, ώστε να μην περιορίζεται πλέον μόνο στη μελέτη του περιεχόμενου που πιθανώς ενδιαφέρει το χρήστη. Είναι ανάγκη πλέον να προχωρήσουμε ένα βήμα πιο πέρα και να συσχετίσουμε το περιεχόμενο αυτό με την τρέχουσα κατάσταση του χρήστη.

4.2 Αρχιτεκτονική συστημάτων εξατομίκευσης

Η αρχιτεκτονική συστημάτων εξατομίκευσης είναι μια αρχιτεκτονική βασισμένη σε κινητούς πράκτορες, η οποία μπορεί να χρησιμοποιηθεί για την ανάπτυξη συστημάτων που παρέχουν εξατομικευμένες υπηρεσίες στο διαδίκτυο, και συγκεκριμένα στο ασύρματο διαδίκτυο (WAP). Η αρχιτεκτονική αυτή, ακολουθεί μια νέα προσέγγιση στο πρόβλημα της εξατομίκευσης: τη δημιουργία συστημάτων εξατομίκευσης με χρήση μόνο κινητών πρακτόρων [3, 4].

Ένας επιπρόσθετος ρόλος της αρχιτεκτονικής αυτής είναι η πρόθεση της να παρέχει ένα είδος διαμεσολαβητή μεταξύ του παροχέα των πληροφοριών και του πελάτη, ο οποίος ζητά κάποιο πληροφοριακό περιεχόμενο. Έτσι, από τη μια υπάρχουν οι εξυπηρετητές με τους οποίους εισάγεται ένας χρήστης στο σύστημα (Σχήμα 4.1:8), και από την άλλη υπάρχουν οι εξυπηρετητές που ενώνουν τους

παροχείς περιεχομένου με την αρχιτεκτονική. (Σχήμα 4.1:7). Μεταξύ των δύο αυτών εξυπηρετητών υπάρχουν κινητοί πράκτορες, οι οποίοι επιλέγουν, ετοιμάζουν και παραδίδουν το επιθυμητό περιεχόμενο (Σχήμα 4.1:2&3).



Σχήμα 4.1: Γενική όψη της αρχιτεκτονικής mPersona και των συστατικών της

Με βάση την πιο πάνω αρχιτεκτονική (Σχήμα 4.1), χρειαζόμαστε ένα μηχανισμό για περιγραφή του περιεχομένου, καθώς και ένα μηχανισμό για τη διαχείριση του προφίλ των χρηστών. Έχοντας τους δύο αυτούς μηχανισμούς, και με τη βοήθεια ενός τρίτου, ο οποίος επιλέγει το περιεχόμενο με βάση το προφίλ του χρήστη και το μορφοποιεί κατάλληλα, παρουσιάζεται στο χρήστη το επιθυμητό αποτέλεσμα.

Πιο συγκεκριμένα, η αρχιτεκτονική συστημάτων εξατομίκευσης αποτελείται από τα πιο κάτω υπομέρη:

- Το σύστημα περιγραφής της δομής περιεχομένου (Σχήμα 4.1:1&5)
- Το σύστημα επιλογής περιεχομένου (Σχήμα 4.1:1&6)
- Το σύστημα μορφοποίησης του περιεχομένου (Σχήμα 4.1:2&3)
- Το σύστημα διαχείρισης των προφίλ των χρηστών (Σχήμα 4.1:4)

4.2.1 Το σύστημα περιγραφής της δομής περιεχομένου

Ο πιο συνηθισμένος τρόπος αναπαράστασης του περιεχομένου ενός παροχέα πληροφοριών στο ασύρματο περιβάλλον είναι με ιεράρχηση του τύπου της πληροφορίας. Μια ιεράρχηση που στην αρχιτεκτονική αυτή αναπαρίσταται με δεντρική δομή. Η δεντρική αυτή δομή εξυπηρετεί ακόμη ένα σκοπό: να ιεραρχήσει τη σημασιολογία του περιεχομένου από το γενικό στο αναλυτικό. Για το λόγο αυτό,

είναι απαραίτητη η περιγραφή του κάθε κόμβου, τόσο σε σημασιολογικό επίπεδο όσο και σε ένα επίπεδο πλοήγησης. Έτσι, μπορεί να περιγραφεί με ακρίβεια η δομή της πλοήγησης, καθώς και η δομή του περιεχομένου μιας σελίδας του παροχέα πληροφοριών.

Το τελικό αποτέλεσμα αυτού του υποσυστήματος είναι ένα δέντρο με μετα-δεδομένα. Σε κάθε κόμβο του δέντρου αυτού, αποθηκεύουμε μετα-πληροφορίες για κάθε αντίστοιχο πληροφοριακό κόμβο, ενώ η δομή του, μάς δίνει τις επιλογές πλοήγησης που έχουμε ανά πάσα στιγμή.

Εδώ είναι σημαντικό να σημειώσουμε ότι, έχοντας αυτό το δέντρο με τα μετα-δεδομένα, μπορεί να εφαρμοστεί σ' αυτό μια οποιαδήποτε τεχνική, έτσι ώστε να επιλεχθούν οι επιθυμητοί κόμβοι. Ένας από τους στόχους της παρούσας εργασίας είναι ο καθορισμός μιας τέτοιας τεχνικής, την οποία θα αναλύσουμε σε μεταγενέστερο στάδιο.

4.2.2 Το σύστημα επιλογής περιεχομένου

Το σύστημα επιλογής περιεχομένου χρησιμοποιεί το δέντρο περιγραφής της δομής ενός κόμβου και, σε συνδυασμό με το προφίλ του χρήστη, δημιουργεί ένα δένδρο στο οποίο υπάρχουν μόνο οι κόμβοι οι οποίοι ενδιαφέρουν το χρήστη. Για το λόγο αυτό γίνεται πρώτα εύρεση των ενδιαφερόντων κόμβων. Στο σημείο αυτό, συγκρίνουμε τους κόμβους από το δέντρο περιγραφής δομής με το προφίλ και κρατούμε τους κόμβους που παρουσιάζουν κάποιο ενδιαφέρον για το χρήστη. Στη συνέχεια, απορρίπτονται από το δέντρο πλοήγησης οι κόμβοι, οι οποίοι δεν βρίσκονται σε κάποιο από τα μονοπάτια πλοήγησης προς τους επιλεγμένους κόμβους. Στο τελευταίο βήμα, μειώνουμε ακόμη περισσότερο το δέντρο εξαλείφοντας αχρείαστους εσωτερικούς κόμβους.

Το σύστημα αυτό μας δίνει τελικά ένα καινούριο ελαττωμένο δέντρο, το οποίο περιλαμβάνει μόνο τους επιθυμητούς κόμβους. Το δέντρο αυτό όμως δεν περιλαμβάνει μετα-πληροφορίες του αρχικού δέντρου. Μας δίνει απλά τις πληροφορίες πλοήγησης.

4.2.3 Το σύστημα μορφοποίησης του περιεχομένου

Το σύστημα αυτό αποτελείται από δύο κομμάτια. Το πρώτο κομμάτι παίρνει σαν είσοδο του ένα δέντρο πλοήγησης και μεταφέρει στο επόμενο κομμάτι τον

ζητούμενο κόμβο από τον παροχέα περιεχομένου. Ο κόμβος αυτό μεταφέρεται τροποποιημένος, αφού το υποσύστημα αυτό είναι υπεύθυνο να αλλάξει τους συνδέσμους που περιέχει, σύμφωνα με το δέντρο πλοήγησης.

Στη συνέχεια, το δεύτερο μέρος του συστήματος μορφοποίησης παίρνει τους ενημερωμένους κόμβους περιεχομένου από το πρώτο και τους μορφοποιεί σύμφωνα με το προφίλ της τερματικής συσκευής του χρήστη. Για να γίνει όμως αυτό, θα πρέπει οι κόμβοι περιεχομένων να είναι γραμμένοι σε μια γλώσσα που να είναι κατανοητή από τα μέρη του συστήματος. Μια τέτοια γλώσσα θα ήταν πάρα πολύ εύκολο να οριστεί με χρήση της XML.

4.2.4 Το σύστημα διαχείρισης των προφίλ των χρηστών

Το προφίλ των χρηστών είναι, όπως είδαμε, απαραίτητο να περιέχει τόσο τα θεματικά ενδιαφέροντα του χρήστη, όσο και τα χαρακτηριστικά της συσκευής που χρησιμοποιεί. Η αποθήκευση των προφίλ αυτών μπορεί να γίνει είτε με την συγγραφή αρχείων, είτε με τη χρήση μιας βάσης δεδομένων. Είναι σημαντικό, επίσης, τα δύο αυτά προφίλ να είναι ανεξάρτητα μεταξύ τους, έτσι ώστε να καθίσταται εύκολη η προσαρμογή της δομής τους σε περίπτωση αλλαγών. [3, 4]

4.3 Σύστημα Εξατομίκευσης για κινητούς χρήστες

Στο [28] προτείναμε ένα σύστημα εξατομίκευσης, το οποίο λαμβάνει υπόψη τα χαρακτηριστικά του κινητού χρήστης, και αλλάζει δυναμικά το προφίλ του χρήστη με βάση τα χαρακτηριστικά αυτά. Έτσι τα αποτελέσματα που παρουσιάζονται στο χρήστη είναι ανάλογα της χρονικής στιγμής, της τοποθεσίας και της κατάστασης του χρήστη την στιγμή της αναζήτησης.

Το πιο πάνω σύστημα προτείνει μια μέθοδο εξατομίκευσης με χρήση προφίλ χρηστών στα οποία διατηρούμε ποσοστά προτίμησης σε χαρακτηριστικά που ενδιαφέρουν το χρήστη. Η περιγραφή του προφίλ και των παροχέων περιεχομένου ακολουθούν μια κοινή οντολογία. Αυτό κάνει πολύ πιο εύκολη την αντιστοιχία του προφίλ με τις υπηρεσίες ώστε να μπορέσουμε να εξάγουμε τα ενδιαφέροντα του χρήστη. Για περισσότερες πληροφορίες στις προαναφερθείς οντολογίες βλέπε Παράρτημα Α.

Το σημαντικό στην υλοποίηση αυτή, είναι η δυναμική αλλαγή του προφίλ ώστε να προσαρμόζεται στις ιδιαίτερες ανάγκες του κινητού χρήστη που εμφανίζονται στο

ασύρματο δίκτυο. Ενδιαφέρον επίσης είναι και το πως αναπαριστούμε στο προφίλ τους παράγοντες του χρόνου, και της δραστηριότητας του χρήστη, ώστε να συμπεριλάβουμε σε αυτό πληροφορίες που μας δίνουν στοιχεία για την αλλαγή της συμπεριφοράς του χρήστη σε διαφορετικές καταστάσεις.

4.3.1 Αναπαράσταση προφίλ

Στο προφίλ του χρήστη κρατούμε τα χαρακτηριστικά που περιγράφουν κάθε υπηρεσία που φαίνεται να τον ενδιαφέρει σε ζευγάρια <χαρακτηριστικό, ποσοστό προτίμησης>. Κάθε υπηρεσία χαρακτηρίζεται από κατηγορίες, υποκατηγορίες και χαρακτηριστικά στις υποκατηγορίες αυτές. Σε κάθε βαθμίδα κρατούμε και το αντίστοιχο ποσοστό προτίμησης που αναλογεί. Το ποσοστό προτίμησης, προσαρμόζεται και ενημερώνεται με βάση τις επιλογές του χρήστη. Εάν ο χρήστης επιλέξει μια υπηρεσία που περιλαμβάνει μια συγκεκριμένη κατηγορία-υποκατηγορία-χαρακτηριστικό, όλα αυτά θα ενημερωθούν και το ποσοστό προτίμησής τους θα αυξηθεί. Στην αντίθετη περίπτωση το ποσοστό προτίμησης μειώνεται. Το ποσοστό προτίμησης ενός χαρακτηριστικού λαμβάνεται υπόψη στην αξιολόγηση μιας υπηρεσία εάν αντιπροσωπεύει το χρήστη, εάν θα πρέπει να παρουσιαστεί σε αυτόν ή όχι. Ωστόσο το τελικό ποσοστό ενός χαρακτηριστικού αλλάζει με βάση το ποσοστό προτίμησης που έχει η κατηγορία και υποκατηγορία στην οποία ανήκει.

Το σημείο στο οποίο αξίζει να επικεντρωθούμε είναι ο τρόπος που αναπαρίσταται ο χρόνος και οι δραστηριότητες (user experiences) του χρήστη με βάση τα οποία το προφίλ διαμορφώνεται. Για να πετύχουμε εξατομίκευση με βάση το χρόνο, είναι σημαντικό να μπορούμε να διαφοροποιήσουμε τις προτιμήσεις του χρήστη στο προφίλ του και πιο συγκεκριμένα πως αυτές αλλάζουν μέσα σε ένα 24ωρο. Για να μπορέσουμε να αναπαραστήσουμε το χρόνο, χωρίζουμε τη μέρα σε χρονικές περιόδους με βάση τις δραστηριότητες του χρήστη μέσα στο 24ωρο. Το πιο πάνω είναι εφικτό, εάν μελετήσουμε τη καθημερινότητα του χρήστη και μετά τη χωρίσουμε σε χρονικές περιόδους με βάση τις δραστηριότητες του. Με το διαχωρισμό σε χρονικές περιόδους μειώνουμε τους πιθανούς συνδυασμούς μεταξύ του χρόνου και των προτιμήσεων του χρήστη. Έχοντας από τη μια τις χρονικές περιόδους και από την άλλη τις προτιμήσεις του χρήστη για να εξάγουμε την πληροφορία που χρειάζεται ο χρήστη σε μια δεδομένη στιγμή, αρκεί να προσαρμόσομε το πως αλλάζει το ποσοστό προτίμησης στα χαρακτηριστικά των υπηρεσιών σε κάθε χρονική περίοδο. Το σχήμα 4.1 δείχνει πως αναπαριστούμε την πληροφορία αυτή στο προφίλ του χρήστη.

```
<restaurantsTypeByTime categoryWeight="50">
    <timeZone time="0-3">
        <restaurantType name="Fast Food" weight="90"/>
    </timeZone>
    <timeZone time="3-6"></timeZone>
    <timeZone time="6-9"></timeZone>
    <timeZone time="9-12">
        <restaurantType name="Fast Food" weight="90"/>
        <restaurantType name="Greek" weight="70"/>
        <restaurantType name="Cypriot" weight="60"/>
        <restaurantType name="Kebab House" weight="95"/>
    </timeZone>
    <timeZone time="12-15">
        <restaurantType name="Pizzarias" weight="85"/>
    </timeZone>
    <timeZone time="15-18"></timeZone>
    <timeZone time="18-21">
        <restaurantType name="Chinese" weight="90"/>
        <restaurantType name="Italian" weight="80"/>
        <restaurantType name="Mexican" weight="75"/>
    </timeZone>
    <timeZone time="21-24"></timeZone>
</restaurantsTypeByTime>
```

Πέραν από το την αναπαράσταση των χρονικών περιόδων, η αναπαράσταση των αντίστοιχων ποσοστών προτίμησης για τις διαφορετικές δραστηριότητες (experiences) του χρήστη, είναι επίσης σημαντική. Χρησιμοποιώντας το σύστημα των ποσοστών προτίμησης, για κάθε χαρακτηριστικό, όπως ακριβώς και για κάθε χρονική περίοδο, μπορούμε να επαναλάβουμε τα ποσοστά προτίμησης για τις διαφορετικές δραστηριότητες του χρήστη. Με το τρόπο αυτό απλά προσθέτοντας ακόμη ένα σύνολο ποσοστών προτίμησης μπορούμε να έχουμε τη συμπεριφορά του χρήστη για μια νέα δραστηριότητά του. Έτσι αλλάζουμε όχι μόνο την σύνθεση των χρονικών περιόδων αλλά και τις ίδιες τις προτιμήσεις του, που πιθανότατα να αλλάζουν όταν ο χρήστης βρίσκεται σε διαφορετική δραστηριότητα.

Επιπλέον η περιγραφή των χρονικών δραστηριοτήτων του χρήστη γίνεται σε δύο επίπεδα. Για κάθε υπηρεσία που παρέχεται στον χρήστη ορίζονται και οι χρονικές περίοδοι που ο χρήστης προτιμά αυτή την υπηρεσία. Επιπλέον για κάθε τύπο της υπηρεσίας αυτής, οι χρονικές αυτές περίοδοι επαναπροσδιορίζονται. Έτσι μπορούμε να διακρίνουμε πώς αλλάζουν οι προτιμήσεις του χρήστη κατά την πάροδο του χρόνου στην ίδια υπηρεσία. Για παράδειγμα πιθανότητα να έχουμε ένα χρήστη που στις εργάσιμες ώρες να προτιμά περισσότερο τα φαστφουντάδικα ενώ τα αγαπημένα του εστιατόρια να είναι τα κινέζικα τα οποία προτιμά όμως πιο αργά το βράδυ. Η ίδια ακριβώς λογική επαναλαμβάνεται και για τις διαφορετικές δραστηριότητες που αναφέραμε πιο πάνω.

4.3.2 Η διαχείριση του προφίλ

Η διαχείριση του προφίλ θέτει δύο βασικά ερωτήματα. Πώς χρησιμοποιούμε το προφίλ για να μπορέσουμε να εξάγουμε τις προτιμήσεις του χρήστη και πώς ο ενημερώνουμε ώστε να αντιπροσωπεύει όσο πιο αξιόπιστα τα ενδιαφέροντα και τις συμπεριφορές του χρήστη.

Για να μπορέσουμε να εξάγουμε τις προτιμήσεις του χρήστη από το προφίλ του, αντιστοιχούμε το προφίλ με τις υπηρεσίες που προσφέρονται. Όπως έχουμε ήδη αναφέρει το προφίλ και οι υπηρεσίες περιγράφονται με την ίδια οντολογία, κάτι που κάνει την όλη διαδικασία πολύ πιο απλή. Συνοπτικά η διαδικασία αποτελείται από τα πιο κάτω βήματα:

- Για κάθε χαρακτηριστικό το οποίο υπάρχει στο προφίλ και εμφανίζεται στη υπηρεσία που εξετάζουμε, βρίσκουμε το ποσοστό προτίμησης συνδυάζοντας τα χαρακτηριστικά προτίμησης της αντίστοιχης κατηγορίας, υποκατηγορίας και του ίδιου του χαρακτηριστικού

- Αφού βρούμε τα ποσοστά προτίμησης για όλα τα κοινά χαρακτηριστικά, το τελικό ποσοστό προτίμησης που αντιπροσωπεύει την συγκεκριμένη υπηρεσία δίνεται από το μέσο όρο των χαρακτηριστικών αυτών.

Το σύστημα αφού βρει τα αντίστοιχα ποσοστά προτιμήσεις για τις υπηρεσίες που του ζητήθηκε, ταξινομεί τα αποτελέσματα εμφανίζοντας στο χρήστη αυτό με το πιο ψηλό ποσοστό.

Το πώς ενημερώνουμε το προφίλ είναι καίριας σημασίας και επηρεάζει άμεσα τα αποτελέσματα της εξατομίκευσης. Στην υλοποίησης αυτή ο αλγόριθμος ενημέρωσης του προφίλ χρησιμοποιούσε ποσοστά προτίμησης από -1 έως 100, και το -1 επέτρεπε στο σύστημα να αγνοήσει πλήρως κάποιο χαρακτηριστικό. Η ενημέρωση του προφίλ στηρίζεται στις επιλογές του χρήστη. Κάθε φορά που ο χρήστης επιλέγει να δει πληροφορίες για μια υπηρεσία το σύστημα ενημερώνει το προφίλ για την κίνηση αυτή. Η ενημέρωση αυτή δεν περιορίζεται μόνο στο περιεχόμενο της υπηρεσίας που επιλέχθηκε αλλά από την επιλογή του χρήστη παίρνουμε πληροφορίες και για τα χαρακτηριστικά του χρήστη την ώρα της αναζήτησης. Δηλαδή τον χρόνο, την κατάστασή και την τοποθεσία του. Το περιεχόμενο που ενημερώνεται , ενημερώνεται σε σχέση με αυτούς τους τρεις άξονες. Το προφίλ ενημερώνεται με τα πιο κάτω βήματα:

- Για όλα τα χαρακτηριστικά που είναι παρών στο προφίλ και στην υπηρεσία που επέλεξε ο χρήστης, το βάρος προτίμησης αυξάνεται κατά μία μονάδα

- Για όλα τα χαρακτηριστικά που υπάρχουν στο προφίλ αλλά απουσιάζουν από την υπηρεσία που επέλεξε ο χρήστης, το βάρος προτίμησης μειώνεται κατά ένα

- Όλα τα χαρακτηριστικά που εμφανίζονται στην υπηρεσία που επέλεξε ο χρήστης αλλά απουσιάζουν από το προφίλ αυξάνονται κατά ένα.

4.4 Σκοπός της παρούσας εργασίας

Στη παρούσα εργασία μελετούμε διαφορετικούς αλγορίθμους ενημέρωσης του προφίλ και συγκεκριμένα αλγορίθμους που χρησιμοποιούν το σύνολο των επιλογών του χρήστη (clickstream).

Το πώς ενημερώνεται το προφίλ είναι ύψιστης σημασίας γιατί τα ποσοστά προτίμησης στο προφίλ είναι αυτό που θα μας δώσει τα ενδιαφέροντα του χρήστη. Ωστόσο στην εξατομίκευση για κινητούς χρήστες υπάρχει μια σημαντική πρόκληση. Πώς ενημερώνω ουσιαστικά το προφίλ ώστε να μπορέσω να συσχετίσω το περιεχόμενο με την κατάσταση του χρήστη, ώρα, τόπο και δραστηριότητα. Πώς εντοπίζονται τα χαρακτηριστικά αυτά και πως καταγράφονται στο προφίλ ώστε να μπορούμε να έχουμε μια ολοκληρωμένη αναπαράσταση των ενδιαφερόντων του χρήστη και του πώς αυτά αλλάζουν ανάλογα με την κατάσταση στην οποία βρίσκεται.

Ωστόσο, στους αλγορίθμους αυτούς υπάρχει ένα κενό το οποίο θα πρέπει να διερευνηθεί πως μπορούμε να το καλύψουμε. Οι αλγόριθμοι αυτοί που ενημερώνουν το προφίλ με βάση το clickstream τρέχουν περιοδικά, σε συχνά χρονικά διαστήματα. Πώς χειριζόμαστε όμως και πώς παρουσιάζουμε πληροφορίες στον χρήστη μέχρι να εκτελεστούν για πρώτη φορά, ώστε να μπορέσουμε να έχουμε κάποια δεδομένα για τον χρήστη; Αυτό είναι ένα θεμελιώδες ερώτημα για τους αλγορίθμους clickstream αφού η απόδοση του προφίλ όταν ο χρήστης πρωτοέρθει σε επαφή με το σύστημα είναι αυτό που θα τον πείσει ή όχι να χρησιμοποιήσει το σύστημα. Στην παρούσα εργασία μελετούμε τρόπους για την αποδοτική αντιμετώπιση του προβλήματος αυτού, έτσι ώστε το προφίλ να παραμένει αποδοτικό ακόμη και στις πρώτες χρήσεις του συστήματος.

Τέλος, η όλη εργασία ελέγχεται με διαφορετικά σενάρια, όπου μπορούμε να διακρίνουμε τη συμπεριφορά του χρήστη και να παρατηρήσουμε την αποδοτικότητα των αλγορίθμων κάτω από αυτά τα σενάρια.

Κεφάλαιο 5

Αλγόριθμοι Ενημέρωσης Προφίλ Χρηστών.

5.1 Οι αλγόριθμοι Clickstream

Οι αλγόριθμοι clickstream χρησιμοποιούνται συνήθως για collaborative filtering και υιοθετούν μοντέλα εξατομίκευσης για αποδοτικές και αποτελεσματικές εισηγήσεις υπηρεσιών στο χρήστη. Με τους αλγορίθμους αυτούς, η επεξεργασία των δεδομένων καθώς και η δημιουργία των μοντέλων χρηστών γίνονται offline. Συνήθως οι αλγόριθμοι αυτοί χρησιμοποιούνται για:

- Προσαρμογή της διαπροσωπίας ιστοσελίδων, με το να προβλέπουν συναφή σελίδες, κείμενα, κατηγορίες ή προϊόντα
- Αντιμετώπιση του προβλήματος της υπερφόρτωσης με πληροφορίες, παρέχοντας στο χρήστη διασυνδέσμους που σχετίζονται με το αντικείμενο του ενδιαφέροντος του.
- Ελαχιστοποίηση του χρόνου καθυστέρησης κατά την εμφάνιση σελίδων με prefetching συσχετιζόμενων σελίδων.

Συνήθως οι αλγόριθμοι αυτοί, χρησιμοποιούν κανόνες συσχέτισης και συσταδοποίηση για να συσχετίσουν αντικείμενα οι χρήστες μεταξύ τους και να μπορέσουν να προσφέρουν εξατομικευμένες υπηρεσίες. [29]

Στην παρούσα εργασία κάνουμε μια διαφορετική χρήση των αλγορίθμων αυτών. Εδώ οι αλγόριθμοι αυτοί, επεξεργάζονται σε τακτά χρονικά διαστήματα το clickstream ενός συγκεκριμένου χρήστη, με σκοπό να μελετήσουν την συμπεριφορά του. Σκοπός, των αλγορίθμων αυτών δεν είναι απλά να εξάγουν τα ενδιαφέροντα ενός χρήστη αλλά και να εντοπίσουν τη συμπεριφορά του χρήστη κατά τη διάρκεια της μέρας. Πιο συγκεκριμένα, οι αλγόριθμοι αυτοί μελετούν τις απαιτήσεις του χρήστη λαμβάνοντας υπόψη το χρόνο της αναζήτησης, με αποτέλεσμα να δημιουργούν ένα προφίλ το όποιο δεν περιορίζεται στο να δώσει τα «likes and dislikes» του χρήστη, αλλά προσπαθεί να δώσει ένα διάγραμμα των δραστηριοτήτων (activity diagram) του.

5.2 Διατήρηση του Clickstream

Για κάθε χρήστη, διατηρούμε ένα log File στο οποίο κρατούμε τις επιλογές του χρήστη. Κάθε φορά που αυτός επιλέγει μια υπηρεσία, η υπηρεσία αυτή καταγράφεται μαζί με την ώρα της αναζήτησης. Πιο συγκεκριμένα στο αρχείο αυτό καταγράφει τις πιο κάτω πληροφορίες:

```
<logRecord date="2008-03-16" experience="weekend"
service="restaurants" serviceDescription="C:\Profiling\restaurant-
restaurant17.xml" time="12:38:54"/>
```

Τα logs αυτά διαβάζονται από το σύστημα το οποίο τα επεξεργάζεται με διάφορους αλγορίθμους για να ενημερώσει το προφίλ του χρήστη για αλλαγές στη συμπεριφορά του χρήστη.

5.3 Εξαγωγή ενδιαφερόντων του χρήστη από το Clickstream

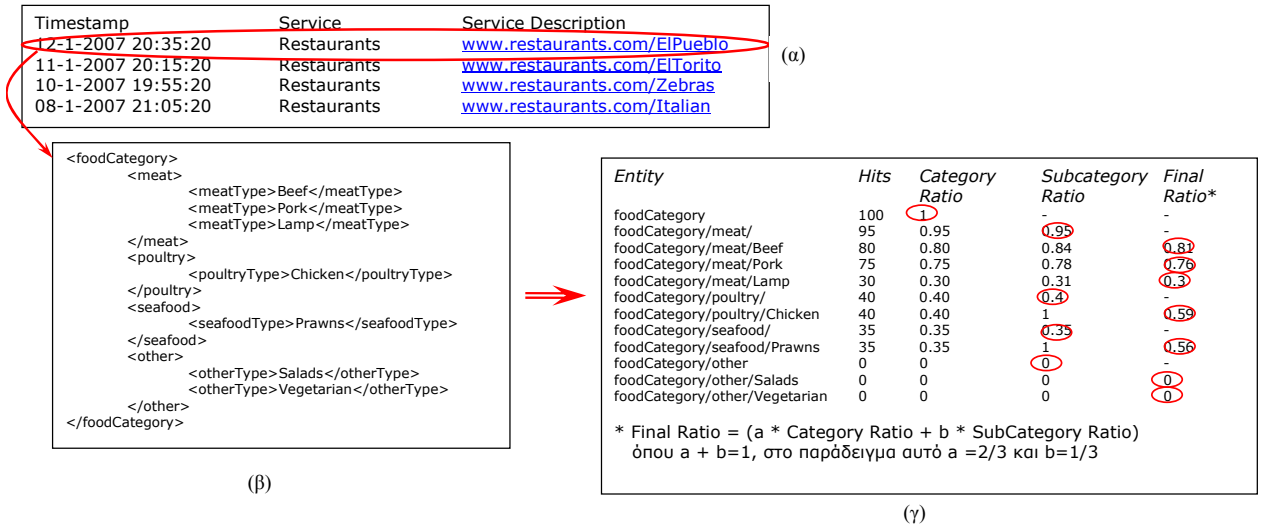
Για την εξαγωγή των ενδιαφερόντων του χρήστη, επεξεργαζόμαστε το clickstream που αντιστοιχεί στον χρήστη αυτόν. Κάθε εγγραφή στο Cliskstream περιλαμβάνει ένα σύνδεσμο στο αρχείο που περιγράφει ένα στιγμιότυπο μιας υπηρεσίας, το οποίο ζήτησε ο χρήστης στο παρελθόν. Το αρχείο αυτό περιγράφει, ακολουθώντας την κοινή οντολογία που ήδη αναφέραμε στο κεφάλαιο 4.3, την υπηρεσία και τα διάφορα χαρακτηριστικά της. Το σύστημα στην συνέχεια αναλύει κάθε τέτοιο αρχείο και εξάγει τις κατηγορίες και τα χαρακτηριστικά που την

περιγράφουν και τα οποία μπορούν να πάρουν κάποιο ποσοστό προτίμησης στο προφίλ του χρήστη. Για κάθε κατηγορία και χαρακτηριστικό διατηρείται ένας counter ο οποίος ενημερώνεται κάθε φορά που το ίδιο χαρακτηριστικό ανιχνεύεται σε μια άλλη υπηρεσία που επέλεξε ο χρήστης.

Έτσι στο τέλος της διαδικασίας, παίρνουμε μια λίστα με όλα τα χαρακτηριστικά που περιγράφουν τις υπηρεσίες που ζήτησε ο χρήστης, καθώς και τον αριθμό που μας δίνει πόσες φορές ο χρήστης αναζήτησε υπηρεσία με κάποιο συγκεκριμένο χαρακτηριστικό. Το αποτέλεσμα της πιο πάνω διαδικασίας δίνεται στο σχήμα 5.3.γ.

Η διαδικασία αυτή είναι πάρα πολύ σημαντική, καθώς πλεονεκτεί στο γεγονός ότι με το διαχωρισμό αυτό, δεν συγκρίνουμε πλέον τα χαρακτηριστικά μιας συγκεκριμένης υπηρεσίας με το προφίλ, όπως γινόταν μέχρι τώρα. Στη μέθοδο αυτή συγκρίνουμε όλα τα χαρακτηριστικά που ζήτησε ο χρήστης με το προφίλ. Με το τρόπο, αυτό μπορούμε να διαχωρίσουμε τα χαρακτηριστικά που ενδιαφέρουν τον χρήστη και όχι τα στιγμιότυπα τηςς υπηρεσίες. Έτσι, ανάλογα του βαθμού που ενδιαφέρουν το χρήστη, μπορεί να διαφοροποιηθεί κι ο τρόπος ενημέρωσης τους. (Βλέπε Clustered Clickstream Update Algorithm).

Σχηματικά η πιο πάνω μέθοδος δίνετε πιο κάτω:



Σχήμα 5.3: Μέθοδος εύρεσης των κατηγοριών και χαρακτηριστικών μιας υπηρεσίας που ενδιαφέρουν το χρήστη

Αφού έχουμε πάρει τη λίστα με τις κατηγορίες και τα χαρακτηριστικά που ενδιαφέρουν το χρήστη, και παράλληλα γνωρίζουμε το πόσες φορές εμφανίστηκαν σε υπηρεσίες που επέλεξε ο χρήστης, υπολογίζουμε έναν όρο που μας δείχνει πόσο σημαντικό είναι ένα χαρακτηριστικό. Αυτή η σημαντικότητα του κάθε χαρακτηριστικού εξαρτάται από την κατηγορία και την υπό-κατηγορία στην οποία ανήκει και την αντίστοιχη σημαντικότητά τους αλλά και από το σύνολο των υπόλοιπων χαρακτηριστικών. Για τον υπολογισμό της σημαντικότητας ενός χαρακτηριστικού υπολογίζουμε τη σημαντικότητα της κατηγορίας και της υποκατηγορίας στην οποία ανήκει. Η σημαντικότητα της κατηγορίας στην οποία ανήκει είναι ο αριθμός των click στο χαρακτηριστικό αυτό δια το σύνολο των clicks στην κατηγορία. Η σημαντικότητα της υπο-κατηγορίας στην οποία ανήκει ένα χαρακτηριστικό είναι ο λόγος των clicks που έγιναν στο χαρακτηριστικό δια του αριθμού των clicks που έγιναν στην υποκατηγορία αυτή. Η δύο αυτοί όροι κανονικοποιούνται πολλαπλασιαζόμενοι με δύο ανεξάρτητους όρους α και β, όπου $\alpha+\beta=1$ και $\alpha \geq \beta$. Ο όρος α επηρεάζει τη σημαντικότητα της κατηγορίας και ο β την σημαντικότητα της υποκατηγορίας. Με την συνθήκη $\alpha \geq \beta$ δίνεται μεγαλύτερη ή ίση βαρύτητα στη σημαντικότητα που κληρονομεί ένα χαρακτηριστικό λόγω της σημαντικότητας της κατηγορίας του. Αυτό γιατί η σημαντικότητα μιας κατηγορίας πρέπει να είναι τουλάχιστον ίση με την σημαντικότητα της υποκατηγορίας γιατί αυτή περικλείει όλα τα clicks στις υποκατηγορίες και τα χαρακτηριστικά της.

Παράδειγμα:

Για να εξηγήσουμε καλύτερα τα πιο πάνω χρησιμοποιούμε τα δεδομένα του σχήματος 5.3.γ για να υπολογίσουμε το ποσοστό σημαντικότητας στο χαρακτηριστικό foodCategory/meat/Beef.

Το ποσοστό σημαντικότητας της κατηγορίας foodCategory είναι ίσο με τον αριθμό των clicks που έγιναν για το χαρακτηριστικό beef(80) δια τον αριθμό των clicks που έγιναν στην κατηγορία στην οποία ανήκει, δηλαδή της meatCategory (100). Ο λόγος των δυο αυτών αριθμών (80/100) μας δίνει την σημαντικότητα κατηγορίας για το χαρακτηριστικό beaf ίση με 0.8.

Το ποσοστό σημαντικότητας της υπο-κατηγορίας meet είναι ίσο με τον αριθμό των clicks που έγιναν για το χαρακτηριστικό beef(80) δια τον αριθμό των clicks που εγιναν στην υπο-κατηγορία στην οποία ανήκει, δηλαδή της meat (95). Ο λόγος των δύο αυτών αριθμών (80/95) μας δίνει την σημαντικότητα υπο-κατηγορίας για το χαρακτηριστικό beef ίση με 0.84.

Η τελική σημαντικότητα του χαρακτηριστικού beef δίνεται με την εξίσωση α*0.8+β*0.84. Στο παράδειγμά μας τα α και β διαλέγονται να είναι 2/3 και 1/3 αντίστοιχα άρα το τελικό αποτέλεσμα είναι ίσο με 0.81.

5.4 Αλγόριθμοι Clickstream για ενημέρωση των χαρακτηριστικών υπηρεσίας του προφίλ

Στο σύστημα αναπτύσσουμε τρεις διαφορετικούς αλγορίθμους για ενημέρωση του προφίλ, οι οποίοι χρησιμοποιούν το clickstream για να εξάγουν στοιχεία για την συμπεριφορά των χρηστών και να ενημερώσουν το προφίλ το χρηστών. Οι αλγόριθμοι αυτοί διαφέρουν στο κατά πόσο διαφοροποιούν τον αλγόριθμο ενημέρωσης του ποσοστού ανάλογα με το πόσο συχνά εμφανίζεται ένα χαρακτηριστικό στις επιλογές του χρήστη και στο κατά πόσο λαμβάνουν υπόψη το προηγούμενο ποσοστό προτίμησης.

5.4.1 Clustered Clickstream Update Algorithm

Ο αλγόριθμος αυτός, παίρνει το clickstream και ταξινομεί τις επιλογές του χρήστη σε τρεις ομάδες. Οι ομάδες αυτές διαχωρίζουν τις επιλογές του χρήστη ανάλογα με τη συχνότητά τους. Στη συνέχεια για κάθε ομάδα, χρησιμοποιείται διαφορετικός αλγόριθμος για ενημέρωση των χαρακτηριστικών που ανήκουν σε αυτή, στο προφίλ.

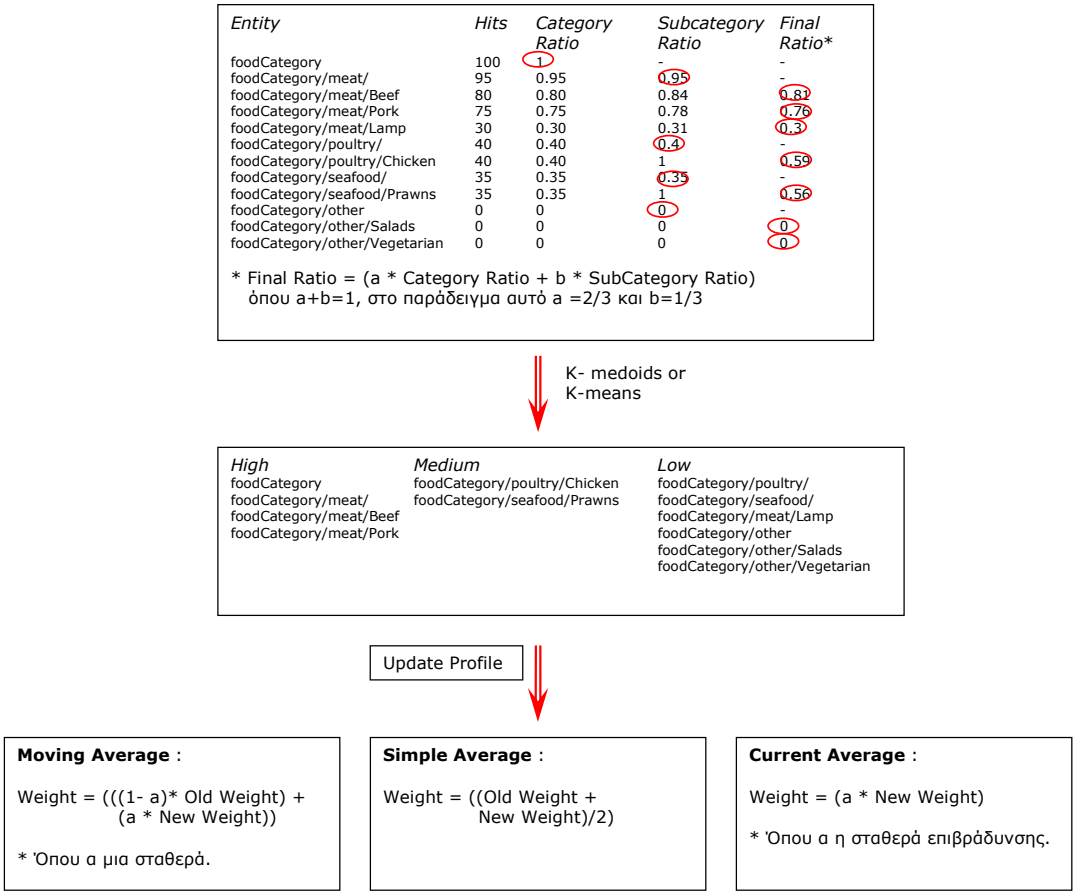
Ο αλγόριθμος αυτός πλεονεκτεί στο τρόπο που αντιμετωπίζει τα διάφορα χαρακτηριστικά των υπηρεσιών. Ακριβώς επειδή γίνεται πρώτα ο διαχωρισμός των χαρακτηριστικών από τις υπηρεσίες που χαρακτηρίζουν, μας δίνεται η δυνατότητα να επικεντρωθούμε σε αυτά καθ' αυτά τα χαρακτηριστικά και να τα αντιμετωπίσουμε με διαφορετικό τρόπο ανάλογα με την συχνότητα που εμφανίζονται στις επιλογές του χρήστη. Έτσι, μπορούμε να δώσουμε περισσότερη σημασία στα χαρακτηριστικά που ο χρήστης επιλέγει συχνότερα και να τους δώσουμε πιο γρήγορα το ποσοστό προτίμησης που τους αντιστοιχεί.

Λαμβάνοντας υπόψη ότι το κυριότερο στο προφίλ είναι να βρούμε τα χαρακτηριστικά που ενδιαφέρουν πολύ τον χρήστη, και όχι να εντοπίσουμε γενικά τα χαρακτηριστικά του ή να εντοπίσουμε αυτά που τον ενδιαφέρουν λίγο, για να ενημερώσουμε τα χαρακτηριστικά αυτά που ανήκουν στις υψηλές προτιμήσεις του χρήστη χρησιμοποιούμε ένα αλγόριθμο που εντοπίζει γρήγορα αλλαγές και προβλέπει την τάση των δεδομένων. Έτσι για τα χαρακτηριστικά που ανήκουν σε αυτή τη κατηγορία χρησιμοποιούμε Moving Average για την ενημέρωση του ποσοστού προτίμησης που αντιστοιχεί σε αυτά. Για τα χαρακτηριστικά που ανήκουν στη μεσαία κατηγορία προτίμησης χρησιμοποιούμε ένα απλό αλγόριθμο, ο οποίος βρίσκει το μέσο όρο ζήτησης ενός χαρακτηριστικού. Για τα χαρακτηριστικά που φαίνεται να ενδιαφέρουν ελάχιστα το χρήστη χρησιμοποιούμε ένα αλγόριθμο ο οποίος επιβραδύνει το ποσοστό προτίμησης τους.

Ο διαχωρισμός των χαρακτηριστικών σε υψηλού, μεσαίου και χαμηλού ενδιαφέροντος γίνεται με την χρήση του αλγορίθμου k-means, όπως αυτός περιγράφτηκε στο κεφάλαιο 2.7.2. Στη περίπτωση του δικού μας προβλήματος, το μόνο χαρακτηριστικό το οποίο έχει να διαχωρίσει ο αλγόριθμος αυτός είναι η σημαντικότητα του χαρακτηριστικού, έτσι όπως ορίστηκε στην προηγούμενη παράγραφο. Έτσι είσοδος στον αλγόριθμο k-means είναι μια λίστα με την σημαντικότητα των χαρακτηριστικών που μας ενδιαφέρει να μελετήσουμε. Το αποτέλεσμα που επιστρέφει ο αλγόριθμος είναι μια λίστα με διαχωρισμό στον οποίο ανήκει το κάθε χαρακτηριστικό (υψηλού, μεσαίου, χαμηλού ενδιαφέροντος).

Το σημαντικό στον αλγόριθμο αυτό είναι ότι κατατάσσει τα χαρακτηριστικά σε ομάδες προτιμήσεις και χρησιμοποιεί διαφορετικούς αλγορίθμους για να ενημερώσει τα ποσοστά προτιμήσεις για κάθε ομάδα. Εάν ενημερώνουμε με τον ίδιο τρόπο όλες τις κατηγορίες τότε δεν μπορούμε να ξεχωρίσουμε τα χαρακτηριστικά που ενδιαφέρουν τον χρήστη από κάποια υπηρεσία αφού τα ενημερώνουμε όλα με τον

ίδιο τρόπο. Χρησιμοποιώντας διαφορετικό αλγόριθμο για κάθε κατηγορία, μπορούμε να κάνουμε πιο εμφανές τα χαρακτηριστικά που πραγματικά ενδιαφέρουν τον χρήστη σε μια υπηρεσία και να του παρέχουμε υπηρεσίες που θα του δώσουν περισσότερες επιλογές σε αυτά τα χαρακτηριστικά και κατ' επέκταση μεγαλύτερη πιθανότητα να βρει αυτό που χρειάζεται. Η λογική αυτή στηρίζεται στην υπόθεση ότι ο χρήστης τείνει να ζητά υπηρεσίες που περιέχουν τα χαρακτηριστικά που τον ενδιαφέρουν.



5.4.1.1 Αλγόριθμοι Ενημέρωσης των ποσοστών προτίμησης ανα ομάδα προτίμησης

Moving Average

Ο υπολογισμός του moving average γίνεται με βάση τον πιο κάτω τύπο:

$$W(t) = (1-a)\cdot W(t-1) + a \cdot X(t)$$

όπου:

W(t) = Το βάρος προτίμησης ενός χαρακτηριστικού ή μιας κατηγορίας την παρούσα χρονική περίοδο

W(t-1) = Το βάρος προτίμησης ενός χαρακτηριστικού ή μιας κατηγορίας την προηγούμενη χρονική περίοδο

X(t) = Η χρήση ενός χαρακτηριστικού από το χρήστη την χρονική περίοδο t (συχνότητα επιλογής)

a = μια σταθερά, $0 \le a \le 1$ (forgetting factor)

t = η χρονική περίοδος

Ο κύριος στόχος του αλγορίθμου αυτού είναι να "εξομαλύνει" (smooth) τα δεδομένα ώστε η τάση τους να είναι περισσότερο διακριτή. Με την μέθοδο αυτή είναι ευκολότερο να εντοπιστεί η τάση των δεδομένων και στη συγκεκριμένη περίπτωση σε ποια ενδιαφέροντα τείνει ο χρήστης. Δηλαδή ποιες υπηρεσίες και ποιες πληροφορίες τείνουν να είναι περισσότερο ενδιαφέρον για αυτόν.

Το moving average είναι η μέση τιμή των δεδομένων για ένα συγκεκριμένων αριθμό χρονικών περιόδων. Κινείται γιατί για κάθε υπολογισμό χρησιμοποιεί δεδομένα από τις τελευταίες x χρονικές περιόδους έτσι ώστε να παραμένουν συγχρονισμένα στο παρών. Με το τρόπο αυτό κάνουμε πιο εμφανές το ενδιαφέρον η μη ενδιαφέρον του χρήστη σε ένα χαρακτηριστικό.

Το πόσο σημαντικό είναι το προηγούμενο βάρος προτίμησης και σε ποιό βαθμό πρέπει να ληφθεί υπόψη καθορίζεται από την σταθερά a. Εάν αυτή η σταθερά πάρει την τιμή 0, αυτό σημαίνει ότι δίνουμε σημασία μόνο στην προηγούμενη τιμή του ποσοστού προτίμησης. Εάν αντίθετα πάρει την τιμή 1, σημασία δίνεται μόνο στην τρέχουσα τιμή και αγνοείται η προηγούμενη τιμή τελείως. Για την παρούσα υλοποίηση θεωρήθηκε καλό η τιμή αυτή να δίνει περισσότερο βάρος στη τρέχουσα τιμή του ποσοστού βαρύτητας. Αυτό γιατί το τρέχον ποσοστό βαρύτητας καθορίζεται από τις επιλογές του χρήστη και λαμβάνει υπόψη το σύνολο των εγγραφών που έχουν καταγραφεί στο clickstream του χρήστη για κάποια χρονική περίοδο. Άρα είναι αρκετά αξιόπιστο να παρουσιάζει μια τιμή πολύ κοντά στην πραγματική επιθυμητή τιμή από τον χρήστη. Ωστόσο η προηγούμενη τιμή δεν θα ήταν καλό να παραληφθεί τελείως γιατί μπορεί να μας δώσει σημαντικές πληροφορίες και κυρίως σε αρνητικά ποσοστά. Η τιμή που τελικά επιλέχθηκε μετά από κάποιο αρχικό έλεγχο στο σύστημα για τις τιμές που δίνει, είναι η 0.6666 (2/3). Με την τιμή αυτή δίνουμε

λιγότερο σημασία στις προηγούμενες τιμές αλλά όχι καθόλου, και δίνουμε αντίστοιχα πιο μεγάλη σημασία στις τρέχουσες τιμές.

Η πρώτη συστάδα που δημιουργείται είναι η συστάδα με τα πιο συχνά εμφανιζόμενα χαρακτηριστικά στις επιλογές του χρήστη. Κατ'επέκταση είναι η συστάδα με τα πιο σημαντικά για τον χρήστη χαρακτηριστικά. Τα χαρακτηριστικά της συστάδας αυτής ενημερώνονται με Moving Average, γιατί ο αλγόριθμος αυτός κινείται γρήγορα προς τη τάση των δεδομένων. Έτσι δίνουμε την ευκαιρία στα συχνά εμφανιζόμενα χαρακτηριστικά να πάρουν ένα ψηλό αντιπροσωπευτικό ποσοστό προτίμησης πιο γρήγορα.

Simple Average

Ο υπολογισμός με Simple Average γίνεται με βάση το πιο κάτω τύπο:

$$W(t) = \frac{W(t-1) + X(t)}{2}$$

όπου:

W(t) = Το βάρος προτίμησης ενός χαρακτηριστικού ή μιας κατηγορίας στο τρέχον χρονικό διάστημα για το οποίο μελετούμε το clickstream του χρήστη
W(t-1) = Το βάρος προτίμησης ενός χαρακτηριστικού ή μιας κατηγορίας το αντίστοιχο προηγούμενο χρονικό διάστημα.
X(t) = Η χρήση ενός χαρακτηριστικού από το χρήστη το χρονικό διάστημα t (συχνότητα επιλογής)

Ο μέθοδος αυτή, είναι ο αριθμητικός μέσος όρος στα ποσοστά προτίμησης του χρήστη.

Ο αλγόριθμος αυτός χρησιμοποιείται για να ενημερώσει την δεύτερη συστάδα που δημιουργείται. Η συστάδα αυτή είναι η συστάδα με μέτριου ενδιαφέροντος χαρακτηριστικά έτσι τα χαρακτηριστικά αυτά ενημερώνονται με Simple Average, ο οποίος συνδυάζει την τρέχουσα και προηγούμενη τιμή με την ίδια βαρύτητα. Με το τρόπο αυτό αποφεύγουμε να δώσουμε μεγαλύτερη προτεραιότητα (όπως γίνεται στον moving average) ή καθόλου προτερεότητα (όπως γίνεται στον current average) στα χαρακτηριστικά αυτά. Αντίθετα τους δίνουμε μια τιμή για να τα διατηρήσουμε στη μετρίου ενδιαφέροντος κατηγορία.

Current Average

Ο υπολογισμός του Current Average γίνεται με βάση το πιο κάτω τύπο:

$$W(t) = aX(t)$$

όπου:
X(t) = Η χρήση ενός χαρακτηριστικού από το χρήστη το χρονικό διάστημα t (συχνότητα επιλογής)
a = μια σταθερά, $0 \le a \le 1$ (forgetting factor)
t = το χρονικό διάστημα για το οποίο εξετάζουμε το clickstream του χρήστη.

Ο αλγόριθμος αυτός αποτελεί το δεύτερο κομμάτι του moving average. Με το τρόπο αυτό αγνοούμε το προηγούμενο ποσοστό του χαρακτηριστικού και λαμβάνουμε υπόψη μόνο τις τρέχον επιλογές του χρήστη. Έτσι αναγκάζουμε το ποσοστό προτίμησης στα χαρακτηριστικά αυτά που είναι χαμηλά στις προτιμήσεις του χρήστη, να πάρουν το τρέχον ποσοστό τους. Με το τρόπο αυτό επιτυγχάνουμε να επιβραδύνουμε την τάση του ποσοστού αυτού να αυξηθεί.

Ο αλγόριθμος αυτός χρησιμοποιείται για να ενημερώσει την τρίτη συστάδα που δημιουργείται και είναι η συστάδα με χαμηλού ενδιαφέροντος χαρακτηριστικά. Με τον τρόπο αυτό επιτυγχάνουμε να αγνοούμε προηγούμενα ποσοστά που πιθανώς να δίνουν ψευδή τάση για τα δεδομένα και να έτσι τα χαρακτηριστικά αυτά έχουν χαμηλή προτεραιότητα κατά την επιλογή τους.

5.4.2 Moving Average Clickstream Update Algorithm

Ο αλγόριθμος αυτός ακολουθεί την ίδια διαδικασία όπως και ο αλγόριθμος Clustered Clickstream Update Algorithm αλλά ενημερώνει όλα τα χαρακτηριστικά με τον ίδιο τρόπο. Δεν δημιουργεί ομάδες προτίμησης και δεν χρησιμοποιεί διαφορετικούς αλγορίθμους για κάθε ομάδα.

Στον αλγόριθμο αυτό όλα τα χαρακτηριστικά ενημερώνονται με την χρήση του Moving Average όπως περιγράφτηκε στη προηγούμενη παράγραφο. Έτσι εντοπίζουμε από νωρίς την τάση για όλα τα ποσοστά προτίμησης στο προφίλ.

5.4.3 Flat Clickstream Update Algorithm

Ο αλγόριθμος αυτός ακολουθεί την ίδια διαδικασία όπως και ο αλγόριθμος Clustered Clickstream Update Algorithm αλλά ενημερώνει όλα τα χαρακτηριστικά με τον ίδιο τρόπο. Δεν δημιουργεί ομάδες προτίμησης και δεν χρησιμοποιεί διαφορετικούς αλγορίθμους για κάθε ομάδα.

Στον αλγόριθμο αυτό όλα τα χαρακτηριστικά ενημερώνονται απλά με το τρέχον ποσοστό που τους δίνεται με βάση το πόσες φορές ο χρήστης επέλεξε υπηρεσία που να περιέχει το συγκεκριμένο χαρακτηριστικό. Έτσι δίνουμε στα χαρακτηριστικά το τρέχον ποσοστό προτίμησης που τους αναλογεί με βάση το clickstream του χρήστη.

Σκοπός μας είναι τελικά να συγκρίνουμε τα αποτελέσματα και την απόδοση και των τριών πιο πάνω αλγορίθμων και να εξάγουμε τα αντίστοιχα συμπεράσματα για την ποιότητά τους.

5.5 Αλγόριθμοι Clickstream για ενημέρωση των χρονικών περιόδων (Time Zones) του προφίλ

Έχοντας το Clickstream του χρήστη, είναι πολύ πιο εύκολο να εντοπίσουμε τις δραστηριότητές του κατά τη διάρκεια της μέρας. Για κάθε υπηρεσία μπορούμε να διαχωρίσουμε, χρησιμοποιώντας το timestamp των εγγραφών στο clickstream, τις χρονικές περιόδους που ο χρήστης τείνει να χρησιμοποιεί μια συγκεκριμένη υπηρεσία. Έτσι μπορούμε να προσφέρουμε στο χρήστη χρονικά εξατομικευμένες υπηρεσίες με βάση την ώρα της αναζήτησης.

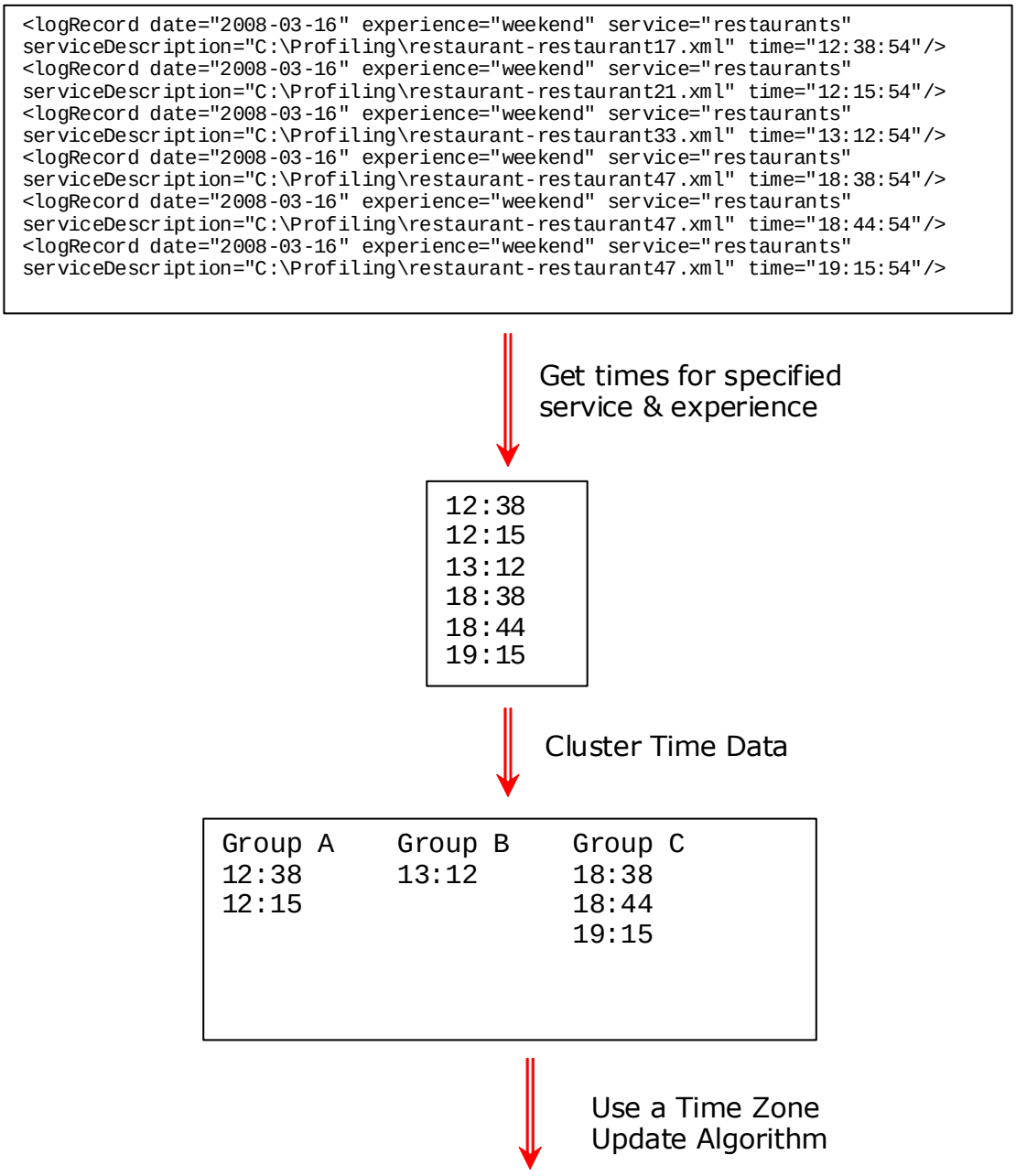
Για την υλοποίηση της πιο πάνω μεθοδολογίας αναλύουμε τρεις διαφορικούς αλγορίθμους.

5.5.1 Προεπεξεργασία εγγραφών clickstream

Η βασική ιδέα των αλγορίθμων που επεξεργάζονται το clickstream είναι η συλλογή των εγγραφών που υπάρχουν σε αυτό για μια συγκεκριμένη υπηρεσία και κατ’ επέκταση η επεξεργασία των εγγραφών αυτών ώστε να πάρουμε τις πληροφορίες που θέλουμε για την υπηρεσία αυτή.

Επιλέγουμε από το clickstream του χρήστη τις εγγραφές που αναφέρονται στη συγκεκριμένη υπηρεσία και για το συγκεκριμένο experience που εξετάζουμε. Στη

συνέχεια επεξεργαζόμαστε το timestamp από κάθε τέτοια εγγραφή, και παίρνουμε από αυτή μόνο το μέρος που μας δίνει την ώρα της αναζήτησης. Αυτό το σύνολο με τις ώρες αναζήτησης στην υπηρεσία που μας ενδιαφέρει, το περνούμε σαν είσοδο σε αλγόριθμο clustering, σκοπός του οποίου είναι να διαχωρίσει το σύνολο αυτό σε μικρότερα υποσύνολα. Έτσι διαχωρίζουμε τις επιλογές του χρήστη, σε χρονικές ομάδες. Στη συνέχεια επεξεργαζόμαστε με τρεις διαφορετικούς αλγορίθμους τις χρονικές αυτές ομάδες, έτσι ώστε να τις διαφοροποιήσουμε με βάση την πυκνότητα των εγγραφών που ανήκουν σε αυτές.



5.5.2 Flat Clickstream Time Zones Update Algorithm

Ο αλγόριθμος αυτός δεν διασπά επιπλέον ούτε ενώνει τις χρονικές περιόδους που εξάγονται από τον αλγόριθμο clustering. Παίρνει τις χρονικές περιόδους που εξάγονται και στη συνέχεια απλά προσθέτει μηδενικές χρονικές περιόδους στις ενδιάμεσες περιόδους στις οποίες ο χρήστης δεν ζητά την συγκεκριμένη υπηρεσία.

Ο αλγόριθμος αυτός συνοψίζεται πιο κάτω:

1. Αρχικά παίρνουμε τις εγγραφές από το αρχείο εγγραφών που σχετίζονται με την υπηρεσία και το experience που θέλουμε να ανανεώσουμε.

2. Στη συνέχεια μελετούμε το timestamp κάθε εγγραφής και χωρίζουνε τις εγγραφές σε N ομάδες χρησιμοποιώντας ένα αλγόριθμο clustering. Ο αλγόριθμος αυτός, διαχωρίζει τις εγγραφές με βάση την ώρα της αναζήτησης. Δηλαδή δημιουργεί ομάδες με εγγραφές που βρίσκονται χρονικά κοντά.

3. Οι ομάδες που δημιουργούνται, δεν δημιουργούν συμπληρωμένες χρονικές περιόδους που σχηματίζουν τελικά στο σύνολό τους ένα εικοσιτετράωρο. Έτσι γεμίζουμε τα κενά με μηδενικές χρονικές περιόδους, δηλαδή περιόδους που έχουν ποσοστό προτίμησης για την συγκεκριμένη υπηρεσία μηδέν. Από τη στιγμή που μια χρονική περίοδος δεν ανήκει στις ομάδες που εξάχθηκαν από τον αλγόριθμο clustering, σημαίνει ότι δεν υπάρχει κάποια εγγραφή σε αυτά τα διαστήματα, και έτσι μπορούμε να θεωρήσουμε ότι ο χρήστης δεν ενδιαφέρεται για την συγκεκριμένη υπηρεσία αυτές τις χρονικές στιγμές.

4. Το ποσοστό προτίμησης που δίνεται σε κάθε χρονική περίοδο είναι ο αριθμός των εγγραφών που υπάρχουν στη συγκεκριμένη περίοδο διά των ολικών εγγραφών για τη συγκεκριμένη υπηρεσία και experience.

$$\Pr eference = \frac{\#\log \mathrm{Re}cordsInTimeZone}{\#Total\ \mathrm{Re}cords}$$

5.5.3 Density Based Clickstream Time Zones Update Algorithm

Η διαφορά του αλγορίθμου αυτού, από τον προηγούμενο αλγόριθμο είναι στο ότι χωρίζει στο διπλάσιο αριθμό ομάδων τις εγγραφές που υπάρχουν στο αρχείο

εγγραφών για μια συγκεκριμένη υπηρεσία και στη συνέχεια, προσπαθεί να τις ενώσει μόνο εάν πληρούν μία συγκεκριμένη συνθήκη.

Συγκεκριμένα ελέγχει την πυκνότητα των εγγραφών σε γειτονικές χρονικές περιόδους, εάν αυτή είναι όμοια κατά ένα ποσοστό, τότε ενώνει τα χρονικά διαστήματα.

Ο αλγόριθμος αυτός εξαρτάται σημαντικά από το ποσοστό ομοιότητας των δύο χρονικών περιόδων. Εάν το ποσοστό αυτό είναι πολύ μεγάλο κινδυνεύουμε να έχουμε πολλές χρονικές περιόδους με μικρά ποσοστά προτίμησης. Εάν πάλι το ποσοστό ομοιότητας είναι πολύ μεγάλο, τότε κινδυνεύουμε να έχουμε μεγάλες χρονικές περιόδους με μεγάλα ποσοστά προτίμησης τα οποία δεν τις αντιπροσωπεύουν. Η ιδανική τιμή για το ποσοστό ομοιότητας μπορεί να καθοριστεί ελέγχοντας την απόδοση του προφίλ ενός χρήστη, σε διαφορετικές τιμές. Έτσι η τιμή του ποσοστού αυτού δίνεται σαν παράμετρος στο σύστημα και με το τρόπο αυτό μπορούμε να μελετήσουμε την ιδανική τιμή που θα μπορούσε να πάρει το ποσοστό αυτό.

Ο αλγόριθμος αυτός θεωρητικά πλεονεκτεί από τον προηγούμενο γιατί προσπαθεί να ενώσει χρονικές περιόδους με όμοια πυκνότητα εγγραφών. Έτσι αναμένουμε να έχουμε πιο ποιοτικές χρονικές περιόδους, που θα έχουν πιο αντιπροσωπευτικό ποσοστό προτίμησης. Αποφεύγουμε να έχουμε γειτονικά όμοιες χρονικές περιόδους σε ότι αφορά τη πυκνότητα τους σε εγγραφές και κατ' επέκταση το ποσοστό προτίμησης, αφού το ποσοστό προτίμησης είναι αναλογία των εγγραφών που ανήκουν σε μια χρονική περίοδο δια του συνολικού αριθμού των εγγραφών.

Ο αλγόριθμος αυτός συνοψίζεται πιο κάτω:

1. Αρχικά παίρνουμε τις εγγραφές από το αρχείο εγγραφών που σχετίζονται με την υπηρεσία και το experience που θέλουμε να ανανεώσουμε.

2. Στη συνέχεια μελετούμε το timestamp κάθε εγγραφής και χωρίζουνε τις εγγραφές σε N x 2 ομάδες χρησιμοποιώντας ένα αλγόριθμο clustering. Ο αλγόριθμος αυτός, διαχωρίζει τις εγγραφές με βάση την ώρα της αναζήτησης. Δηλαδή δημιουργεί ομάδες με εγγραφές που βρίσκονται χρονικά κοντά.

3. Οι ομάδες που δημιουργούνται, δεν δημιουργούν συμπληρωμένες χρονικές περιόδους που σχηματίζουν τελικά στο σύνολό τους ένα εικοσιτετράωρο. Έτσι γεμίζουμε τα κενά με μηδενικές χρονικές περιόδους, δηλαδή περιόδους που έχουν ποσοστό προτίμησης για την συγκεκριμένη υπηρεσία μηδέν. Από τη στιγμή που μια χρονική περίοδος δεν ανήκει στις ομάδες που εξάχθηκαν από τον αλγόριθμο clustering, σημαίνει ότι δεν υπάρχει κάποια εγγραφή σε αυτά τα διαστήματα, και έτσι μπορούμε να θεωρήσουμε ότι ο χρήστης δεν ενδιαφέρεται για την συγκεκριμένη υπηρεσία τη αυτές τις χρονικές στιγμές.

4. Στη συνέχεια παίρνουμε τις χρονικές περιόδους που τελικά σχηματίστηκαν από το προηγούμενο βήμα και ενώνουμε αυτές που έχουν όμοια πυκνότητα εγγραφών. Η πυκνότητα των εγγραφών μιας χρονικής περιόδου δίδεται από των αριθμό των εγγραφών που υπάρχουν στη χρονική αυτή περίοδο δια τη διάρκεια της χρονικής αυτής περιόδου όπως φαίνεται πιο κάτω:

$$TZdensity = \frac{\#Log\,\mathrm{Re}cord\,\sin TimeZone}{TimeZoneDuration}$$

Για να καθορίσουμε κατά πόσο δύο γειτονικές χρονικές περίοδοι έχουν όμοια πυκνότητα εγγραφών, ορίζουμε ένα βαθμό ομοιότητας και ελέγχουμε κάθε φορά κατά πόσο οι δύο χρονικές περίοδοι έχουν κατά το βαθμό ομοιότητας ίδια πυκνότητα.

Παράδειγμα:

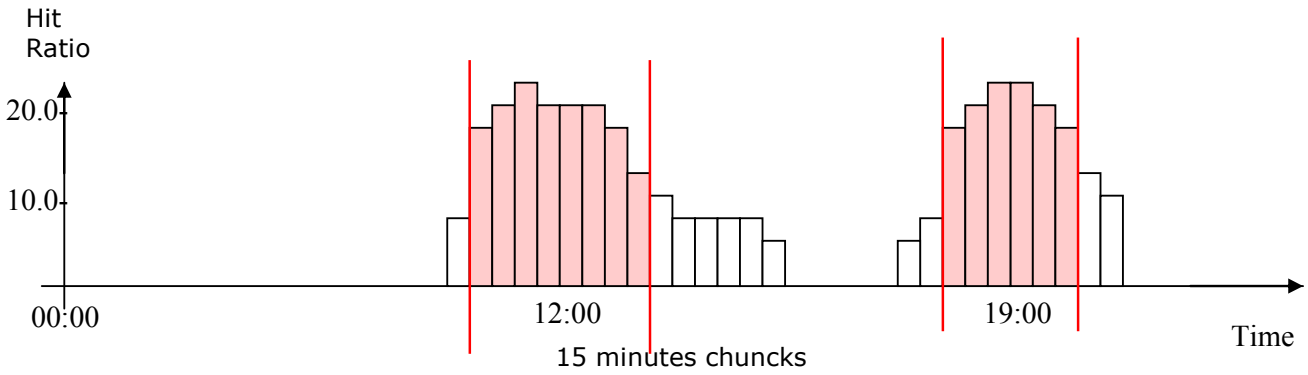
Εάν για παράδειγμα ο βαθμός ομοιότητας είναι 10 τότε βρίσκουμε την πυκνότητα Α και Β για τις δύο χρονικές περιόδους αντίστοιχα. Θεωρούμε ότι οι δύο χρονικές περίοδοι είναι όμοιες εάν ικανοποιούν τη συνθήκη: B-B*10% < A< B + B*10%.

5. Το ποσοστό προτίμησης που δίνεται σε κάθε χρονική περίοδο είναι ο αριθμός των εγγραφών που υπάρχουν στη συγκεκριμένη περίοδο διά των ολικών εγγραφών για τη συγκεκριμένη υπηρεσία και experience.

$$\Pr eference = \frac{\#\log \mathrm{Re}cordsInTimeZone}{\#Total\,\mathrm{Re}cords}$$

5.5.4 Histogram Clickstream Time Zones Update

Ο αλγόριθμος αυτός προσπαθεί να δημιουργήσει ένα ιστόγραμμα της συμπεριφοράς του χρήστη κατά τη διάρκεια του χρόνου. Έτσι για κάθε υπηρεσία σε κάθε experience χωρίζει το χρόνο σε πολύ μικρές χρονικές περιόδους και βρίσκει τον αριθμό εγγραφών στις περιόδους αυτές. Με το τρόπο αυτό τελικά σχηματίζει ένα ιστόγραμμα της πιο κάτω μορφής:



Στη συνέχεια ο αλγόριθμος προσπαθεί να ενώσει τις μικρές αυτές περιόδους ώστε να αντιπροσωπεύουν τις χρονικές περιόδους που ο χρήστης δείχνει ενδιαφέρον για την συγκεκριμένη υπηρεσία. Για να το κάνει αυτό συγκρίνεται η τυπική απόκληση της τρέχον χρονικής περιόδου με την τυπική απόκληση της χρονικής περιόδου που δημιουργείται όταν προσθέσουμε σε αυτή ακόμα ένα μικρό διάστημα.

Η τυπική απόκληση μας δίνει την διασπορά των δεδομένων, στην περίπτωσή μας, των clicks. Όσο πιο μικρή είναι αυτή η τιμή τόσο πιο μαζεμένα βρίσκονται τα δεδομένα. Αυτό σημαίνει ότι έχουμε κοντινά clicks τα οποία θα μπορούσαν να ομαδοποιηθούν για να μας δώσουν μια χρονική περίοδο στο προφίλ του χρήστη. Χρησιμοποιώντας το μέτρο αυτό, και ανάλογα με την τιμή που έχουμε θέσει σαν κατώφλι, επιτυγχάνουμε να σπάσουμε το σύνολο των clicks του χρήστη σε μικρότερα χρονικά διαστήματα με τέτοιο τόπο ώστε να μην έχουμε ούτε πολλά μικρά χρονικά διαστήματα την στιγμή που θα μπορούσαμε να τα συμμαζέψουμε σε ένα μεγαλύτερο, ούτε πολλά μεγάλα με μεγάλη διασπορά που θα είχε ως αποτέλεσμα το ποσοστό προτίμησης σε αυτό το διάστημα να μην είναι αντιπροσωπευτικό. Δημιουργώντας χρονικές περιόδους στις οποίες κρατούμε μικρή την τιμή της διασποράς, επιτυγχάνουμε να έχουμε πιο αντιπροσωπευτικό ποσοστό προτίμησης για την τιμή αυτή.

Το κατώφλι που χρησιμοποιούμε για την μέγιστη τυπική απόκληση που επιτρέπουμε να υπάρχει σε μια χρονική περίοδο, δεν είναι σταθερό. Αυτό γιατί το κατώφλι της διασποράς πρέπει να είναι συναρτήσει των clicks που υπάρχουν σε μια χρονική περίοδο. Με αυτό το τρόπο προσαρμόζουμε την τυπική απόκληση στον αριθμό των clicks. Πιο συγκεκριμένα εάν σε δύο δεκαπεντάλεπτα ιστογράμματα έχουμε 25 και 30 clicks ταυτόχρονα τότε πιθανότατα να επιθυμούμε να ενώσουμε τα ιστογράμματα αυτά. Εάν όμως έχουμε 250 και 300 και πάλι επιθυμούμε να ενώσουμε τα ιστογράμματα αυτά. Ωστόσο η τυπική απόκληση στη πρώτη περίπτωση είναι 5 ενώ στη δεύτερη είναι 50. Η αναλογία των ποσοστών όμως παραμένει η ίδια. Έτσι το κατώφλι που χρησιμοποιούμε είναι ένα ποσοστό πάνω στα clicks.

Πιο συγκεκριμένα ακολουθούμε τον πιο κάτω αλγόριθμο:

1. Αρχικά παίρνουμε τις εγγραφές από το αρχείο εγγραφών που σχετίζονται με την υπηρεσία και το experience που θέλουμε να ανανεώσουμε.

2. Στη συνέχεια μελετούμε το timestamp κάθε εγγραφής και χωρίζουνε τις εγγραφές σε Ν ομάδες χρησιμοποιώντας ένα αλγόριθμο clustering. Ο αλγόριθμος αυτός, διαχωρίζει τις εγγραφές με βάση την ώρα της αναζήτησης. Δηλαδή δημιουργεί ομάδες με εγγραφές που βρίσκονται χρονικά κοντά.

3. Οι ομάδες που δημιουργούνται, δεν δημιουργούν συμπληρωμένες χρονικές περιόδους που σχηματίζουν τελικά στο σύνολό τους ένα εικοσιτετράωρο. Έτσι γεμίζουμε τα κενά με μηδενικές χρονικές περιόδους, δηλαδή περιόδους που έχουν ποσοστό προτίμησης για την συγκεκριμένη υπηρεσία μηδέν. Από τη στιγμή που μια χρονική περίοδος δεν ανήκει στις ομάδες που εξάχθηκαν από τον αλγόριθμο clustering, σημαίνει ότι δεν υπάρχει κάποια εγγραφή σε αυτά τα διαστήματα, και έτσι μπορούμε να θεωρήσουμε ότι ο χρήστης δεν ενδιαφέρεται για την συγκεκριμένη υπηρεσία τη αυτές τις χρονικές στιγμές.

4. Όλες τις μη μηδενικές χρονικές περιόδους τις σπάζουμε σε 15λέπτα ιστογράμματα.

5. Ενώνουμε τα 15λεπτα ιστογράμματα βρίσκοντας κάθε φορά την τυπική απόκληση των δύο χρονικών περιόδων που πιθανώς να ενωθούν ως εξής:

- Εάν οι δύο χρονικές περίοδοι που θέλουμε να ενώσουμε δεν αναλογούν στην ίδια αντιστοιχία clicks για την ίδια χρονική περίοδο, τότε κανονικοποιούμε βρίσκοντας την αντιστοιχία αυτή με τον πιο κάτω τύπο:

$$N' = \frac{T}{NxT'}$$

Όπου:

N’: ο νέος αριθμός clicks

N : Ο αριθμός των clicks που αντιστοιχεί σε χρόνο T

T’: Το νέο χρονικό διάστημα στο οποίο θέλουμε να προσαρμόσουμε τα clicks της χρονικής περιόδου. Συνήθως χρησιμοποιούμε τη χρονική περίοδο που έχει το χρονικά πιο μικρό διάστημα από τις δύο χρονικές περιόδους που θέλουμε να ενώσουμε.

- Βρίσκουμε το κατώφλι που αντιστοιχεί στα clicks της χρονικής περιόδου με τα πιο πολλά clicks ως εξής:

$$T = aN$$

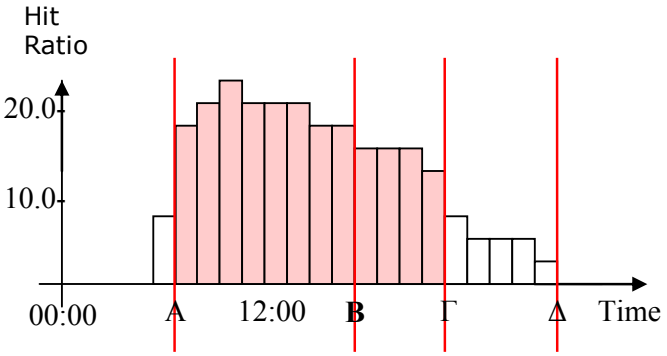
Όπου:

T: το νέο κατώφλι

a: το ποσοστό που καθορίζει το κατώφλι

N: ο αριθμός των clicks στη χρονική περίοδο που εξετάζουμε

Το a είναι η πιο ευαίσθητη παράμετρος του αλγορίθμου. Η τιμή της θα καθορίσει εάν θα ενωθούν δύο χρονικές περίοδοι ή όχι. Η τιμή που βρέθηκε κατά την διαδικασία ελέγχου να προσαρμόζεται καλύτερα στις ανάγκες της εργασίας είναι το 0.3. Μεγαλύτερες τιμές μας έδιναν πολλές και μικρές χρονικές περιόδους, ενώ πιο μικρές τιμές μας έδιναν πολύ μεγάλες χρονικές περιόδους. Και στις δύο αυτές περιπτώσεις οι χρονικές περίοδοι που εμφανίζοντας δεν αντιπροσώπευαν το χρήστη. Μια σχηματική αναπαράσταση των πιο πάνω φαίνεται στο πιο κάτω σχήμα:



Μια μεγάλη τιμή στο α, θα μας έδινε τρεις διαφορετικές χρονικές περιόδους. Την ΑΒ, ΒΓ και ΓΔ. Μια πιο μικρή τιμή στο α, θα μας έδινε μόνο μια περίοδο, την ΑΔ. Η επιλεγμένη τιμή για το α, θα μας δώσει τις χρονικές περιόδους ΑΓ και ΓΔ που είναι το επιθυμητό.

- Βρίσκουμε την τυπική απόκληση. Ο τύπος που δίνει την τυπική απόκληση δίνεται πιο κάτω:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2}.$$

Ωστόσο επειδή στην περίπτωσή μας το *N = 2* ο τύπος αυτός γίνεται:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{N}}$$

$$= \sqrt{\frac{(x_1 - \frac{x_1 + x_2}{2})^2 + (x_2 - \frac{x_1 + x_2}{2})^2}{2}}$$

$$= \sqrt{\frac{(\frac{x_1 - x_2}{2})^2 + (\frac{x_2 - x_1}{2})^2}{2}}$$

$$= \sqrt{\frac{x_1^2 - 2x_1x_2 + x_2^2 + x_2^2 - 2x_1x_2 + x_1^2}{8}}$$

$$= \sqrt{\frac{2x_1^2 - 4x_1x_2 + 2x_2^2}{2 \cdot 4}}$$

$$= \sqrt{\frac{(x_1 - x_2)^2}{4}}$$

$$= \left|\frac{(x_1 - x_2)}{2}\right|$$

Όπου Χ1, Χ2 οι τιμές προτίμησης στις δύο περιόδους.

- Εάν η τυπική απόκληση που βρήκαμε για τις δύο αυτές χρονικές περιόδους ικανοποιεί το κατώφλι που τους αντιστοιχεί, τότε τις ενώνουμε.

6. Το ποσοστό προτίμησης που δίνεται σε κάθε χρονική περίοδο είναι ο αριθμός των εγγραφών που υπάρχουν στη συγκεκριμένη περίοδο διά των ολικών εγγραφών για τη συγκεκριμένη υπηρεσία και experience.

$$\Pr eference = \frac{\#\log \mathrm{Re} cordsInTimeZone}{\#Total\ \mathrm{Re} cords}$$

Κεφάλαιο 6

Αρχικά Προφίλ Χρηστών.

Για την διαδικασία δημιουργίας των αρχικών προφίλ, υπάρχουν τρεις διαδικασίες οι οποίες θα πρέπει να καθοριστούν:

- Με ποια κριτήρια ομαδοποιούμε όμοιους χρήστες
- Πώς καθορίζουμε το default προφίλ των όμοιων χρηστών και πότε το ενημερώνουμε
- Με βάση ποια κριτήρια τοποθετούμε ένα χρήστη σε μια ομάδα χρηστών.

6.1 Κριτήρια Ομαδοποίηση Χρηστών

Αυτό που χρειαζόμαστε είναι ομάδες χρηστών, οι οποίοι έχουν κοινά ενδιαφέροντα. Για την δημιουργία των ομάδων αυτών, πρέπει να λάβουμε υπόψη την ομοιότητα των χρηστών σε ότι αφορά τα κοινά τους ενδιαφέροντα καθώς και την ομοιότητά τους σε ότι αφορά τις τα ενδιαφέροντά τους σε σχέση με τις χρονικές περιόδους που τα ζητούν.

Για να μπορέσουμε να ομαδοποιήσομε χρήστες με όμοια συμπεριφορά, χρησιμοποιούμε αλγορίθμους συσταδοποίησης. Συγκεκριμένα χρησιμοποιήθηκε ο αλγόριθμος k-means ελαφρά διαφοροποιημένος ώστε να προσαρμοστεί στην εύρεση

όμοιων προφίλ. Η πρόκληση στην διαφοροποίηση του αλγορίθμου εκτεινόταν στους δύο πιο κάτω άξονες:

- Το καθορισμό του αλγόριθμου εύρεσης της διαφοράς δύο αντικειμένων, στην περίπτωσή μας δύο προφίλ.
- Στο πώς καθορίζουμε το κέντρο μιας συστάδας.

6.1.1 Διαφορά μεταξύ δύο προφίλ

Συνήθως κατά την εξόρυξη δεδομένων χρησιμοποιούνται σταθερά χαρακτηριστικά τα οποία παίρνουν συγκεκριμένες τιμές (αριθμητικές ή όχι) και είναι συγκεκριμένου αριθμού. Σε αντίθεση το προφίλ ενός χρήστη δεν είναι απλά ένα σύνολο από συγκεκριμένα χαρακτηριστικά που παίρνουν κάποιες προκαθορισμένου τύπου τιμές. Το προφίλ ενός χρήστη δεν περιγράφει απλά κάποιες τιμές χαρακτηριστικών, αλλά και τη σχέση μεταξύ τους.

Για το λόγο αυτό στην εύρεση της διαφοράς δύο προφίλ ο ορίστηκε ο πιο κάτω αλγόριθμος. Ο αλγόριθμος αυτός συγκρίνει δύο προφίλ χρηστών και βρίσκει την ομοιότητα μεταξύ τους σε ότι αφορά τις υπηρεσίες που προτιμούν και τα χαρακτηριστικά τους, αλλά και σε ότι αφορά τις χρονικές περιόδους που χρειάζονται τις υπηρεσίες αυτές και τα χαρακτηριστικά τους.

Αλγόριθμος εύρεσης της απόσταση μεταξύ των χαρακτηριστικών δύο προφίλ

Για την απόσταση μεταξύ δύο προφίλ απαιτείται κάτι περισσότερο από απλά σύγκριση των κοινών υπηρεσιών και χαρακτηριστικών. Αυτό, γιατί στα προφίλ των χρηστών κρατούμε τα ποσοστά προτίμησης που είναι αυτά στην ουσία που μας δίνουν τα ενδιαφέροντα του χρήστη. Η ύπαρξη απλά ενός χαρακτηριστικού δεν μας λέει και πάρα πολλά. Αυτό γιατί, το ποσοστό προτίμησης μπορεί να κυμανθεί από 100 έως και -1 για χαρακτηριστικά που δεν ενδιαφέρουν καθόλου το χρήστη. Άρα η ύπαρξη ενός χαρακτηριστικού στο προφίλ δεν μας δίνει από μόνο στοιχεία για το κατά πόσο αρέσει ή όχι το χαρακτηριστικό αυτό σε ένα χρήστη.

Έτσι, η απόσταση μεταξύ δύο προφίλ είναι απαραίτητο να εξαρτάται κυρίως από την διαφορά στα ποσοστά προτίμησης των κοινών χαρακτηριστικών. Ωστόσο θα πρέπει να δώσουμε κάποια ξεχωριστή σημασία και απόσταση στα χαρακτηριστικά τα οποία δεν είναι κοινά και στα χαρακτηριστικά τα οποία έχουν ποσοστό προτίμησης -

1.Ο αλγόριθμος που ακολουθείτε για την εύρεση της απόστασης μεταξύ δύο προφίλ δίνετε πιο κάτω:

- Για να υπολογίσουμε το ποσοστό προτίμησης ενός στιγμιοτύπου πολλαπλασιάζουμε το ποσοστό προτίμησης για αυτό το χαρακτηριστικό με το ποσοστό προτίμησης της υποκατηγορίας στην οποία ανήκει. Ομοίως χειριζόμαστε και τα ποσοστά προτίμησης των υποκατηγοριών.

- Για τα κοινά στιγμιότυπα, υποκατηγορίες και χαρακτηριστικά που έχουν θετικό ποσοστό προτίμησης:

Distance($x_{1,\dots,n}$,$y_{1,\dots,n}$)=*distance*($x_{1,\dots,n-1}$,$y_{1,\dots,n-1}$)+*difference*(x_n,y_n)**difference*(x_n,y_n)
Difference(x_n,y_n) = *norm*(x_n)-*norm*(y_n)
Norm(x_n)=(x_n - *Min*)/(*Max-Min*)
όπου n ο αριθμός στιγμιοτύπων μιας κατηγορίας, ή ο αριθμός υποκατηγοριών μιας κατηγορίας.

- Για τα στιγμιότυπα, κατηγορίες ή υποκατηγορίες που δεν είναι κοινά ή έχουν ο ένας από τους δύο έχει στο συγκεκριμένο χαρακτηριστικό, ποσοστό προτίμησης -1, τότε:

Distance($x_{1,\dots,n}$,$y_{1,\dots,n}$)=*distance*($x_{1,\dots,n-1}$,$y_{1,\dots,n-1}$)+*difference*(x_n,y_n)**difference*(x_n,y_n)
Difference(x_n,y_n) = *norm*(x_n), *όπου* x_n *το υπάρχον χαρακτηριστικό*
Επιπλέον εάν Difference(x_n,y_n)<0,5 → *Difference*(x_n,y_n)= 1- *Difference*(x_n,y_n)

- Για τα χαρακτηριστικά που έχουν ποσοστό προτίμησης ίσο με -1 αλλά υπάρχουν μόνο στο ένα προφίλ δίνουμε απόσταση ίση με 1

- Η τελική απόσταση είναι το άθροισμα των τριών αυτών αποστάσεων (Στιγμιοτύπων, υποκατηγορίας και κατηγορίας). Ωστόσο η βαρύτητα που δίνουμε στην απόσταση μεταξύ στιγμιοτύπων είναι και πάλι μεγαλύτερη από τις άλλες δύο καθότι είναι και πιο σημαντική. Έτσι η τελική απόσταση δίνεται από την πιο κάτω εξίσωση:

*Distance = a*InstacesDistance + b*SubcategoryDistance+c*CategoryDistance*
Όπου a>b,c

Παράδειγμα:

Υπηρεσία 1
<poultry - 70>
  <chicken - 90>
  <duck - 49>
</poultry>

Υπηρεσία 2
<poultry- 45>
  <chicken - 90>
  <rabbit - -1>
</poultry>

Poultry	70	Poultry	45
poultry/chicken	90*0.7= 63	poultry/chicken	90*0.45=40,5
poultry/duck	49*0.7=34,3	poultry/rabbit	1

Diff (poultry/chicken)=0,63-0,405=0,225
Diff(poultry/duck)=0,343→1-0,343=0,657
Diff(poultry/rabbit)=1
Distance in Instances = $(0,225)^2 + (0,657)^2 + 1^2 = 1,481$

Diff(poultry)=0,7-0,45=0,25
Distance in SubCategories = $(0,25)^2$
Distance in Categories = 0
Total Distance = $\frac{3}{5} * 1,481 + \frac{1}{5} 0,625 + \frac{1}{5} 0 = 0,9$

Αλγόριθμος εύρεσης της απόσταση μεταξύ των χρονικών περιόδων δύο προφίλ

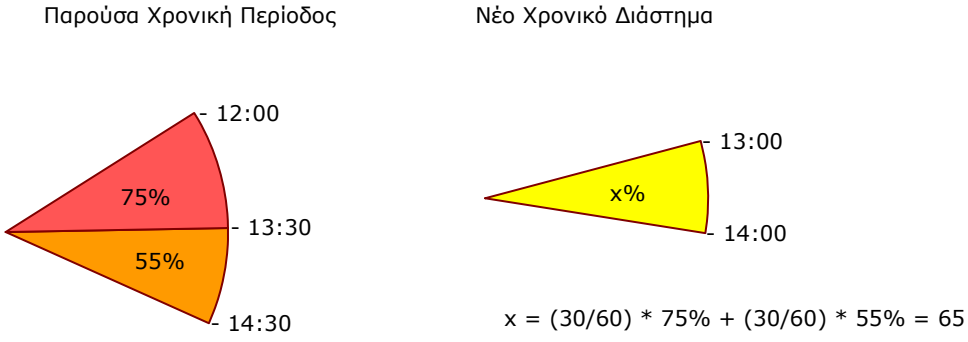
Για την απόσταση εύρεσης των χρονικών περιόδων μεταξύ δύο προφίλ χρειαζόμαστε ένα κοινό τρόπο αναφοράς στις χρονικές περιόδους. Δεν μπορούμε να συγκρίνουμε χρονικές περιόδους με διαφορετικά όρια. Για το λόγω αυτό αλλάζουμε τις χρονικές περιόδους σε κάθε προφίλ ώστε να είναι οι ίδιες και στα δύο. Πιο συγκεκριμένα χωρίζουμε τις χρονικές περιόδους σε διατήματα 1 ώρας αρχίζοντας από το 0 έως το 24.

Το σημαντικό στη διαδικασία αυτή είναι πως θα ανακατανέμουμε το ποσοστό προτίμησης μιας χρονικής περιόδους στα νέα χρονικά διαστήματα που θα την αποτελούν. Σχετικά με την ανακατανομή αυτή, μπορεί να υπάρξουν οι δύο πιο κάτω περιπτώσεις:

1. Το νέο διάστημα να περιλαμβάνεται ολόκληρο μέσα σε μια ήδη υπάρχον χρονική περίοδο. Στη περίπτωση αυτή, το ποσοστό προτίμησης που θα πάρει η νέα χρονικό διάστημα είναι ακριβώς το ίδιο με το ποσοστό προτίμησης της ήδη υπάρχουσας χρονικής περιόδου.

2. Το νέο χρονικό διάστημα περιλαμβάνεται σε δύο ή και περισσότερες γειτονικές χρονικές περιόδους. Στη περίπτωση αυτή ακολουθούμε την πιο κάτω διαδικασία:

   - Βρίσκουμε τη διάρκεια του νέου χρονικού διαστήματος στην κάθε μια υπάρχουσα χρονική περίοδο.
   - Ακολούθως κανονικοποιούμε τη διάρκεια αυτή ώστε να μας δώσει την αναλογία της στο σύνολο του νέου χρονικού διαστήματος. Η συνολική διάρκεια του νέου χρονικού διαστήματος είναι 60 λεπτά. Άρα κανονικοποιούμε διαιρώντας με 60 τη διάρκεια που βρήκαμε ότι αντιστοιχεί στα υπάρχον χρονικά διαστήματα.
   - Η κανονικοποιημένη διάρκεια, θα πολλαπλασιαστεί με το ποσοστό προτίμησης της χρονικής περιόδου που αντιστοιχεί για να μας δώσει το ποσοστό προτίμησης αυτής της διάρκειας στο νέο χρονικό διάστημα που δημιουργούμε. Στο τέλος προσθέτουμε όλα τα ποσοστά προτίμησης που πήραμε για το νέο χρονικό διάστημα. Σχηματικά τα πιο πάνω φαίνονται στο Σχήμα 6.1.



Σχήμα 6.1 : Εύρεση ποσοστού προτίμησης στα νέα χρονικά διαστήματα που χρησιμοποιούνται κατά την σύγκριση των χρονικών περιόδων σε δύο προφίλ.

Στη συνέχεια τα χρονικά διαστήματα στα δύο προφίλ είναι τα ίδια και έτσι μπορούν

να αποτελέσουν χρονικές περιόδους και να συγκριθούν όπως συγκρίνεται ένα οποιοδήποτε άλλο χαρακτηριστικό στο προφίλ.
Η ίδια ακριβώς διαδικασία ακολουθείται και για τα typeByTime. Σε αυτή τη περίπτωση τα διάφορα είδη (types) της υπηρεσία ανακατανέμονται ανάλογα με τις νέες χρονικές περιόδους που δημιουργούνται και τα ποσοστά προτίμησης που παίρνουν καθορίζονται όπως πιο πάνω.

6.1.2 Καθορισμός του κέντρου μιας συστάδας

Ο αλγόριθμος k-means μπορεί να εφαρμοστεί εφόσο μπορούμε να ορίσουμε το κέντρο μιας συστάδας. Ωστόσο αυτό δεν είναι εύκολα εφικτό και όχι πάντα. Στη περίπτωση κατηγορηματικών χαρακτηριστικών (χαρακτηριστικά που παίρνουν διακριτές τιμές, χωρίς να μπορούμε να ορίσουμε κάποια σειρά/βαθμίδα μεταξύ τους), για παράδειγμα, αυτό είναι δύσκολο να υλοποιηθεί. Επιπλέον για να βρεθεί το κέντρο της συστάδας, πρέπει να ληφθούν υπόψη όλες οι οντότητες που αποτελούν την συστάδα αυτή.

Στην παρούσα υλοποίηση το κέντρο της συστάδας ορίστηκε η τομή των προφίλ που την αποτελούν. Με αυτό εννοούμε την δημιουργία ενός προσωρινού προφίλ το οποίο κρατά τις κοινές υπηρεσίες και χαρακτηριστικά. Τα ποσοστά προτίμησης που παίρνουν τα χαρακτηριστικά είναι ο μέσος όρων των αντίστοιχων ποσοστών στα προφίλ που αποτελούν την συστάδα. Σε ότι αφορά τα time zones χρησιμοποιείται παρόμοιος αλγόριθμος με αυτόν που χρησιμοποιείται για την εύρεση της διαφοράς των time zones μεταξύ δύο προφίλ, όπως περιγράφτηκε στην παράγραφο 6.1.1.. Πιο συγκεκριμένα χωρίζουμε τις χρονικές περιόδους σε περιόδους μιας ώρας και βρίσκουμε τα ποσοστά προτίμησης για κάθε ώρα, λαμβάνοντας υπόψη το μέσο όρων των ποσοστών προτίμησης στα προφίλ που αποτελούν την συστάδα. Στη συνέχεια χρησιμοποιούμε τον αλγόριθμο ιστογραμμάτων που περιγράψαμε στο υποκεφάλαιο 5.5.4 για να ενώσουμε τα ιστογράμματα μιας ώρας σε λογικές χρονικές περιόδους για τους χρήστες.

6.1.3 Αλγόριθμος k-means για προφίλ χρηστών

Η διαφοροποίηση του αλγορίθμου k-means ώστε να μπορεί να εφαρμοστεί στην ομαδοποίηση προφίλ χρηστών δίνεται πιο κάτω:

1. Αποφασίζουμε k προφίλ ως τα κέντρα των k αρχικών συστάδων.
2. Βρίσκουμε την απόσταση όλων των προφίλ με τα κέντρα των συστάδων.
3. Τοποθετούμε τα προφίλ στις πιο κοντινές συστάδες

4. Ορίζουμε το κέντρο της κάθε συστάδας ως η τομή των προφίλ που την αποτελούν. Για να το κάνουμε αυτό:
   - Βρίσκουμε τα κοινά paths για δύο προφίλ με τα αντίστοιχα ποσοστά προτίμησης
   - Βρίσκουμε τα ποσοστά προτίμησης κάθε προφίλ για τις χρονικές περιόδους και τις σπάζουμε σε περιόδους μιας ώρας.
   - Ενημερώνουμε ένα άδειο προφίλ μόνο με τα κοινά paths και με ποσοστά προτίμησης τον μέσο όρων των αντίστοιχων ποσοστών προτίμησης. Για τις χρονικές περιόδους εφαρμόζουμε τον αλγόριθμο ιστογράμματος ώστε να ενώσουμε τις περιόδους μιας ώρας σε λογικές χρονικές περιόδους. Έτσι δημιουργούμε το προφίλ που είναι η τομή των δύο προφίλ.
   - Στη συνέχεια επαναλαμβάνουμε τα βήματα 1,2 και 3 χρησιμοποιώντας σαν κέντρο μιας συστάδας, τα προφίλ τομής. Επαναλαμβάνουμε τα βήματα 1,2,3 και 4 έως ότου να μην έχουμε μετακινήσεις προφίλ στις συστάδες.

6.2 Καθορισμός των Default προφίλ χρηστών

Για την δημιουργία των default προφίλ, θα πρέπει να λάβουμε υπόψη το σύνολο των προφίλ των χρηστών που ανήκουν στην ίδια ομάδα. Είναι σημαντικό να καθοριστούν στρατηγικές για το πώς αντιμετωπίζουμε τα πιο κάτω:

Εάν κάποιες κατηγορίες, υποκατηγορίες και στιγμιότυπα υπάρχουν αλλά σε μερικούς χρήστες, πώς καθορίζουμε το αν αυτά τα χαρακτηριστικά θα πρέπει να συμπεριληφθούν στο κοινό προφίλ.

- Ποιο ποσοστό προτίμησης δίνεται σε κάθε χαρακτηριστικό.

- Τα χαρακτηριστικά τα οποία απουσιάζουν από κάποιους χρήστες πως επηρεάζουν το ποσοστό προτίμησης στο κοινό προφίλ.

- Πώς διαμορφώνονται τα Time Zones στο κοινό προφίλ.

Για τα πιο πάνω χειριζόμαστε όλους τους χρήστες μιας ομάδας σαν ένας χρήστης, και ενημερώνουμε ένα αρχικό προφίλ όπως ακριβώς θα ενημερώναμε το προφίλ ενός κανονικού χρήστη. Η μόνη διαφορά είναι ότι για να συμπεριλάβουμε ένα χαρακτηριστικό στο κοινό προφίλ θα πρέπει να εμφανίζεται στο προφίλ τουλάχιστον στο Ν% των χρηστών (π.χ. 30%, 50% κλπ. Μπορούμε να εξετάσουμε διαφορετικές τιμές για να βρούμε τη βέλτιστη τιμή). Επιπλέον η ενημέρωση αυτή θα λαμβάνει

υπόψη όχι το clickstream του κάθε χρήστη αλλά το προφίλ του. Απλά θα χρησιμοποιηθούν οι ίδιες τεχνικές. Το προφίλ που προκύπτει από τη διαδικασία αυτή είναι το κοινό προφίλ που δίνεται στην ομάδα.

6.3 Κριτήρια για την τοποθέτηση ενός νέου χρήστη σε μια ομάδα

Για την ομαδοποίηση χρηστών, κατηγοριοποιούμε τους χρήστες με βάση τα δημογραφικά δεδομένα τους.

Για να γίνει αυτό πρέπει να εξάγουμε τα φυσικά και κοινωνικά χαρακτηριστικά μιας ομάδας χρηστών ώστε να μπορέσουμε να κατηγοριοποιήσουμε νέους χρήστες χρησιμοποιώντας μόνο φυσικά ή κοινωνικά χαρακτηριστικά.

Αναμένουμε ότι χρήστες τις ίδιας ηλικίας, του ίδιου εργασιακού χώρου, της ίδιας μόρφωσης στο σύνολό τους πιθανότατα να συγκλίνουν στις ίδιες προτιμήσεις. Έτσι κατά την εισαγωγή ενός χρήστη στο σύστημα ζητάμε πληροφορίες για τα χαρακτηριστικά αυτά. Αρχικά κατηγοριοποιούμε τους χρήστες με βάση τα ενδιαφέροντά τους, με αλγορίθμους συσταδοποίησης. Στη συνέχεια, με αλγορίθμους classification πάνω στα φυσικά και κοινωνικά χαρακτηριστικά των χρηστών, βρίσκουμε ποια χαρακτηριστικά είναι κοινά και περιγράφουν τους χρήστες κάθε ομάδας.

Έτσι μπορούμε να βασιστούμε στα κοινωνικά – φυσικά χαρακτηριστικά ενός χρήστη για να μπορέσουμε να τον κατατάξουμε σε μία από τις ήδη υπάρχουσες ομάδες.

Κεφάλαιο 7

Έλεγχος Αλγορίθμων και Αποτελέσματα

7.1 Πως καθορίζουμε τη συμπεριφορά των Χρηστών στο Testing

7.1.1 Δημιουργία προφίλ χρηστών

Κατά την δημιουργία των προφίλ για κάθε υπηρεσία καλούνται συναρτήσεις που δίνουν στιγμιότυπα στα χαρακτηριστικά (κατηγορίες) της υπηρεσίας. Για κάθε χαρακτηριστικό παίρνουμε ένα τυχαίο αριθμό στιγμιοτύπων (ανόμοιων μεταξύ τους) που αναφέρονται σε αυτό και τους δίνουμε τυχαία βάρη. Επιπλέον σε κάθε χαρακτηριστικό δίνουμε ένα τυχαίο βάρος (category weight).

Η κατηγορία timeZones συμπληρώνεται αλλά όχι τελείως τυχαία. Αυτό γιατί υπάρχουν κάποιες υπηρεσίες όπως για παράδειγμα τα εστιατόρια, οι οποίες χρησιμοποιούνται συγκεκριμένες ώρες τις μέρες. Σε αυτές τις περιόδους δίνεται μεγαλύτερη πιθανότητα να έχουν ψηλότερο ποσοστό προτίμησης. Επιπλέον για αρχή

στα προφίλ αυτά δίνουμε συγκριμένα timeZones τα οποία μπορούν με την χρήση του προφίλ να αλλάξουν.

Η κατηγορία Type_By_Time συμπληρώνεται αλλά επίσης όχι τελείως τυχαία γιατί κάτι τέτοιο δεν θα μας εξυπηρετούσε αφού το πότε επιλέγεται ο τύπος μιας υπηρεσίας εξαρτάται πολλές φορές και από τον χρόνο. Εδώ απλά δίνονται τυχαία βάρη σε τυχαία είδη της υπηρεσίας. Όταν όμως τα στιγμιότυπα αυτά χωρίζονται στις χρονικές περιόδους δεν χωρίζονται τελείως αλλά με κάποια μεγαλύτερη πιθανότητα στα timeZones που έχουν μεγαλύτερη πιθανότητα να έχουν ψηλότερο ποσοστό προτίμησης.

Με τον τρόπο που περιγράφτηκε πιο πάνω πετυχαίνουμε την δημιουργία προφίλ χρηστών τελείως τυχαία λαμβάνοντας υπόψη μόνο τον παράγοντα χρόνο όπου αυτό είναι απαραίτητο.

7.1.2 Εύρεση της επιθυμητής υπηρεσίας

Για να μπορέσουμε να ελέγξουμε το σύστημα, χρειαζόμαστε το σύστημα να ελέγχου να μπορεί από μόνο του να επιλέγει μια υπηρεσία και ένα στιγμιότυπο σαν τα επιθυμητά από τον χρήστη, και στη συνέχει να υπολογίζει τη θέση στην οποία εμφανίζεται αυτή η υπηρεσία και το στιγμιότυπο στο χρήστη.

Κατά την εύρεση και το καθορισμό της επιθυμητής υπηρεσίας από το σύστημα χρησιμοποιείται το χαρακτηριστικό TimeZones που υπάρχει στο προφίλ του κάθε χρήστη. Το στοιχείο αυτό χωρίζει το εικοσιτετράωρο σε χρονικές περιόδους. Κάθε χρονική περίοδος αντιστοιχείται με ένα βάρος προτίμησης για την υπηρεσία στο συγκεκριμένο διάστημα. Κατά την εύρεση της επιθυμητής υπηρεσίας, βρίσκουμε το αντίστοιχο timeZones στο προφίλ του χρήστη για την υπηρεσία αυτή, και ψάχνουμε το βάρος προτίμησης που αντιστοιχεί στη χρονική στιγμή της αναζήτησης. Στη συνέχεια το σύστημα ταξινομεί τις υπηρεσίες ώστε αυτές με το μεγαλύτερο ποσοστό προτίμησης στο συγκεκριμένο χρονικό διάστημα να είναι πρώτες. Από αυτές τις ταξινομημένες υπηρεσίες, θα αποφασίσει το σύστημα την επιθυμητή υπηρεσία για τον χρήστη. Το κάνει αυτό τυχαία, δίνοντας όμως μεγαλύτερη πιθανότητα στις υπηρεσίες που εμφανίζονται πρώτες στη σειρά ταξινόμησης.

Ο αλγόριθμος που τυχαία επιλέγει την υπηρεσία χωρίς να αγνοεί το βαθμό προτίμησης της από τον χρήστη δίνεται πιο κάτω:

- Το σύστημα παίρνει ένα τυχαίο αριθμό από το 0 έως το 100

- Ανάλογα με το πόσες ταξινομημένες υπηρεσίες υπάρχουν, το σύστημα καθορίζει όρια για κάθε υπηρεσία στα οποία εάν ανήκει αυτός ο αριθμός επιστρέφεται η αντίστοιχη υπηρεσία.

- Τα όρια αυτά είναι μικρότερα για υπηρεσίες που βρίσκονται χαμηλά στην ταξινόμηση και μεγαλύτερα για τις υπηρεσίες που βρίσκονται ψηλά.

- Για παράδειγμα:

Restaurants	50-100
Cafes	25-50
Copycenters	15-25
Bookshop	7-15
Bar	0-7

Στο πιο πάνω παράδειγμα ο τυχαίο αριθμός 75 θα μας επέστρεφε την υπηρεσία εστιατορίων σαν την επιθυμητή για το χρήστη.

7.1.3 Εύρεση του επιθυμητού στιγμιοτύπου

Για την εύρεση του επιθυμητού στιγμιοτύπου της υπηρεσίας που επέλεξε ο χρήστης ακολουθούμε την πιο κάτω διαδικασία:

1. Για κάθε στιγμιότυπο της υπηρεσίας που έχει επιλέξει ο χρήστης, βρίσκουμε τα κοινά του χαρακτηριστικά με τα χαρακτηριστικά της συγκεκριμένης υπηρεσίες που βρίσκονται στο προφίλ του χρήστη.

2. Στη συνέχεια βρίσκουμε το βάρος προτίμησης του συγκεκριμένου στιγμιότυπου της υπηρεσίας που εξετάζουμε με τον πιο κάτω τρόπο: Για κάθε κοινό χαρακτηριστικό στρογγυλοποιούμε το βάρος προτίμησης του ως εξής: Αν είναι μεταξύ 75-100 γίνεται 100, αν είναι μεταξύ 50-75 γίνεται 75, αν είναι μεταξύ 25-50 γίνεται 50 και εάν είναι μεταξύ 0-2 γίνεται 25.

3. Ακολούθως, προσθέτουμε τα βάρη μεταξύ τους και τα διαιρούμε με τον αριθμό τους ώστε να βρούμε το μέσο όρο των χαρακτηριστικών αυτών. Έπειτα τον αριθμό αυτό τον πολλαπλασιάζουμε με το στρογγυλεμένο κατά τον ίδιο τρόπο ποσοστό προτίμησης της κατηγορίας χαρακτηριστικών που βρίσκονται τα χαρακτηριστικά αυτά.

Π.χ

```
<poultry weightCategory="51">
        <poultryType name="Turkey" weight="26"/>
        <poultryType name="Cock" weight="41"/>
        <poultryType name="Duck" weight="23"/>
        <poultryType name="Chicken" weight="65"/>
</poultry>
```

Υπολογισμός ποσοστού προτίμησης:

26+41+23+65= 50+50+25+50=175/4 *51/100=22.31 = 25

4. Αυτό γίνεται για όλες τις κατηγορίες χαρακτηριστικών. Τελικά τα βάρη προτίμησης που αντιστοιχούν σε κάθε κατηγορία προστίθενται και το άθροισμα διαιρείται με τον αριθμό των κατηγοριών αυτών.

5. Με το τρόπο αυτό, κατανέμουμε τα στιγμιότυπα μιας υπηρεσίας σε ομάδες βαρών και επιτυγχάνουμε να έχουμε τις πιο πιθανές προτιμήσεις του χρήστη στη $1^{η}$ ομάδα(100). Από αυτήν, επιλέγουμε τυχαία ένα στιγμιότυπο. Με το τρόπο αυτό προσπαθούμε να εισάγουμε σε κάποιο βαθμό το παράγοντα τύχη.

Η χρήση κάποιου αλγορίθμου που λαμβάνει υπόψη το προφίλ του χρήστη και τα βάρη προτίμησης που υπάρχουν σε αυτό είναι απαραίτητη. Αυτό γιατί το σύστημα από τη φύση του δεν αναμένει κάποια τελείως τυχαία γεγονότα να συμβούν. Τα γεγονότα αυτά συμβαίνουν με βάση κάποια λογική. Την λογική που καθορίζουν οι προτιμήσεις του χρήστη και ο παράγοντας χρόνος. Δύο παράγοντες που αν αγνοηθούν θα οδηγήσουν σε μη βάσιμα αποτελέσματα.

Ο πιο πάνω αλγόριθμος είναι ένας πολύ απλός αλγόριθμος ο οποίος μας δίνει, βασισμένος στο προφίλ του χρήστη, μια ομάδα από τις υπηρεσίες που πιθανότατα να χρειάζεται περισσότερο μια δεδομένη στιγμή.

7.1.4 Δημιουργία Υπηρεσιών για την διαδικασία ελέγχου του συστήματος

Για να ολοκληρώσουμε το κεφάλαιο αυτό που περιγράφει την όλη διαδικασία ελέγχου του συστήματος, πρέπει να αναφέρουμε ότι απαιτείται το σύστημα να έχει ένα μηχανισμό ο οποίος με κάποιο τρόπο θα δημιουργεί στιγμιότυπα υπηρεσιών που θα διατίθενται στο σύστημα. Ο μηχανισμός αυτός έχει υλοποιηθεί και έτσι μπορούμε να έχουμε τυχαία στιγμιότυπα για κάθε υπηρεσία με διαφορετικά χαρακτηριστικά.

7.2 Διαδικασία Ελέγχου

Εάν λάβουμε υπόψη ότι έχουμε να κάνουμε με χρήστες που δεν κάνουν τυχαίες επιλογές αλλά επιλέγουν με βάση τα ενδιαφέροντα και τις προτιμήσεις τους, αναμένουμε ότι όσο πιο πολύ χρησιμοποιείται το σύστημα, το προφίλ πρέπει να συγκλίνει στις επιλογές του χρήστη, και τελικά (θεωρώντας ότι τα ενδιαφέροντα του χρήστη δεν μεταβάλλονται για κάποια Χ περίοδο) να μας δώσει τελικά τέλειο προφίλ για την περίοδο Χ στην οποία τα ενδιαφέροντα του χρήστη δεν μεταβάλλονται τυχαία.

Υπάρχουν δυο μέθοδοι με τους οποίους μπορούμε να ελέγξουμε κατά πόσοι οι αλγόριθμοι ενημέρωσης του προφίλ, βοηθούν στο να έχουμε τελικά προφίλ το οποίο συγκλίνει στα ενδιαφέροντα του χρήστη.

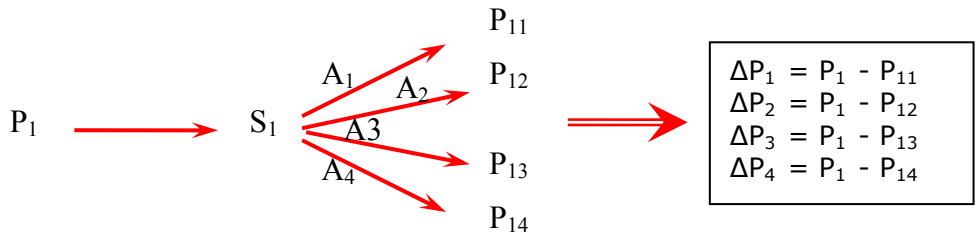
Η πρώτη, είναι εξετάζοντας τη διαφορά μεταξύ του αρχικού και τελικού προφίλ σε τακτά χρονικά διαστήματα, αφού τρέξει πάνω στο αρχικό προφίλ ένα clickstream.

Η δεύτερη είναι εξετάζοντας τα αποτελέσματα του συστήματος μετά από κάποια χρονική περίοδο.

7.2.1 Διαφορά Αρχικού και Τελικού Προφίλ

Για να μπορέσουμε να ελέγξουμε κατά πόσο οι αλγόριθμοι μας βοηθούν το προφίλ να συγκλίνει στα ενδιαφέροντα του χρήστη, τρέχουμε το προφίλ πάνω σε κάποιο clickstream, και συγκρίνουμε το προφίλ που παίρνουμε τελικά με το αρχικό. Αυτό επαναλαμβάνεται μέχρι να παρατηρήσουμε ότι η διαφορά των δύο προφίλ συγκλίνει σε ένα πολύ μικρό αριθμό. Επιπλέον με τη μέθοδο αυτή μπορούμε να εξετάσουμε και το πόσο νωρίς φθάνει στο προφίλ στην ισορροπία.

Για την σύγκριση των δύο προφίλ χρησιμοποιούμε των αλγόριθμο που ορίσαμε για την απόσταση μεταξύ δύο προφίλ στην παράγραφο 6.1.1.



P_1 : Αρχικό Προφίλ
S_1 : $Clickstream_1$
$A_{1...n}$: Διαφορετικοί Αλγόριθμοι Ενημέρωσης Προφίλ
$P_{1...n}$: Τελικά Προφίλ

7.2.2 Σύγκριση αποτελεσμάτων

Για την μέθοδο αυτή, αρχικά τρέχουμε κάποιο clickstream πάνω στο προφίλ του χρήστη και παίρνουμε αποτελέσματα (βλέπε μετρικές). Στη συνέχεια τρέχουμε ξανά ένα δεύτερο clickstream και παίρνουμε και πάλι τα αποτελέσματα του συστήματος. Συγκρίνοντας τα δύο αποτελέσματα μπορούμε να βγάλουμε τα εξής συμπεράσματα:

- Εάν τα αποτελέσματα γίνονται χειρότερα τότε αυτό σημαίνει ότι οι αλγόριθμοι δεν βοηθούν στο να συγκλίνει το προφίλ ή ότι έχουμε ένα χρήστη ο οποίος χρησιμοποιεί τυχαία το σύστημα.

- Εάν τα αποτελέσματα είναι τα ίδια, τότε ή είχαμε τέλειο προφίλ από την αρχή ή ο αλγόριθμός μας δεν βοηθά στην βελτίωση του προφίλ με το να προσομοιώνει τα ενδιαφέροντα του χρήστη.

- Εάν τα αποτελέσματα είναι καλύτερο στις δεύτερες μετρήσεις ο αλγόριθμος συγκλίνει άρα μπορούμε να τον θεωρήσουμε επιτυχημένο

Ωστόσο στο σενάριο αυτό υπάρχει η πιθανότητα το σύστημα να προτείνει όμοιες υπηρεσίες στο χρήστη κι αυτός να προτιμήσει κάποια που να μην είναι, για παράδειγμα, μέσα στις τρεις πρώτες προτεινόμενες, από το σύστημα, επιλογές. Ωστόσο ο χρήστης επιλέγει και πάλι υπηρεσία που είναι κοντά στα ενδιαφέροντά του, αυτή η επιλογή του όμως μπορεί να έχει σημαντική επίδραση στις μετρήσεις.

7.3 Απόσταση μεταξύ δύο προφίλ

Για την απόσταση μεταξύ δύο προφίλ απαιτείται κάτι περισσότερο από απλά σύγκριση των κοινών υπηρεσιών και χαρακτηριστικών. Αυτό, γιατί στα προφίλ των χρηστών κρατούμε τα ποσοστά προτίμησης που είναι αυτά στην ουσία που μας δίνουν τα ενδιαφέροντα του χρήστη. Η ύπαρξη απλά ενός χαρακτηριστικού δεν μας λέει και πάρα πολλά. Αυτό γιατί, το ποσοστό προτίμησης μπορεί να κυμανθεί από 100 έως και -1 για χαρακτηριστικά που δεν ενδιαφέρουν καθόλου το χρήστη. Άρα η ύπαρξη ενός χαρακτηριστικού στο προφίλ δεν μας δίνει από μόνο στοιχεία για το κατά πόσο αρέσει ή όχι το χαρακτηριστικό αυτό σε ένα χρήστη.

Έτσι, η απόσταση μεταξύ δύο προφίλ είναι απαραίτητο να εξαρτάται κυρίως από την διαφορά στα ποσοστά προτίμησης των κοινών χαρακτηριστικών. Ωστόσο θα πρέπει να δώσουμε κάποια ξεχωριστή σημασία και απόσταση στα χαρακτηριστικά τα οποία δεν είναι κοινά και στα χαρακτηριστικά τα οποία έχουν ποσοστό προτίμησης -1.Ο αλγόριθμος που ακολουθείτε για την εύρεση της απόστασης μεταξύ δύο προφίλ δίνετε πιο κάτω:

- Για να υπολογίσουμε το ποσοστό προτίμησης ενός στιγμιοτύπου πολλαπλασιάζουμε το ποσοστό προτίμησης για αυτό το χαρακτηριστικό με το ποσοστό προτίμησης της υποκατηγορίας στην οποία ανήκει. Ομοίως χειριζόμαστε και τα ποσοστά προτίμησης των υποκατηγοριών.

- Για τα κοινά στιγμιότυπα, υποκατηγορίες και χαρακτηριστικά που έχουν θετικό ποσοστό προτίμησης:

$Distance(x_{1,\dots,n}, y_{1,\dots,n}) = distance(x_{1,\dots,n-1}, y_{1,\dots,n-1}) + difference(x_n, y_n) * difference(x_n, y_n)$

$Difference(x_n, y_n) = norm(x_n) - norm(y_n)$

$Norm(x_n) = (x_n - Min)/(Max - Min)$

όπου n ο αριθμός στιγμιοτύπων μιας κατηγορίας, ή ο αριθμός υποκατηγοριών μιας κατηγορίας.

- Για τα στιγμιότυπα, κατηγορίες ή υποκατηγορίες που δεν είναι κοινά ή έχουν ο ένας από τους δύο έχει στο συγκεκριμένο χαρακτηριστικό, ποσοστό προτίμησης -1, τότε:

*Distance($x_{1,\dots,n}$,$y_{1,\dots,n}$)=distance($x_{1,\dots,n-1}$,$y_{1,\dots,n-1}$)+difference(x_n,y_n)*difference(x_n,y_n)*
Difference(x_n,y_n) = norm(x_n), όπου x_n το υπάρχον χαρακτηριστικό
Επιπλέον εάν Difference(x_n,y_n)<0,5 → Difference(x_n,y_n)= 1- Difference(x_n,y_n)

- Για τα χαρακτηριστικά που έχουν ποσοστό προτίμησης ίσο με -1 αλλά υπάρχουν μόνο στο ένα προφίλ δίνουμε απόσταση ίση με 1

- Η τελική απόσταση είναι το άθροισμα των τριών αυτών αποστάσεων (Στιγμιοτύπων, υποκατηγορίας και κατηγορίας). Ωστόσο η βαρύτητα που δίνουμε στην απόσταση μεταξύ στιγμιοτύπων είναι και πάλι μεγαλύτερη από τις άλλες δύο καθότι είναι και πιο σημαντική. Έτσι η τελική απόσταση δίνεται από την πιο κάτω εξίσωση:

*Distance = a*InstacesDistance + b*SubcategoryDistance+c*CategoryDistance*
Όπου a>b,c

Παράδειγμα:

Υπηρεσία 1
<poultry - 70>
<chicken - 90>
<duck - 49>
</poultry>

Υπηρεσία 2
<poultry- 45>
<chicken - 90>
<rabbit - -1>
</poultry>

Poultry	70	Poultry	45
poultry/chicken	90*0.7= 63	poultry/chicken	90*0.45=40,5
poultry/duck	49*0.7=34,3	poultry/rabbit	1

Diff (poultry/chicken)=0,63-0,405=0,225
Diff(poultry/duck)=0,343→1-0,343=0,657
Diff(poultry/rabbit)=1
Distance in Instances = $(0,225)^2 + (0,657)^2 + 1^2 = 1,481$

Diff(poultry)=0,7-0,45=0,25
Distance in SubCategories = $(0,25)^2$
Distance in Categories = 0
Total Distance = $\frac{3}{5} * 1,481 + \frac{1}{5} 0,625 + \frac{1}{5} 0 = 0,9$

7.3.1 Εύρεση των κοινών χρονικών περιόδων

Για την έυρεση των κοινών χρονικών περιόδων στο κοινό προφίλ

7.4 Μετρικές

Η αξιολόγηση το συστήματος είναι από τα πιο σημαντικά μέρη της εργασίας. Για να γίνει αυτό σωστά χρειαζόμαστε μετρικές που θα μας βοηθήσουν να βρούμε την αποτελεσματικότητα του συστήματος σε διάφορες διαστάσεις όπως ακρίβεια, ποιότητα και απόδοση. Συνήθως για την αποτίμηση των διαστάσεων αυτών χρησιμοποιούνται στατιστικά μέτρα. Στη συνέχει παρουσιάζουμε μερικές μετρικές που υπάρχουν για την αξιολόγηση συστημάτων εξατομίκευσης [37] αλλά και δύο δικές μας μετρικές που προσαρμόζονται στο πρόβλημα μας.

7.4.1 Mean Absolute Error

Με την μετρική αυτή υπολογίζουμε το μέσο σφάλμα ανάμεσα στη πρόβλεψη και την επιλογή του χρήστη. Πιο συγκεκριμένα, στη μετρική αυτή συγκρίνουμε τη θέση στην οποία εμφάνιση το σύστημα την τελική επιλογή του χρήστη και βρίσκουμε το σφάλμα από την πιο κάτω εξίσωση:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |f_i - y_i| = \frac{1}{n}\sum_{i=1}^{n} |e_i| \,.$$

Όπου f_i είναι η πρόβλεψη και y_i η πραγματική τιμή. Στο σύστημα που αναπτύξαμε προβλέψιμη είναι η θέση στην οποία εμφανίζεται στο χρήστη η επιθυμητή υπηρεσία. Πραγματική τιμή θεωρούνται, με βάση τη μετρική επιτυχίας που περιγράφετε πιο κάτω στο υποκεφάλαιο 7.4.3, οι πρώτες 5 θέσεις. Έτσι η μετρική αυτή μπορεί να υπολογιστεί και για τις πέντε πραγματικές τιμές που ορίσαμε.

Με το τρόπο αυτό μπορούμε να βρούμε πόσο μακριά βρίσκεται το σύστημα από το να δώσει τα επιθυμητά αποτελέσματα στο χρήστη. Η μετρική αυτή χρησιμοποιείται στο υποκεφάλαιο 8.2 για τα τελικά αποτελέσματα των αλγορίθμων.

7.4.2 Μέτρο Αποτελεσματικότητας

Η μετρική αυτή είναι μια μετρική ποσοτικής αξιολόγησης του συστήματος. Μας δίνει το ποσοστό των επιτυχημένων clicks. Με τον όρο «επιτυχημένα clicks» εννοούμε τα επιθυμητά από το χρήστη clicks τα οποία εμφάνισε το σύστημα.

Ωστόσο επειδή το σύστημα εμφανίζει σχεδόν όλες τις υπηρεσίες και στιγμιότυπα, προσθέτουμε ένα αρνητικό βάρος σε κάθε αποτέλεσμα. Το βάρος αυτό είναι αντιστρόφως ανάλογο της θέσης στην οποία εμφανίστηκε το αποτέλεσμα.

$$ER = \sum_{i=1}^{n} \frac{\sum_{j=1}^{m} \frac{r_{ij}}{j}}{n}$$

Όπου:

n: Ο αριθμός των υπηρεσιών στο σύστημα ή των στιγμιοτύπων υπηρεσίας

m: Η θέση στην οποία εμφανίστηκε η υπηρεσία ή το στιγμιότυπο

r: Το στιγμιότυπο ή η υπηρεσία που εμφανίστηκε στην θέση j

Το ποσοστό αυτό είναι το άθροισμα του αθροίσματος του αριθμού των στιγμιοτύπων ή υπηρεσιών που εμφανίστηκαν στην ίδια θέση, δια την θέση αυτή, και δια το συνολικό αριθμό στιγμιοτύπων ή υπηρεσιών.

Παράδειγμα:

Έστω ότι έχουμε 1000 clicks για επιλογή υπηρεσίας τα οποία εμφανίστηκαν στις πιο κάτω θέσεις:

Θέση Εμφάνισης	*Αριθμός Αποτελεσμάτων (Σενάριο 1)*	*Αριθμός Αποτελεσμάτων (Σενάριο 2)*
1	*100*	*400*
2	*200*	*300*
3	*300*	*200*
4	*400*	*100*

Σενάριο 1: $ER = \frac{\frac{100}{1} + \frac{200}{2} + \frac{300}{3} + \frac{400}{4}}{1000} = \frac{400}{1000} = 40\%$

Σενάριο 2: $ER = \frac{\frac{400}{1} + \frac{300}{2} + \frac{200}{3} + \frac{100}{4}}{1000} = \frac{641}{1000} = 64\%$

Η μετρική αυτή χρησιμοποιείται στα συγκεντρωτικά αποτελέσματα που παρουσιάζονται στο υποκεφάλαιο 8.2.

7.4.3 Μέτρο Επιτυχίας Αλγορίθμου (Ποσοτική ποιότητα βαθμολόγησης από το σύστημα)

Η μετρική αυτή είναι μια μετρική ποιοτικής αξιολόγησης του συστήματος. Ένα αποτέλεσμα θεωρείται επιτυχές εάν έχει εμφανιστεί μέσα στις N πρώτες θέσεις. Εάν ένας χρήστης επιλέξει το v-ιοστό αποτέλεσμα, αυτό θεωρείται επιτυχές εάν έχει εμφανιστεί στη θέση v < N. Το v είναι ο *πραγματικός όρος επιτυχίας* και το N ο *επιθυμητός όρος επιτυχίας*. Όσο πιο μικρός είναι ο επιθυμητός όρος επιτυχίας τόσο πιο μεγάλες προσδοκίες έχουμε από το σύστημα.

Στις μετρήσεις που ακολουθούν ο επιθυμητός όρος επιτυχίας είναι για τις υπηρεσίες 2 και για τα στιγμιότυπα 5. Το N =5 είναι ένα λογικό μέτρο σύγκρισης εάν λάβουμε υπόψη ότι αναφερόμαστε σε χρήστες του κινητού δικτύου οι οποίοι και χρησιμοποιούν κινητές συσκευές. Με μία πολύ εύκολη έρευνα μπορούμε εύκολα να εξάγουμε το συμπέρασμα ότι και η πιο απλή κινητή συσκευή μπορεί να εμφανίσει 5 γραμμές στην οθόνη της. Λαμβάνοντας λοιπόν υπόψη το πιο πάνω το N=5 μας δίνει τα αποτελέσματα τα οποία θα εμφανιστούν στην οθόνη χωρίς να αναγκάσουν το χρήστη να κάνει scrolling.

Η πιο πάνω μετρική λαμβάνεται υπόψη στην εμφάνιση των αποτελεσμάτων στο υποκεφάλαιο 8.2.1. Επιπλέον η μετρική αυτή είναι το μέτρο σύγκρισης που παρουσιάζεται στις γραφικές που ακολουθούν στο κεφάλαιο 8. Η θέση ταξινόμησης (Order Position) μας δίνει τη θέση στην οποία κατατάχθηκε η επιθυμητή από τον χρήστη υπηρεσία. Σκοπός είναι η θέση αυτή να βρίσκεται μέσα στις πρώτες 2 ή 5 ανάλογα αν τα αποτελέσματα αναφέρονται σε αποτελέσματα υπηρεσίας ή στιγμιοτύπων.

7.5 Σενάρια Ελέγχου

Επιπλέον για τον καλύτερο εντοπισμό της αποτελεσματικότητας του συστήματος θα πρέπει να ελέγξουμε πως συμπεριφέρεται το σύστημα σε διάφορα σενάρια όπως:

- Με personalization μόνο
- Με experience μόνο
- Με timing μόνο
- Με κανένα από τα τρία
- Με timing και experience
- Με timing και personalization
- Με personalization και timing

- Με personalization, timing και experience

Στην όλη διαδικασία σημαντικό ρόλο έχει ο τρόπος που επιλέγουμε το επιθυμητό για τον χρήστη σενάριο. Η επιλογή αυτή μπορεί να γίνει με βάση κάποια χαρακτηριστικά στο προφίλ του σε συνδυασμό της τυχαίας επιλογής. Επιπλέον η επιλογή αυτή μπορεί να γίνει και τελείως τυχαία. Στη περίπτωση αυτή έχουμε το χειρότερο σενάριο και μπορούμε να κάνουμε εύκολα συγκρίσεις και να δούμε κατά πόσο η όλη υλοποίηση μας αποφέρει ικανοποιητικά αποτελέσματα.

Εκτενέστερη περιγραφή των πιο πάνω σενάριων καθώς και των αποτελεσμάτων τους γίνεται στο επόμενο κεφάλαιο.

Κεφάλαιο 8

Αποτελέσματα και Συμπεράσματα.

8.1 Αποτελέσματα

Στην ενότητα αυτή θα παρουσιαστούν και θα σχολιαστούν τα αποτελέσματα που έδωσε το σύστημα ακολουθώντας τη διαδικασία ελέγχου όπως περιγράφτηκε στο προηγούμενο κεφάλαιο.

8.1.1 Αποτελέσματα Αλγορίθμων ενημέρωσης προφίλ (Σενάριο 1)

Στο σενάριο αυτό λαμβάνοντας υπόψη τα αρχικά προφίλ του κάθε χρήστη, δημιουργούμε το επιθυμητό clickstream. Στη συνέχεια το σύστημα τρέχει δημιουργώντας ένα σημαντικό clickstream για το χρήστη (1500 εγγραφές). Με βάση το clickstream αυτό ενημερώνεται τελικά το προφίλ του χρήστη. Η πιο πάνω διαδικασία επαναλαμβάνεται 3 φορές. Σε κάθε εκτέλεση των αλγορίθμων που ενημερώνουν τα προφίλ παίρνουμε μετρήσεις. Στις μετρήσεις αυτές μετρούμε:

1. Την θέση στην οποία εμφανίστηκε η επιθυμητή υπηρεσία για τον χρήστη. (Σύνολο από 7 διαφορετικές υπηρεσίες)

2. Την θέση στην οποία εμφανίστηκε το επιθυμητό στιγμιότυπο μιας υπηρεσίας για τον χρήστη. (Σύνολο από 50 διαφορετικά στιγμιότυπα για κάθε υπηρεσία)

Ακολουθούν τα αποτελέσματα του αλγορίθμου αυτού για κάθε αλγόριθμο ενημέρωσης προφίλ που χρησιμοποιήθηκε.

8.1.1.1 Cluster Clickstream Update Algorithm

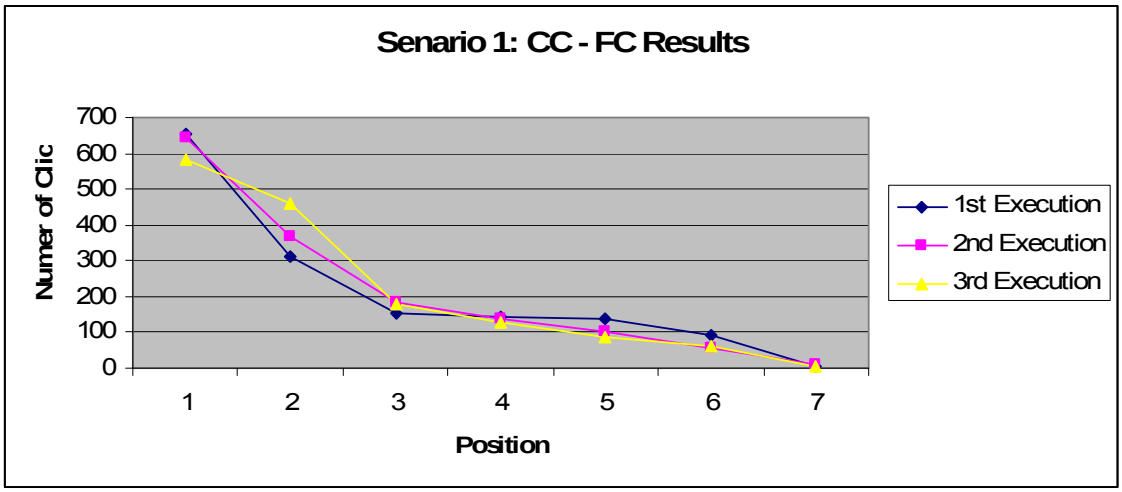
Στους πίνακες που ακολουθούν παρουσιάζονται τα αποτελέσματα του συστήματος για κάθε μια από τις τρεις εκτελέσεις του κάθε αλγορίθμου. Εδώ παρουσιάζονται οι τρεις παραλλαγές του αλγορίθμου Cluster Clickstream Update, με βάση τον αντίστοιχο αλγόριθμο για χρονικές περιόδους που χρησιμοποιεί. Στο πίνακα φαίνονται οι θέσεις που εμφανίστηκαν στο σύστημα τα επιθυμητά από τον χρήστη στιγμιότυπα ή υπηρεσίες σε σχέση με κάθε εκτέλεση και κάθε παραλλαγή του αλγορίθμου.

Αποτελέσματα στην Επιλογή Υπηρεσίας

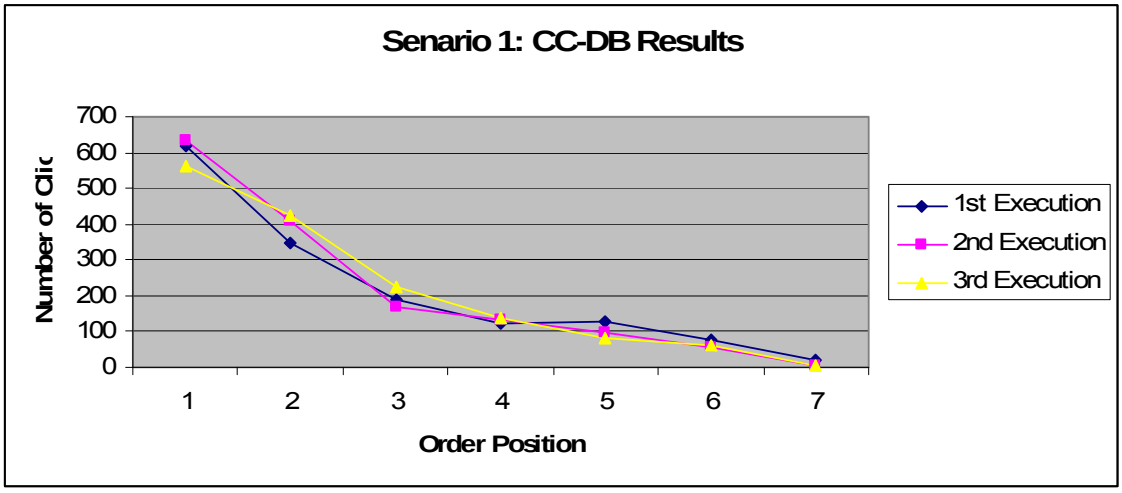
Πιο κάτω παρουσιάζονται τα αποτελέσματα στην επιλογή υπηρεσίας κατά τις τρεις εκτελέσεις των αλγορίθμων. Στο σύστημα υπάρχουν 7 υπηρεσίες και κατ' επέκταση 7 πιθανές θέσεις για να παρουσιαστούν. Στη γραφική παράσταση που ακολουθεί παρουσιάζονται πιο συνοπτικά τα αποτελέσματα του συγκεκριμένου αλγορίθμου σε σχέση με την θέση ταξινόμησης (order position) μιας υπηρεσίας στο σύστημα. Τα αποτελέσματα συγκρίνονται με βάση την μετρική επιτυχίας που περιγράψαμε στο υποκεφάλαιο 7.4.3.

Θέση Εμφάνισης	**Flat Clickstream Time Zones Update Algorithm**			**Density Based Time Zones Update Algorithm**			**Histogram Time Zones Update Algorithm**		
	Εκτελέσεις								
	1η	**2η**	**3η**	**1η**	**2η**	**3η**	**1η**	**2η**	**3η**
1	652	646	584	618	633	562	691	683	603
2	310	368	458	345	410	423	323	369	412
3	155	182	181	188	169	227	171	163	214
4	145	136	126	125	131	137	112	120	119
5	139	104	87	128	95	81	122	97	82
6	92	55	60	75	56	63	69	64	65
7	7	9	4	21	6	7	12	4	5

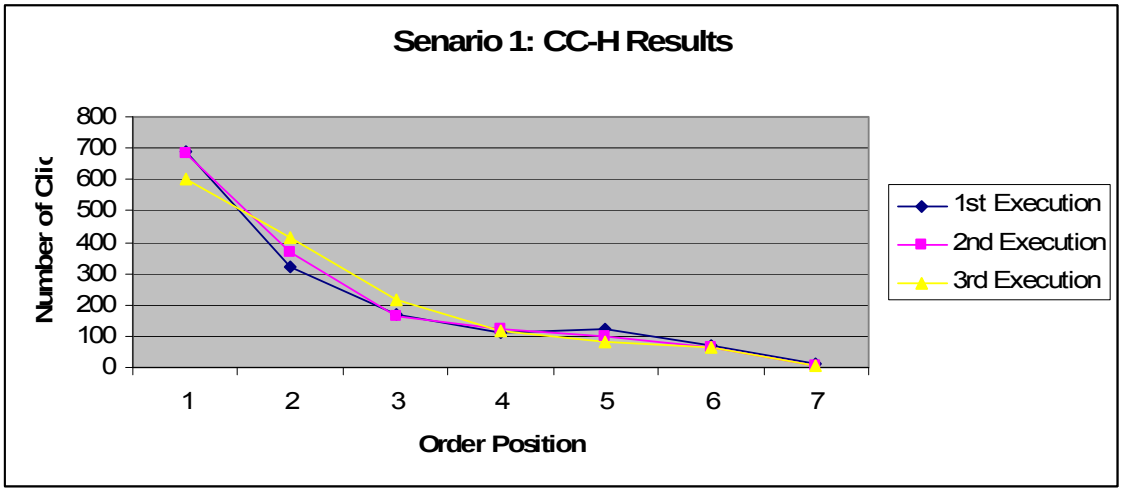
Flat Clickstream Time Zones Update Algorithm



Density Based Time Zones Update Algorithm



Histogram Time Zones Update Algorithm

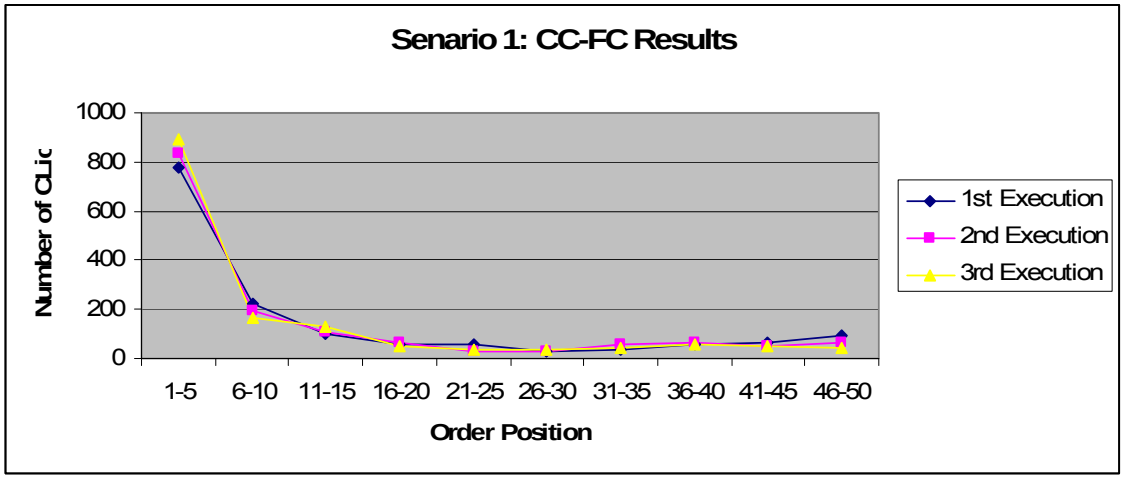


Αποτελέσματα στην Επιλογή Στιγμιοτύπου Υπηρεσίας

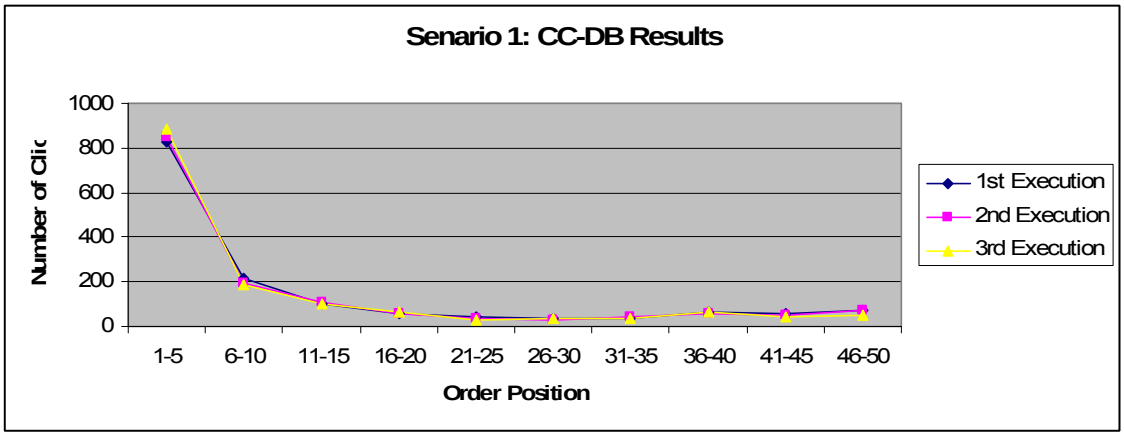
Πιο κάτω παρουσιάζονται τα αποτελέσματα στην επιλογή στιγμιοτύπου κατά τις τρεις εκτελέσεις των αλγορίθμων. Στο σύστημα για κάθε υπηρεσία υπάρχουν 50 στιγμιότυπα και κατ' επέκταση 50 πιθανές θέσεις για να παρουσιαστούν. Στη γραφική παράσταση που ακολουθεί παρουσιάζονται πιο συνοπτικά τα αποτελέσματα του συγκεκριμένου αλγορίθμου σε σχέση με την θέση ταξινόμησης (order position)του επιθυμητού στιγμιοτύπου στο σύστημα. Τα αποτελέσματα συγκρίνονται με βάση την μετρική επιτυχίας που περιγράψαμε στο υποκεφάλαιο 7.4.3.

Θέση Εμφάνισης	**Flat Clickstream Time Zones Update Algorithm**			**Density Based Time Zones Update Algorithm**			**Histogram Time Zones Update Algorithm**		
	Εκτελέσεις								
	1η	**2η**	**3η**	**1η**	**2η**	**3η**	**1η**	**2η**	**3η**
1-5	776	834	889	825	852	885	799	824	894
6-10	221	192	167	215	195	188	213	183	191
11-15	102	111	128	99	106	101	104	123	101
16-20	61	63	50	54	57	67	52	59	65
21-25	55	32	37	40	36	30	40	40	35
26-30	29	31	36	33	31	34	35	40	26
31-35	38	57	45	37	41	36	44	62	35
36-40	61	64	54	68	59	67	65	74	70
41-45	63	50	51	57	52	40	63	44	38
46-50	94	66	43	72	71	52	85	51	45

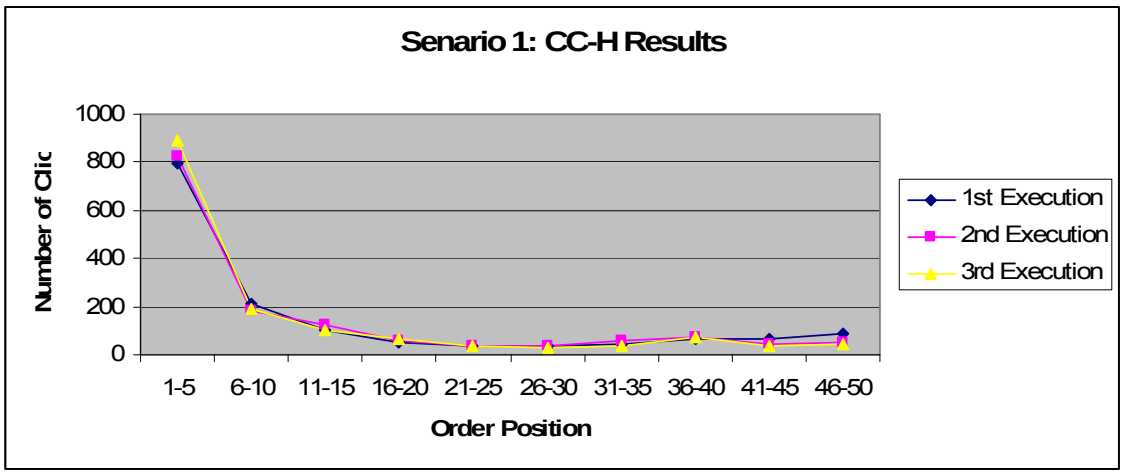
Flat Clickstream Time Zones Update Algorithm



Density Based Time Zones Update Algorithm



Histogram Time Zones Update Algorithm



8.1.1.2 Moving Average Clickstream Update Algorithm

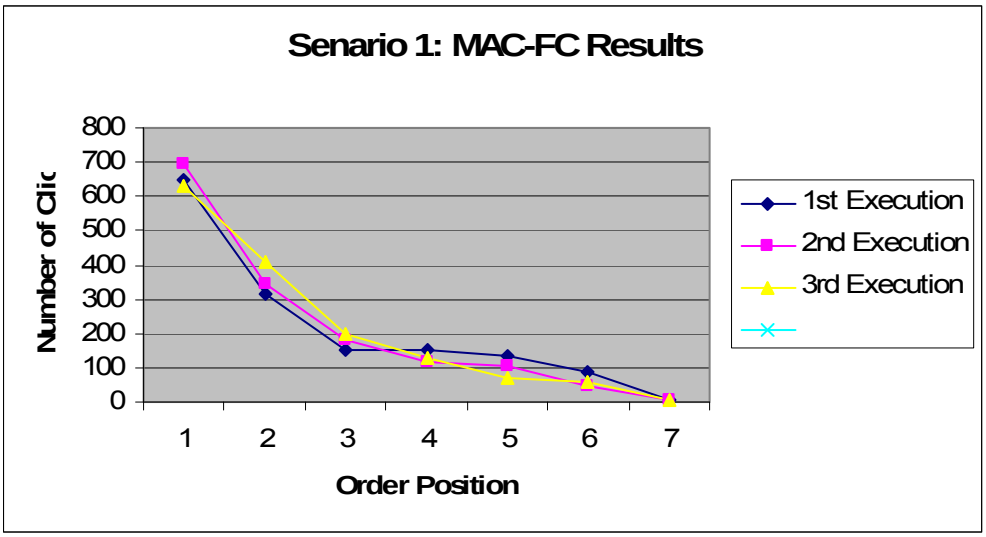
Στους πίνακες που ακολουθούν παρουσιάζονται τα αποτελέσματα του συστήματος για κάθε μια από τις τρεις εκτελέσεις του κάθε αλγορίθμου. Εδώ παρουσιάζονται οι τρεις παραλλαγές του αλγορίθμου Moving Average Clickstream Update, με βάση τον αντίστοιχο αλγόριθμο για χρονικές περιόδους που χρησιμοποιεί. Στο πίνακα φαίνονται οι θέσεις που εμφανίστηκαν στο σύστημα τα επιθυμητά από τον χρήστη στιγμιότυπα ή υπηρεσίες σε σχέση με κάθε εκτέλεση και κάθε παραλλαγή του αλγορίθμου.

Αποτελέσματα στην Επιλογή Υπηρεσίας

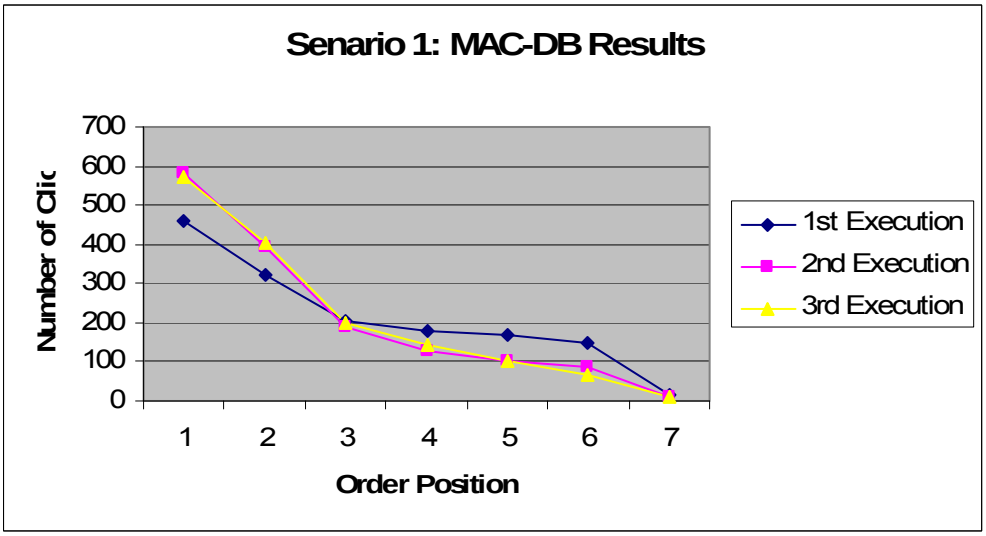
Πιο κάτω παρουσιάζονται τα αποτελέσματα στην επιλογή υπηρεσίας κατά τις τρεις εκτελέσεις των αλγορίθμων. Στο σύστημα υπάρχουν 7 υπηρεσίες και κατ' επέκταση 7 πιθανές θέσεις για να παρουσιαστούν. Στη γραφική παράσταση που ακολουθεί παρουσιάζονται πιο συνοπτικά τα αποτελέσματα του συγκεκριμένου αλγορίθμου σε σχέση με την θέση ταξινόμησης (order position) της επιθυμητής υπηρεσίας στο σύστημα. Τα αποτελέσματα συγκρίνονται με βάση την μετρική επιτυχίας που περιγράψαμε στο υποκεφάλαιο 7.4.3.

Θέση Εμφάνισης	**Flat Clickstream Time Zones Update Algorithm**			**Density Based Time Zones Update Algorithm**			**Histogram Time Zones Update Algorithm**		
	Εκτελέσεις								
	1η	**2η**	**3η**	**1η**	**2η**	**3η**	**1η**	**2η**	**3η**
1	651	697	630	460	585	574	686	656	663
2	314	346	409	323	394	405	294	407	407
3	151	181	197	205	190	201	161	168	198
4	152	118	129	180	128	143	129	119	106
5	137	106	71	169	104	102	137	83	72
6	88	45	59	146	89	65	84	62	47
7	7	7	5	17	10	10	9	5	7

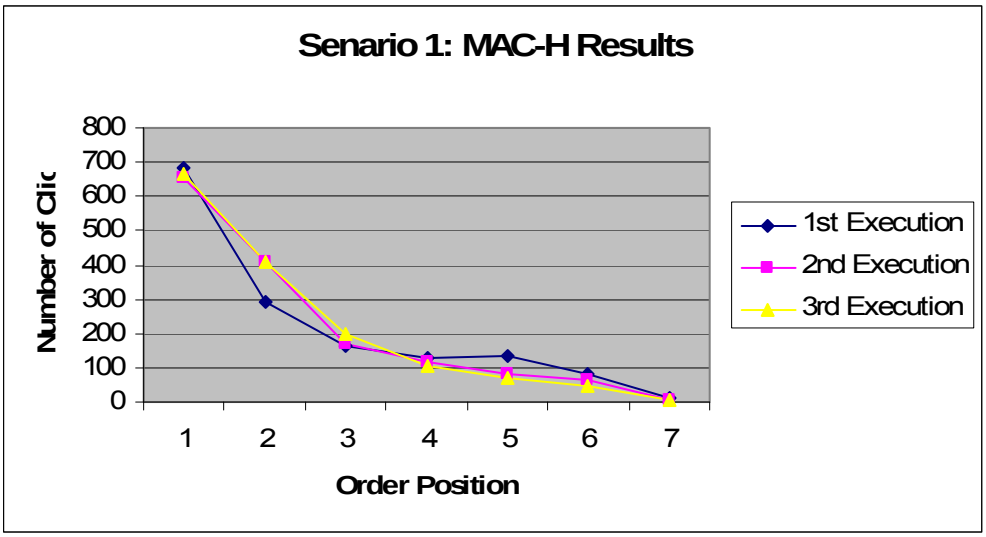
Flat Clickstream Time Zones Update Algorithm



Density Based Time Zones Update Algorithm



Histogram Time Zones Update Algorithm

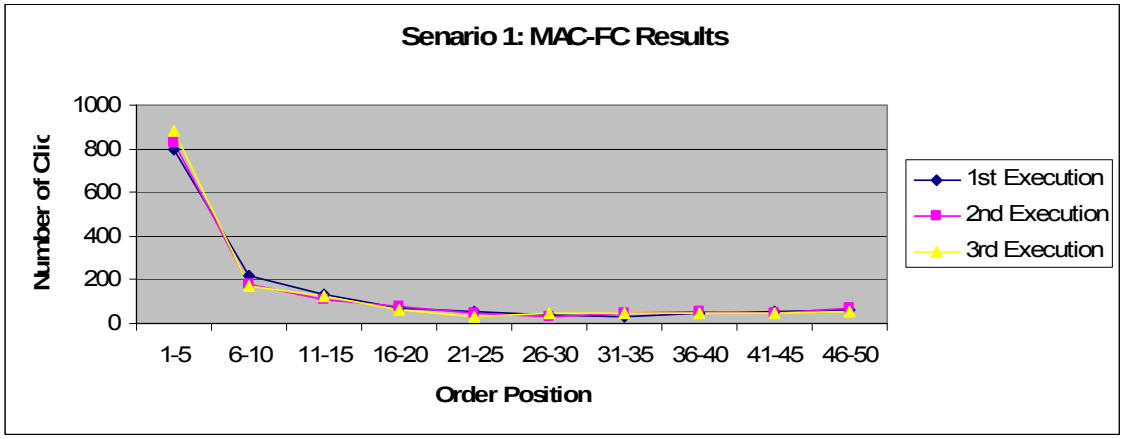


Αποτελέσματα στην Επιλογή Στιγμιοτύπου Υπηρεσίας

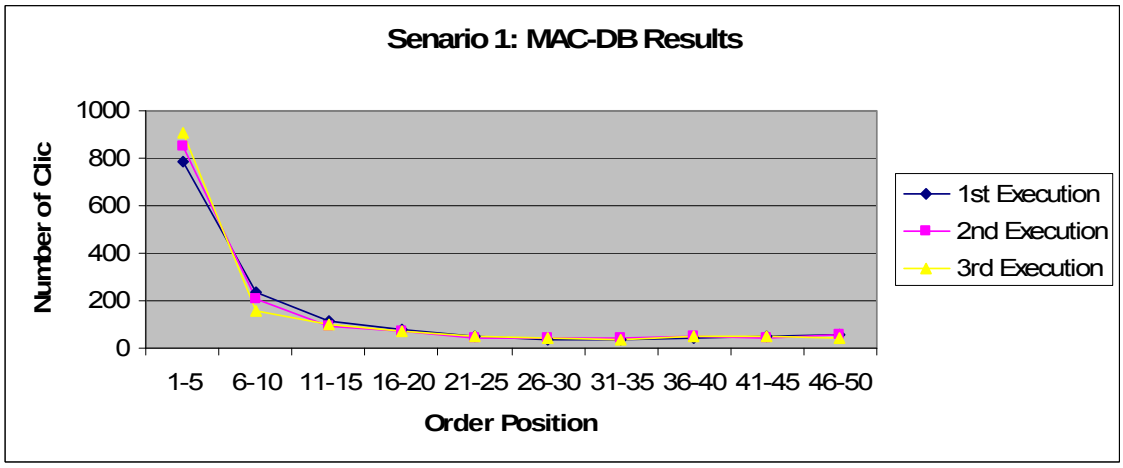
Πιο κάτω παρουσιάζονται τα αποτελέσματα στην επιλογή στιγμιοτύπου κατά τις τρεις εκτελέσεις των αλγορίθμων. Στο σύστημα για κάθε υπηρεσία υπάρχουν 50 στιγμιότυπα και κατ' επέκταση 50 πιθανές θέσεις για να παρουσιαστούν. Στη γραφική παράσταση που ακολουθεί παρουσιάζονται πιο συνοπτικά τα αποτελέσματα του συγκεκριμένου αλγορίθμου σε σχέση με την θέση ταξινόμησης (order position)του επιθυμητού στιγμιοτύπου στο σύστημα. Τα αποτελέσματα συγκρίνονται με βάση την μετρική επιτυχίας που περιγράψαμε στο υποκεφάλαιο 7.4.3.

Θέση Εμφάνισης	**Flat Clickstream Time Zones Update Algorithm**			**Density Based Time Zones Update Algorithm**			**Histogram Time Zones Update Algorithm**		
	Εκτελέσεις								
	1η	**2η**	**3η**	**1η**	**2η**	**3η**	**1η**	**2η**	**3η**
1-5	793	832	881	789	847	905	812	836	871
6-10	215	182	170	237	204	157	224	186	172
11-15	135	110	122	117	95	97	123	106	129
16-20	67	78	61	82	72	70	60	79	64
21-25	56	45	32	53	44	50	53	46	55
26-30	41	33	45	38	42	43	36	42	31
31-35	35	48	43	37	45	34	40	43	31
36-40	45	55	46	42	49	49	41	42	55
41-45	52	45	44	50	43	49	56	58	42
46-50	61	72	56	55	59	46	55	62	50

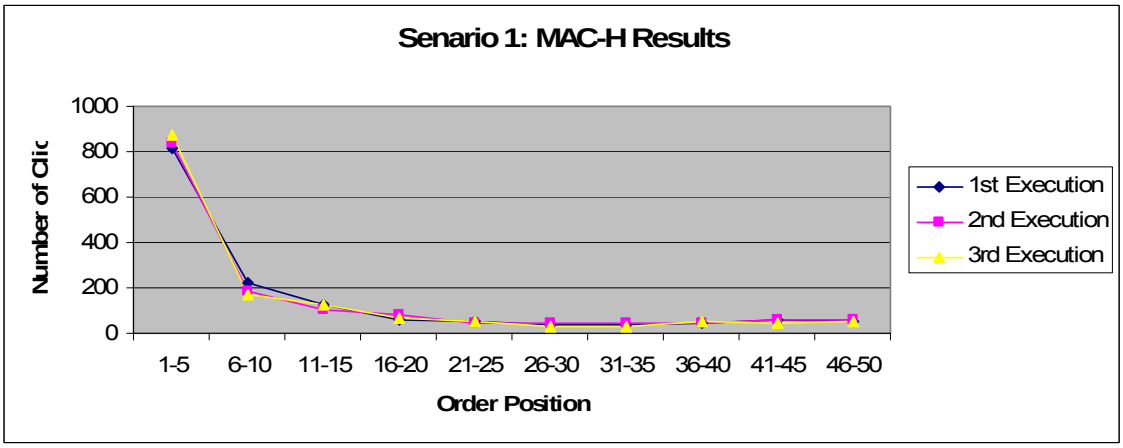
Flat Clickstream Time Zones Update Algorithm



Density Based Time Zones Update Algorithm



Histogram Time Zones Update Algorithm



8.1.1.3 Flat Clickstream Update Algorithm

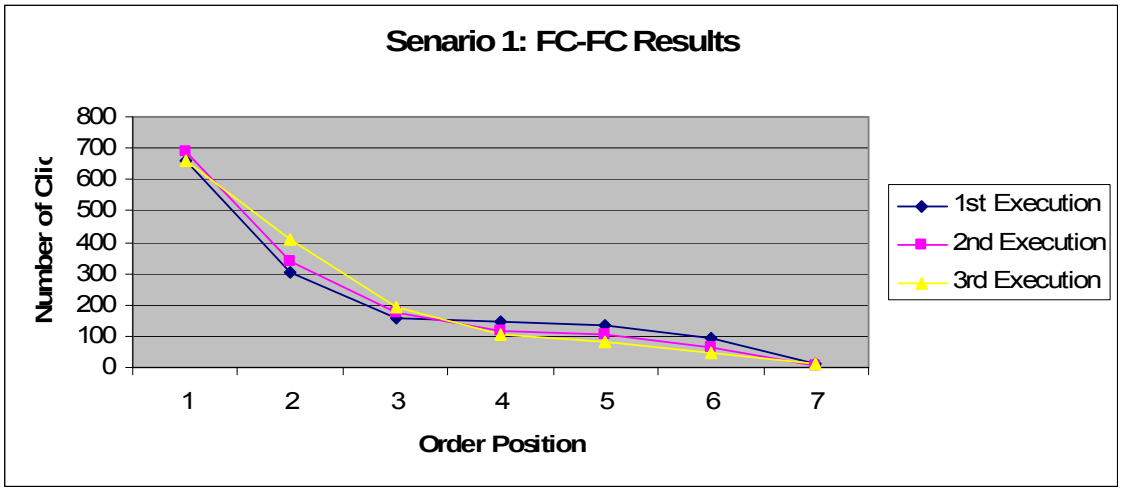
Στους πίνακες που ακολουθούν παρουσιάζονται τα αποτελέσματα του συστήματος για κάθε μια από τις τρεις εκτελέσεις του κάθε αλγορίθμου. Εδώ παρουσιάζονται οι τρεις παραλλαγές του αλγορίθμου Flat Clickstream Update, με βάση τον αντίστοιχο αλγόριθμο για χρονικές περιόδους που χρησιμοποιεί. Στο πίνακα φαίνονται οι θέσεις που εμφανίστηκαν στο σύστημα τα επιθυμητά από τον χρήστη στιγμιότυπα ή υπηρεσίες σε σχέση με κάθε εκτέλεση και κάθε παραλλαγή του αλγορίθμου.

Αποτελέσματα στην Επιλογή Υπηρεσίας

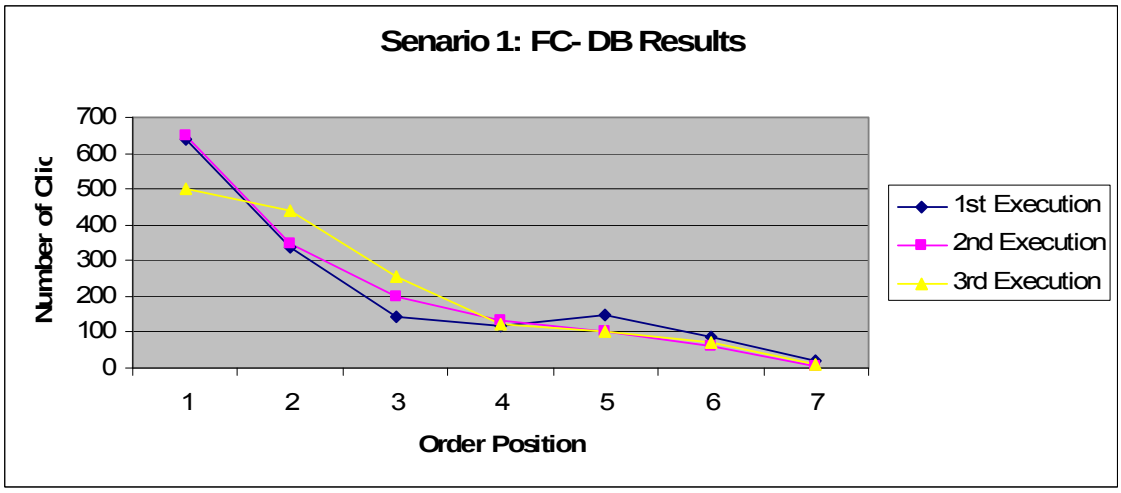
Πιο κάτω παρουσιάζονται τα αποτελέσματα στην επιλογή υπηρεσίας κατά τις τρεις εκτελέσεις των αλγορίθμων. Στο σύστημα υπάρχουν 7 υπηρεσίες και κατ' επέκταση 7 πιθανές θέσεις για να παρουσιαστούν. Στη γραφική παράσταση που ακολουθεί παρουσιάζονται πιο συνοπτικά τα αποτελέσματα του συγκεκριμένου αλγορίθμου σε σχέση με την θέση ταξινόμησης (order position) της επιθυμητής υπηρεσίας στο σύστημα. Τα αποτελέσματα συγκρίνονται με βάση την μετρική επιτυχίας που περιγράψαμε στο υποκεφάλαιο 7.4.3.

Θέση Εμφάνισης	**Flat Clickstream Time Zones Update Algorithm**			**Density Based Time Zones Update Algorithm**			**Histogram Time Zones Update Algorithm**		
	Εκτελέσεις								
	1η	**2η**	**3η**	**1η**	**2η**	**3η**	**1η**	**2η**	**3η**
1	657	689	660	641	649	501	673	678	619
2	306	337	408	338	349	438	309	385	474
3	158	177	194	145	198	255	160	147	167
4	144	117	103	119	134	124	124	123	108
5	135	107	81	148	101	102	137	106	87
6	91	65	45	87	62	71	90	55	41
7	9	8	9	22	7	9	7	6	4

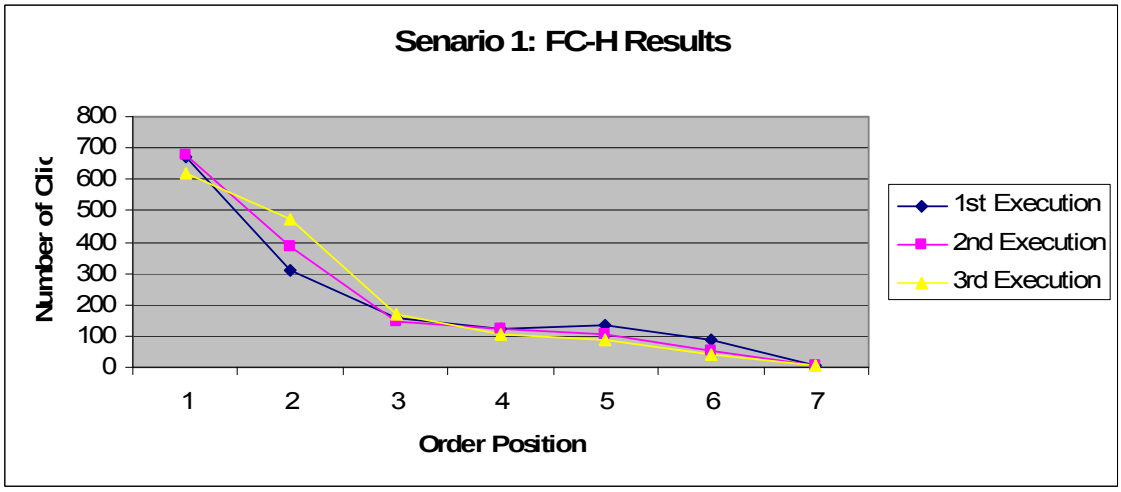
Flat Clickstream Time Zones Update Algorithm



Density Based Time Zones Update Algorithm



Histogram Time Zones Update Algorithm

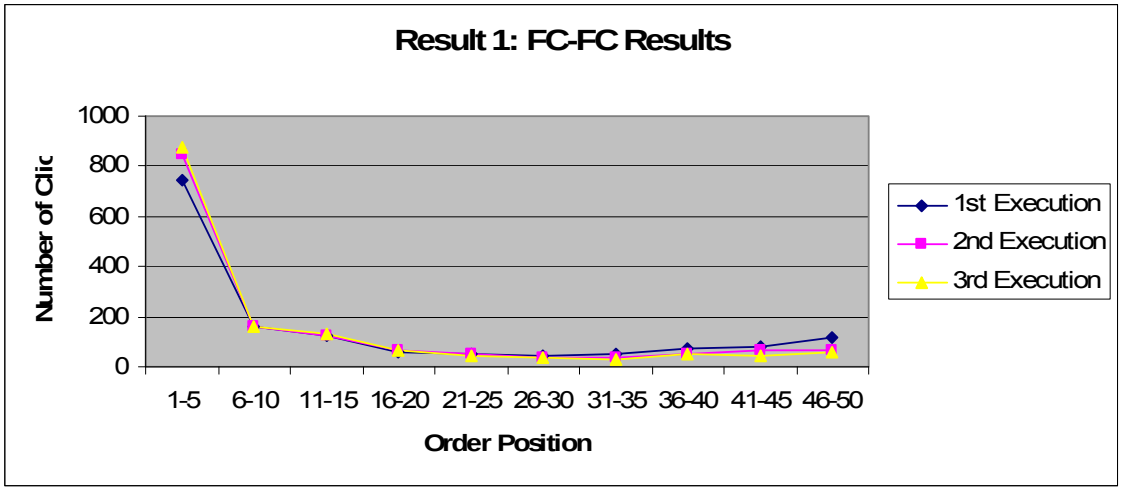


Αποτελέσματα στην Επιλογή Στιγμιοτύπου Υπηρεσίας

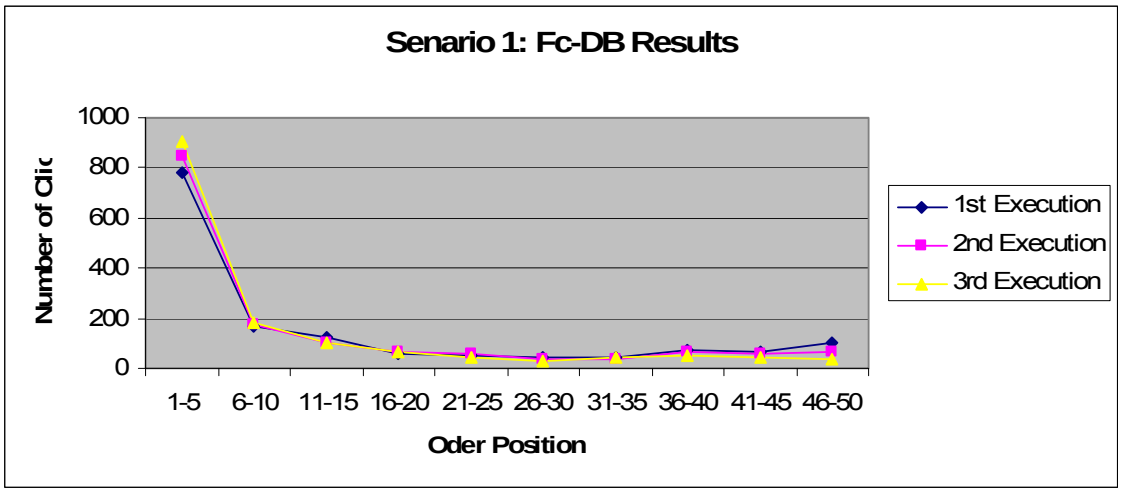
Πιο κάτω παρουσιάζονται τα αποτελέσματα στην επιλογή στιγμιοτύπου κατά τις τρεις εκτελέσεις των αλγορίθμων. Στο σύστημα για κάθε υπηρεσία υπάρχουν 50 στιγμιότυπα και κατ' επέκταση 50 πιθανές θέσεις για να παρουσιαστούν. Στη γραφική παράσταση που ακολουθεί παρουσιάζονται πιο συνοπτικά τα αποτελέσματα του συγκεκριμένου αλγορίθμου σε σχέση με την θέση ταξινόμησης (order position)του επιθυμητού στιγμιοτύπου στο σύστημα. Τα αποτελέσματα συγκρίνονται με βάση την μετρική επιτυχίας που περιγράψαμε στο υποκεφάλαιο 7.4.3.

Θέση Εμφάνισης	**Flat Clickstream Time Zones Update Algorithm**			**Density Based Time Zones Update Algorithm**			**Histogram Time Zones Update Algorithm**		
	Εκτελέσεις								
	1η	**2η**	**3η**	**1η**	**2η**	**3η**	**1η**	**2η**	**3η**
1-5	745	847	875	778	845	904	753	816	881
6-10	164	160	159	169	176	183	174	157	157
11-15	125	121	135	122	103	103	125	127	127
16-20	55	66	68	58	63	64	55	80	65
21-25	51	52	45	48	62	42	48	48	53
26-30	42	33	34	43	36	26	39	49	50
31-35	48	34	30	45	33	44	56	51	40
36-40	75	54	53	73	64	54	62	59	44
41-45	81	64	45	64	55	41	76	55	50
46-50	114	69	56	100	63	39	112	58	33

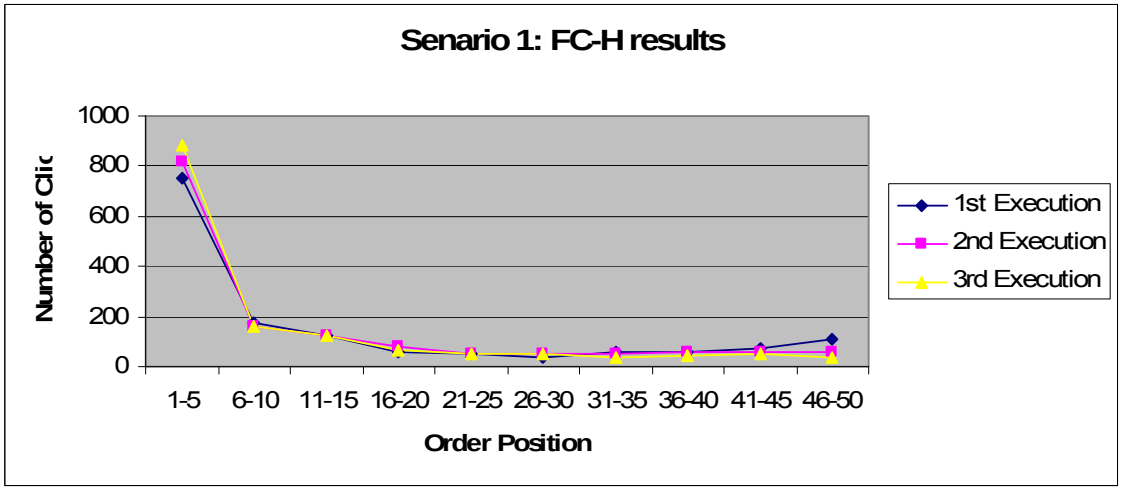
Flat Clickstream Time Zones Update Algorithm



Density Based Time Zones Update Algorithm



Histogram Time Zones Update Algorithm



8.1.1.4 Σύγκριση Αποτελεσμάτων

Επιλογή Υπηρεσίας

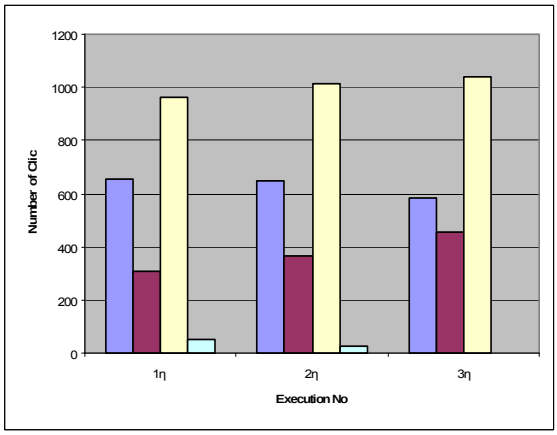
Για να συγκρίνουμε και να μπορέσουμε να βγάλουμε συμπεράσματα από τα αποτελέσματα στην επιλογή υπηρεσίας, εξετάζουμε τις δύο μόνο πρώτες θέσεις. Στις θέσεις αυτές δίνονται ο αριθμός αποτελεσμάτων που εμφανίστηκαν πρώτα και δεύτερα στο χρήστη για επιλογή υπηρεσίας.

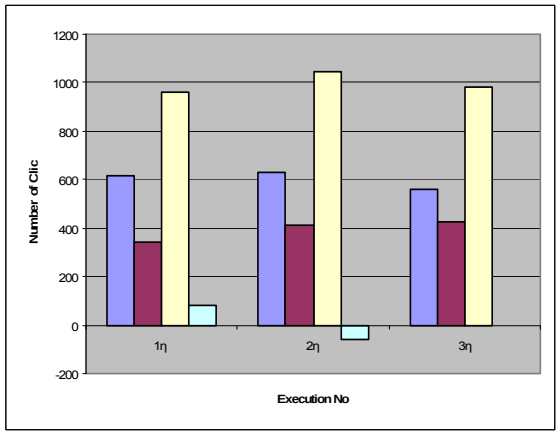
Πιο κάτω ακολουθούν ο συνοπτικός πίνακας με τα αποτελέσματα κάθε αλγορίθμου και των παραλλαγών του και για τις τρεις εκτελέσεις. Στη συνέχεια ακολουθούν οι αντίστοιχες γραφικές. Κάθε γραφική δείχνει τα αποτελέσματα που

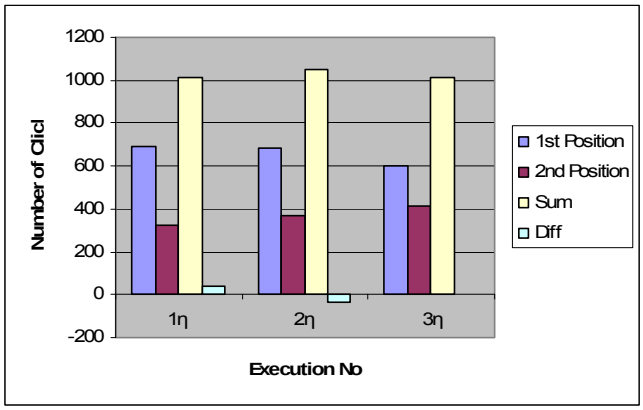
εμφανίστηκαν στην πρώτη και δεύτερη θέση, με βάση την εκτέλεση (1η, 2η ή 3η) του αλγορίθμου.

Θέση Εμφάνισης	**Flat Clickstream Time Zones Update Algorithm**			**Density Based Time Zones Update Algorithm**			**Histogram Time Zones Update Algorithm**		
	Εκτελέσεις								
	1η	**2η**	**3η**	**1η**	**2η**	**3η**	**1η**	**2η**	**3η**
	Cluster Clickstream Update Algorithm								
1	652	646	584	618	633	562	691	683	603
2	310	368	458	345	410	423	323	369	412
Sum	*962*	*1014*	*1042*	*963*	*1043*	*985*	*1014*	*1052*	*1015*
Diff	52	28		80	-58		38	-37	
	Moving Average Clickstream Update Algorithm								
1	651	697	630	460	585	574	686	656	663
2	314	346	409	323	394	405	294	407	407
Sum	*965*	*1043*	*1039*	*783*	*979*	*979*	*980*	*1063*	*1070*
Diff	78	-4		236	0		83	7	
	Flat Clickstream Update Algorithm								
1	657	689	660	641	649	501	673	678	619
2	306	337	408	338	349	438	309	385	474
Sum	*963*	*1026*	*1068*	*979*	*998*	*939*	*982*	*1063*	*1093*
Diff	63	42		19	-59		81	30	

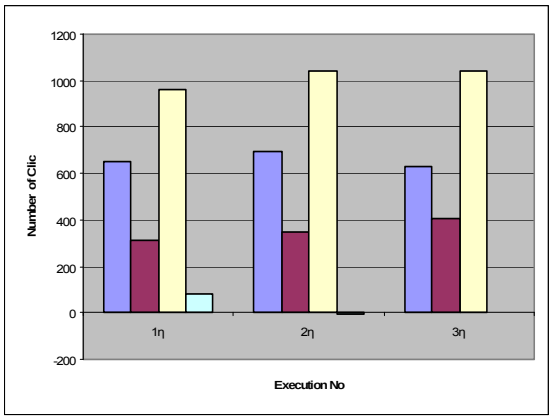
- **Senario 1: Results for CC-FC, CC-DB and CC-H**

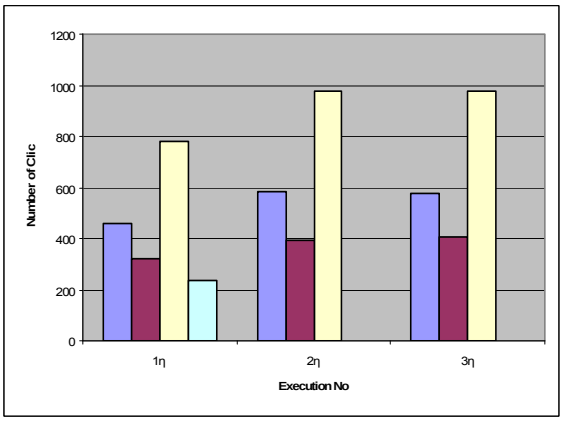


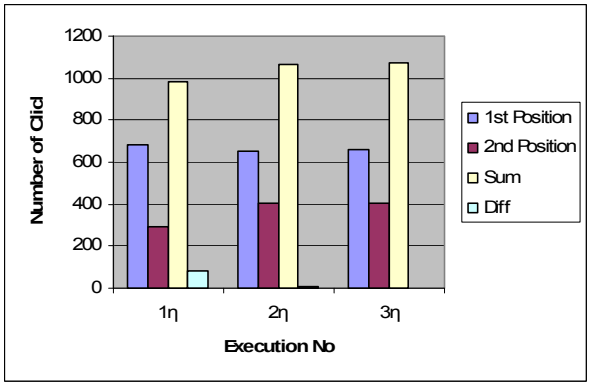




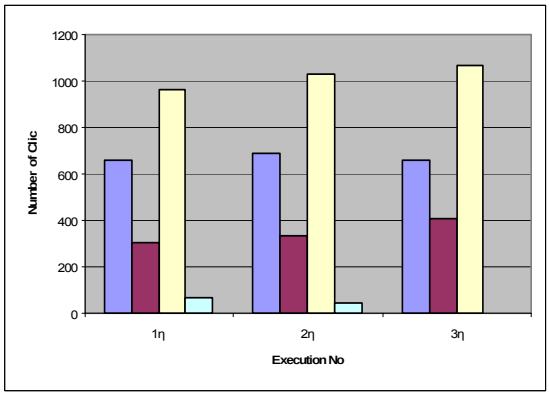
- **Senario 1: Results for MAC-FC, MAC-DB and MAC-H**

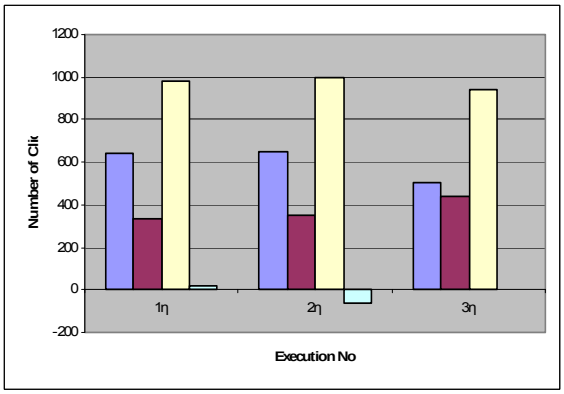


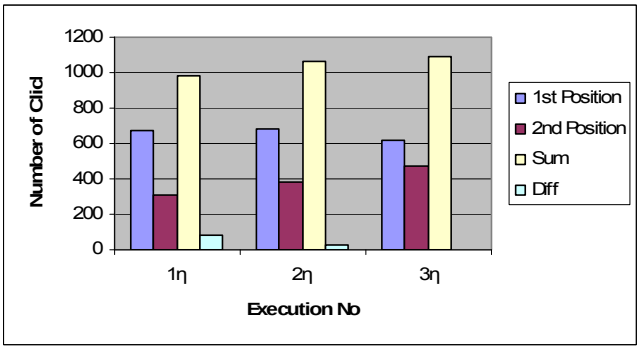




- **Senario 1: Results for FC-FC, FC-DB and FC-H**







Από τα πιο πάνω μπορούμε να εξάγουμε τα εξής:

- Αλγόριθμοι που τείνουν να διαχωρίζουν τη δραστηριότητα του χρήστη σε πολλαπλές χρονικές περιόδους με κατ' επέκταση μικρότερα ποσοστά προτίμησης σε αυτές, δεν δίνουν καλά αποτελέσματα. Ο αλγόριθμος Density Based χωρίζει τις χρονικές περιόδους σε 6 διαφορετικές περιόδους και ενώνει περιόδους με κοντινή συμπεριφορά. Ο Flat Clickstream χωρίζει τις δραστηριότητες του χρήστη σε τρεις χρονικές περιόδους και προσθέτει μόνο τις μηδενικές περιόδους μεταξύ τους. Από το πιο πάνω πίνακα παρατηρούμε ότι η απόδοση του Density Based ο οποίο χωρίζει σε πιο πολλά χρονικά διαστήματα το 24ωρο ενός χρήστη, μας δίνει τα λιγότερο ικανοποιητικά αποτελέσματα.

- Ο αλγόριθμος Histogram ο οποίος χρησιμοποιεί τη μέθοδο της τυπική απόκληση φαίνεται να έχει την καλύτερη επίδοση για τους αλγορίθμους Moving Average και Flat Clickstream. Ωστόσο η απόδοση του αλγορίθμου αυτού εξαρτάται σημαντικά από το κατώφλι το οποίο χρησιμοποιούμε κατά την σύγκριση των τυπικών αποκλίσεων δύο γειτονικών χρονικών περιόδων. Υπάρχει πάντα η πιθανότητα να απορριφθεί η ομαδοποίηση δύο γειτονικών χρονικών περιόδων για πολύ μικρή διαφορά κάτι το οποίο θα επιδράσει αρνητικά στα αποτελέσματα. Ο αλγόριθμος αυτός φαίνεται να είναι πάρα πολύ καλός ωστόσο είναι αρκετά ευαίσθητος και απαιτείται η προσεκτική επιλογή κατωφλιού για να μπορέσει να δώσει ικανοποιητικά αποτελέσματα. Ο καλύτερος τρόπος για την επιλογή αυτή, είναι ελέγχοντας την απόδοση του αλγόριθμου με διαφορετικές τιμές στο κατώφλι αυτό.

Επιλογή Στιγμιότυπου Υπηρεσίας

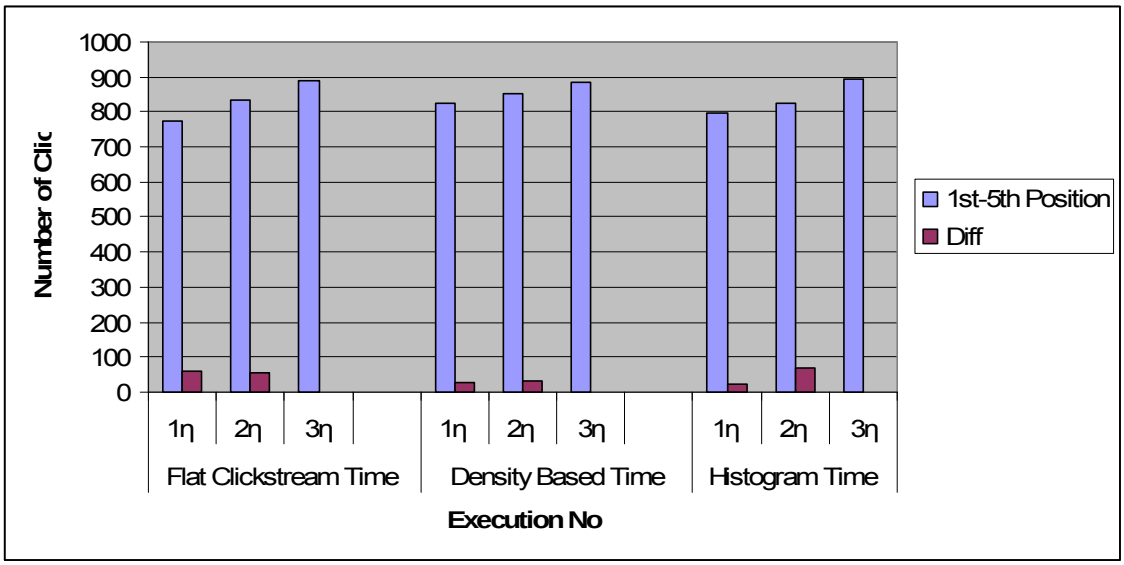
Για την μελέτη των αποτελεσμάτων στα στιγμιότυπα μιας υπηρεσίας, εστιαζόμαστε στην μελέτη των 5 πρώτων θέσεων με βάση το κριτήριο Ποσοτικής Ποιότητας Βαθμολόγησης.

Πιο κάτω ακολουθούν ο συνοπτικός πίνακας με τα αποτελέσματα κάθε αλγορίθμου και των παραλλαγών του και για τις τρεις εκτελέσεις. Στη συνέχεια ακολουθούν οι αντίστοιχες γραφικές. Κάθε γραφική δείχνει τα αποτελέσματα που

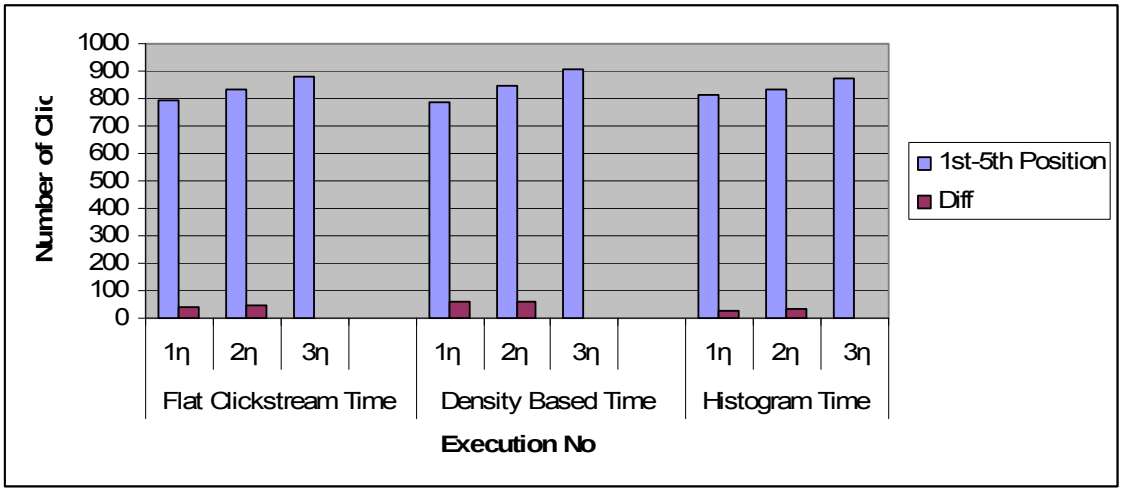
εμφανίστηκαν στις πρώτες 5 θέσεις, με βάση την εκτέλεση (1η, 2η ή 3η) του αλγορίθμου και την παραλλαγή του.

Θέση Εμφάνισης	Flat Clickstream Time Zones Update Algorithm			Density Based Time Zones Update Algorithm			Histogram Time Zones Update Algorithm		
	Εκτελέσεις								
	1η	2η	3η	1η	2η	3η	1η	2η	3η
	Cluster Clickstream Update Algorithm								
1-5	776	834	889	825	852	885	799	824	894
Diff	*58*	*55*		*27*	*33*		*25*	*70*	
	Moving Average Clickstream Update Algorithm								
1-5	793	832	881	789	847	905	812	836	871
Diff	*39*	*49*		*58*	*58*		*24*	*35*	
	Flat Clickstream Update Algorithm								
1-5	745	847	875	778	845	904	753	816	881
Diff	*102*	*28*		*67*	*59*		*63*	*65*	

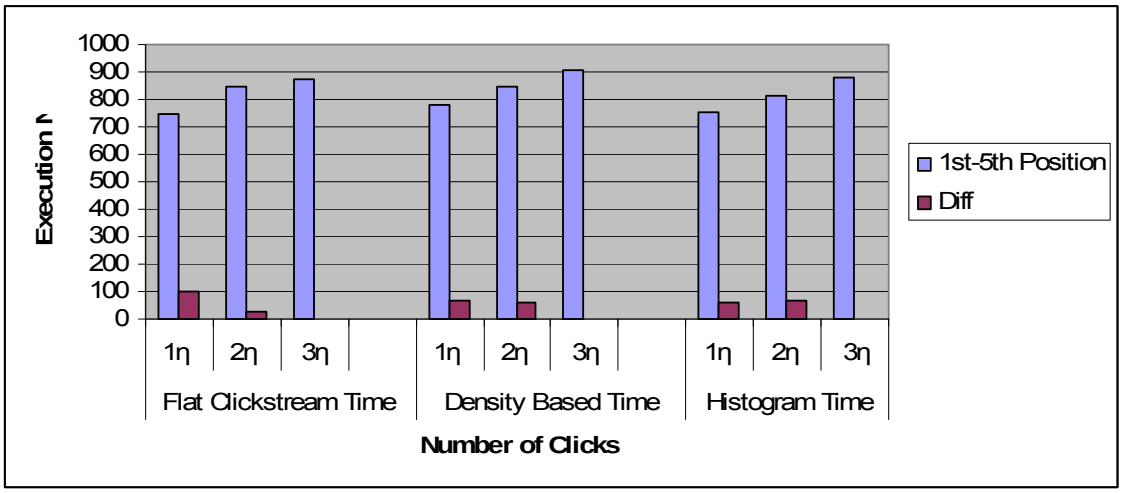
- **Cluster Clickstream Update Algorithm**



- **Moving Average Clickstream Update Algorithm**



- **Flat Clickstream Update Algorithm**



Από τα πιο πάνω αποτελέσματα μπορούμε να εξάγουμε τα πιο κάτω συμπεράσματα:

- Ο αλγόριθμος Density Based για τις χρονικές περιόδους έδωσε πολύ καλύτερα αποτελέσματα από τους άλλους δύο αλγορίθμους. Αυτό συνεπάγεται ότι η αύξηση των αριθμό των χρονικών περιόδων κάτω από το χαρακτηριστικό Type_By_Time δίνει καλύτερα αποτελέσματα. Αυτό πιθανώς να συμβαίνει γιατί, η συμπεριφορά του χρήστη στο τύπο μιας συγκεκριμένης υπηρεσίας χρειάζεται να αλλάζει ακόμη και μέσα στην ίδια χρονική περίοδο. Για παράδειγμα εάν ένας χρήστης έχει στην χρονική περίοδο 13-16 τη δραστηριότητα "lunch time". Πιθανότατα ο χρήστης εάν αναζητήσει κάτι στις 13.00 να ψάχνει για κάποιο κοντινό εστιατόριο. Αντίθετα εάν αναζητήσει κάτι στις 13.30 να ψάχνει κάποιο «ντελιβεράδικο» αφού ο χρόνος του διαλείμματος του τελειώνει σύντομα.

- Καλύτερα αποτελέσματα γενικά φαίνεται να έχει ο αλγόριθμος Cluster Clickstream ο οποίος ωστόσο δεν είχε τα καλά αποτελέσματα που είχαν οι άλλοι δύο αλγόριθμοι με τον αλγόριθμο Density Time Zone. Παρόλα αυτά δίνει πολύ καλύτερα αποτελέσματα στις δύο πρώτες εκτελέσεις του αλγορίθμου.

8.1.2 Αποτελέσματα και σημαντικότητα ποσοστών προτίμησης (Σενάριο 2)

Στα πιο κάτω αποτελέσματα, χρησιμοποιήσαμε τα προφίλ που χρησιμοποιήθηκαν στην τρίτη εκτέλεση των αλγορίθμων του σεναρίου 8.1.1 και μηδενίσαμε τα ποσοστά προτίμησης στο χαρακτηριστικό Time Zones, στη συνέχεια μηδενίσαμε και τα ποσοστά προτίμησης στο χαρακτηριστικό Type_By_Time και τέλος μηδενίσαμε όλα τα ποσοστά προτίμησης στο προφίλ. Σε κάθε εκτέλεση των αλγορίθμων που ενημερώνουν τα προφίλ παίρνουμε μετρήσεις. Στις μετρήσεις αυτές μετρούμε:

1. Την θέση στην οποία εμφανίστηκε η επιθυμητή υπηρεσία για τον χρήστη. (Σύνολο από 7 διαφορετικές υπηρεσίες)
2. Την θέση στην οποία εμφανίστηκε το επιθυμητό στιγμιότυπο μιας υπηρεσίας για τον χρήστη. (Σύνολο από 50 διαφορετικά στιγμιότυπα για κάθε υπηρεσία)

8.1.2.1 Μηδενισμός των ποσοστών προτίμησης του χαρακτηριστικού Time Zones (Σενάριο 2-1)

Στο σενάριο αυτό, μηδενίσαμε το χαρακτηριστικό Time Zones. Το χαρακτηριστικό αυτό είναι υπεύθυνο για την παρουσίαση στο χρήστη όλων των υπηρεσιών του συστήματος, ταξινομημένες με βάση την προτίμησή τους από το χρήστη σε κάποιο συγκεκριμένο χρόνο.

Αποτελέσματα στην Επιλογή Υπηρεσίας

Στο πίνακα που ακολουθεί παρουσιάζονται τα αποτελέσματα στην επιλογή υπηρεσίας για κάθε αλγόριθμο και κάθε παραλλαγή του. Στο σύστημα υπάρχουν 7 υπηρεσίες και κατ' επέκταση 7 πιθανές θέσεις για να παρουσιαστούν. Στη συνέχεια παρουσιάζονται οι γραφικές παραστάσεις των αποτελεσμάτων, μια για κάθε αλγόριθμο. Κάθε γραφική εμφανίζει τα αποτελέσματα για όλες τις παραλλαγές του αλγορίθμου και παράλληλα τα συγκρίνει με τα αποτελέσματα της τρίτης εκτέλεσης του σεναρίου 1, η οποία μας δίνουν την εκτέλεση με τα καλύτερα αποτελέσματα.

Θέση Εμφάνισης	Cluster Clickstream Update Algorithm			Moving Average Clickstream Update Algorithm			Flat Clickstream Update Algorithm		
	Time Zone Algorithm *								
	FC	DB	H	FC	DB	H	FC	DB	H
1	137	188	186	137	181	216	203	202	216
2	13	18	26	13	36	52	50	36	88
3	172	194	201	172	183	177	175	168	168
4	131	141	139	131	145	143	134	134	154
5	203	214	210	203	226	214	216	227	215
6	305	354	351	305	354	332	324	337	291
7	539	391	387	539	375	366	398	396	368

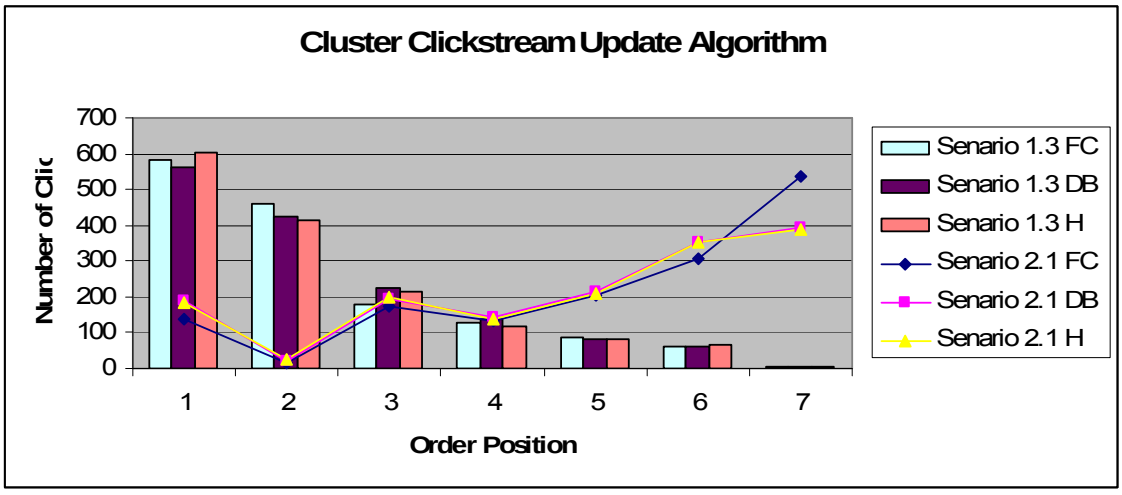
*FC: Flat Clickstream Time Zones Update Algorithm

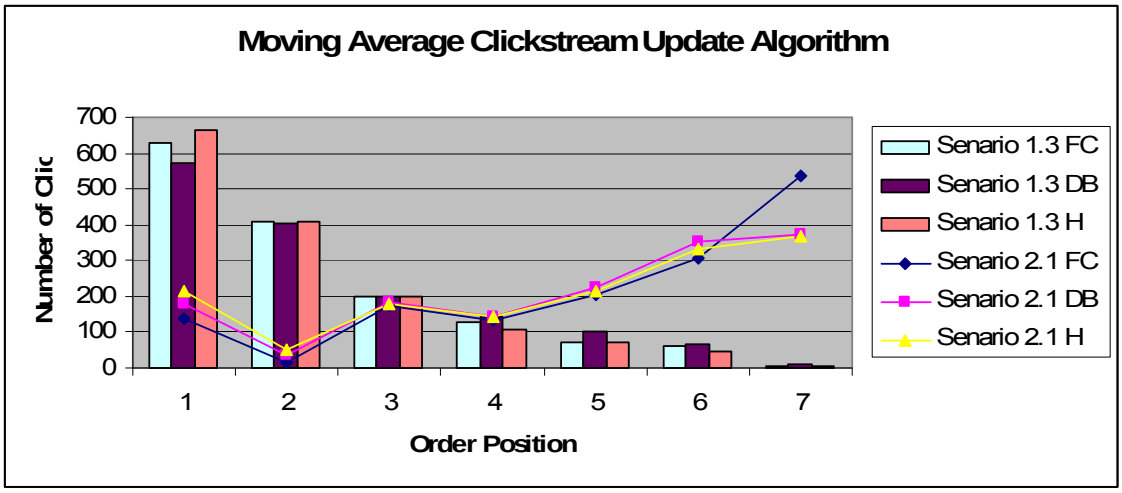
DB: Density Based Time Zones Update Algorithm

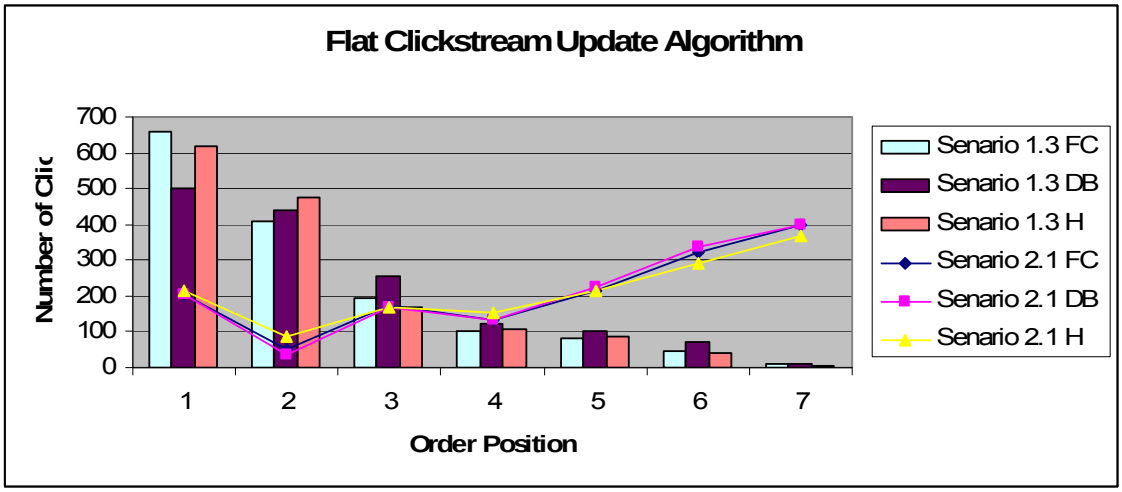
H: Histogram Time Zones Update Algorithm

Διαφορά Αποτελεσμάτων από την Τρίτη Εκτέλεση του Σεναρίου 1

Θέση Εμφάνισης	Cluster Clickstream Update Algorithm			Moving Average Clickstream Update Algorithm			Flat Clickstream Update Algorithm		
	Time Zone Algorithm								
	FC	DB	H	FC	DB	H	FC	DB	H
1	-447	-374	-417	-493	-393	-447	-457	-299	-403
2	-445	-405	-386	-396	-369	-355	-358	-402	-386
3	-9	-33	-13	-25	-18	-21	-19	-87	1
4	5	4	20	2	2	37	31	10	46
5	116	133	128	132	124	142	135	125	128
6	245	291	286	246	289	285	279	266	250
7	535	384	382	534	365	359	389	387	364







Συγκρίνοντας τα πιο πάνω αποτελέσματα με τα αποτελέσματα της τρίτης εκτέλεσης του Σεναρίου 1, παρατηρούμε ότι ο μηδενισμός του χαρακτηριστικού Time_Zones επηρεάζει σημαντικά τα αποτελέσματα σε ότι αφορά την θέση εμφάνισης της υπηρεσίας. Αυτό είναι αναμενόμενο καθ' ότι το χαρακτηριστικό Time_Zones λαμβάνεται υπόψη κατά την εμφάνιση των υπηρεσιών για επιλογή από το χρήστη της επιθυμητής υπηρεσίας. Παρατηρούμε ότι η εξατομίκευση σε αυτό το σημείο μπορεί να δώσει πολύ καλύτερα αποτελέσματα από ένα σύστημα που απλά παρουσιάζει τις υπηρεσίες που προσφέρονται, ή ακόμη και τις υπηρεσίας που ενδιαφέρουν το χρήστη. Ωστόσο ακόμη κι αν μια υπηρεσία είναι αγαπημένη για τον χρήστη, εάν θα επιλεχθεί τελικά ή όχι εξαρτάται από την ώρα της αναζήτησης. Όταν ληφθεί υπόψη το κριτήριο αυτό κατά την εμφάνιση των υπηρεσιών για επιλογή από τον χρήστη και ταξινομηθούν αντίστοιχα οι υπηρεσίες, τα αποτελέσματα είναι ακόμη πιο καλά.

Αποτελέσματα στην Επιλογή Στιγμιοτύπου Υπηρεσίας

Στο πίνακα που ακολουθεί παρουσιάζονται τα αποτελέσματα στην επιλογή στιγμιοτύπου για κάθε αλγόριθμο και κάθε παραλλαγή του. Στο σύστημα υπάρχουν 50 στιγμιότυπα για κάθε υπηρεσία και κατ' επέκταση 50 πιθανές θέσεις για να παρουσιαστούν τα αποτελέσματα.

Θέση Εμφάνισης	**Cluster Clickstream Update Algorithm**			**Moving Average Clickstream Update Algorithm**			**Flat Clickstream Update Algorithm**		
	Time Zone Algorithm								
	FC	**DB**	**H**	**FC**	**DB**	**H**	**FC**	**DB**	**H**
1-5	886	891	899	886	887	874	875	899	893
6-10	174	187	196	165	184	176	167	186	154
11-15	129	106	104	125	107	125	131	110	129
16-20	53	60	60	61	66	63	67	55	59
21-25	40	31	35	32	47	56	44	50	54
26-30	33	34	26	41	48	32	34	28	48
31-35	38	36	35	46	32	31	30	50	40
36-40	53	66	65	48	49	53	51	50	42
41-45	54	37	38	40	39	42	45	37	47
46-50	40	52	42	56	41	48	56	35	34

Συγκρίνοντας τα πιο πάνω αποτελέσματα με αυτά της τρίτης εκτέλεσης του Σεναρίου 1, παρατηρούμε ότι δεν υπάρχουν σημαντικές αλλαγές. Αυτό είναι αναμενόμενο καθ' ότι για την ταξινόμηση των στιγμιοτύπων μιας υπηρεσίας λαμβάνουμε υπόψη μόνο το χαρακτηριστικό Type_By_Time.

8.1.2.2 Μηδενισμός των ποσοστών προτίμησης των χρονικών περιόδων (Σενάριο 2-2)

Στο σενάριο αυτό, μηδενίζουμε τον παράγοντα χρόνο έτσι ώστε να μπορέσουμε να δούμε πως ο παράγοντας αυτός επηρεάζει τα αποτελέσματα που εμφανίζονταις το χρήστη. Για να το κάνουμε αυτό, μηδενίζουμε τα ποσοστά προτίμησης στα χαρακτηριστικά TimeZones και TybeByTime.

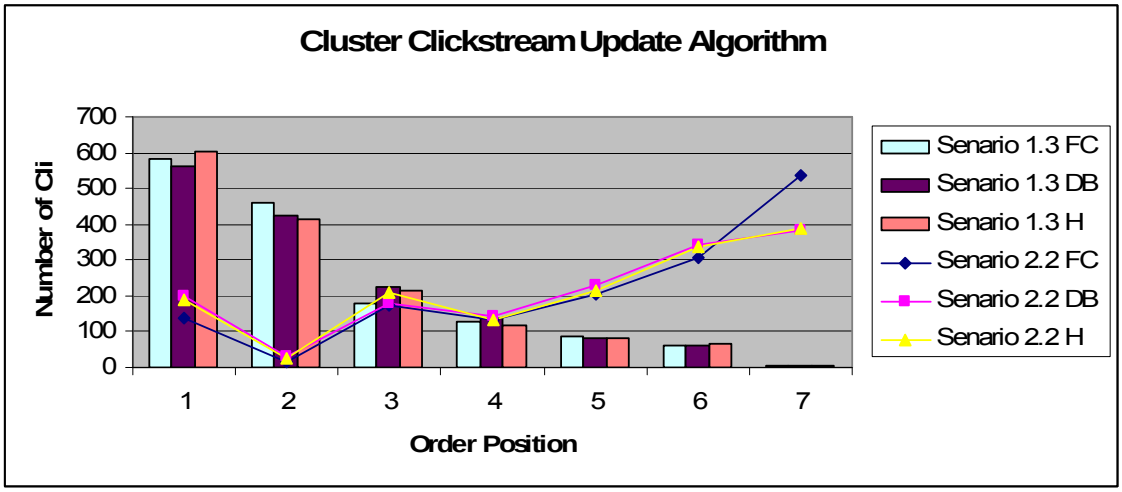
Αποτελέσματα στην Επιλογή Υπηρεσίας

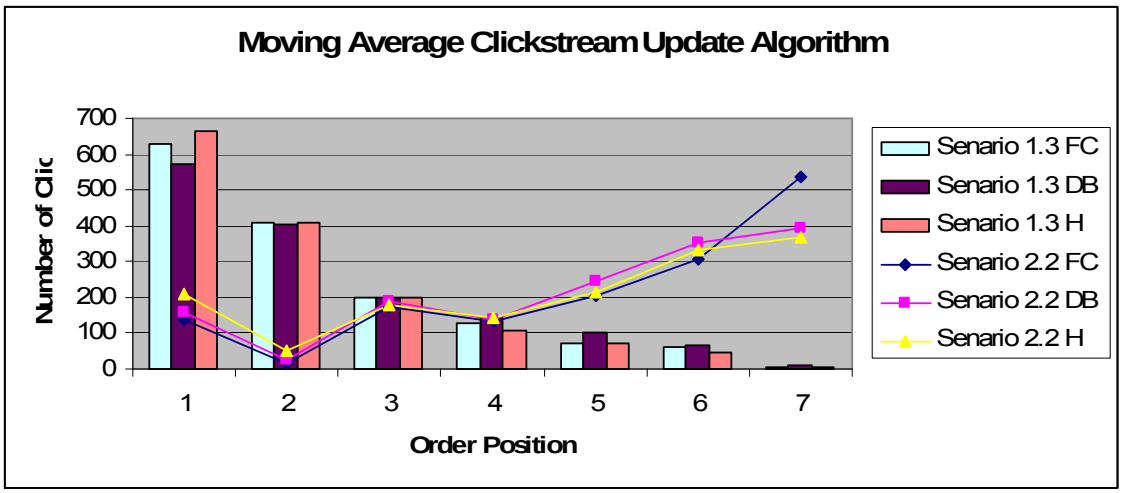
Στο πίνακα που ακολουθεί παρουσιάζονται τα αποτελέσματα στην επιλογή υπηρεσίας για κάθε αλγόριθμο και κάθε παραλλαγή του. Στο σύστημα υπάρχουν 7 υπηρεσίες και κατ' επέκταση 7 πιθανές θέσεις για να παρουσιαστούν. Στη συνέχεια παρουσιάζονται οι γραφικές παραστάσεις των αποτελεσμάτων, μια για κάθε αλγόριθμο. Κάθε γραφική εμφανίζει τα αποτελέσματα για όλες τις παραλλαγές του αλγορίθμου και παράλληλα τα συγκρίνει με τα αποτελέσματα της τρίτης εκτέλεσης του σεναρίου 1, η οποία μας δίνουν την εκτέλεση με τα καλύτερα αποτελέσματα.

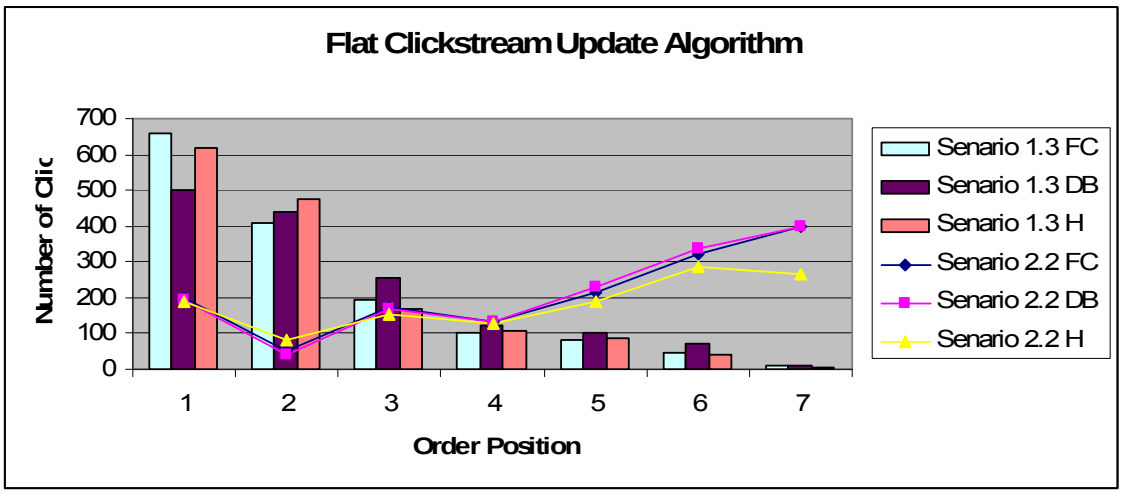
Θέση Εμφάνισης	**Cluster Clickstream Update Algorithm**			**Moving Average Clickstream Update Algorithm**			**Flat Clickstream Update Algorithm**		
	Time Zone Algorithm								
	FC	**DB**	**H**	**FC**	**DB**	**H**	**FC**	**DB**	**H**
1	137	198	191	137	156	212	201	196	189
2	13	30	27	13	28	52	50	40	80
3	172	180	207	172	189	177	175	171	151
4	131	142	134	131	137	143	134	131	127
5	203	228	215	203	245	214	216	229	190
6	305	341	336	305	353	332	324	336	285
7	539	381	390	539	392	370	400	397	268

Διαφορά Αποτελεσμάτων από την Τρίτη Εκτέλεση του Σεναρίου 1

Θέση Εμφάνισης	**Cluster Clickstream Update Algorithm**			**Moving Average Clickstream Update Algorithm**			**Flat Clickstream Update Algorithm**		
	Time Zone Algorithm								
	FC	**DB**	**H**	**FC**	**DB**	**H**	**FC**	**DB**	**H**
1	-447	-364	-412	-493	-418	-451	-459	-305	-430
2	-445	-393	-385	-396	-377	-355	-358	-398	-394
3	-9	-47	-7	-25	-12	-21	-19	-84	-16
4	5	5	15	2	-6	37	31	7	19
5	116	147	133	132	143	142	135	127	103
6	245	278	271	246	288	285	279	265	244
7	535	374	385	534	382	363	391	388	264







Όπως έχουμε ήδη παρατηρήσει και από το προηγούμενο σενάριο, ο παράγοντας χρόνος επηρεάζει σημαντικά τις επιλογές του χρήστη σε ότι αφορά την επιλογή της

επιθυμητής υπηρεσίας, και άγνοια του παράγοντα αυτού δίνει αποτελέσματα τα οποία δεν είναι καθόλου ικανοποιητικά.

Αποτελέσματα στην Επιλογή Στιγμιοτύπων Υπηρεσίας

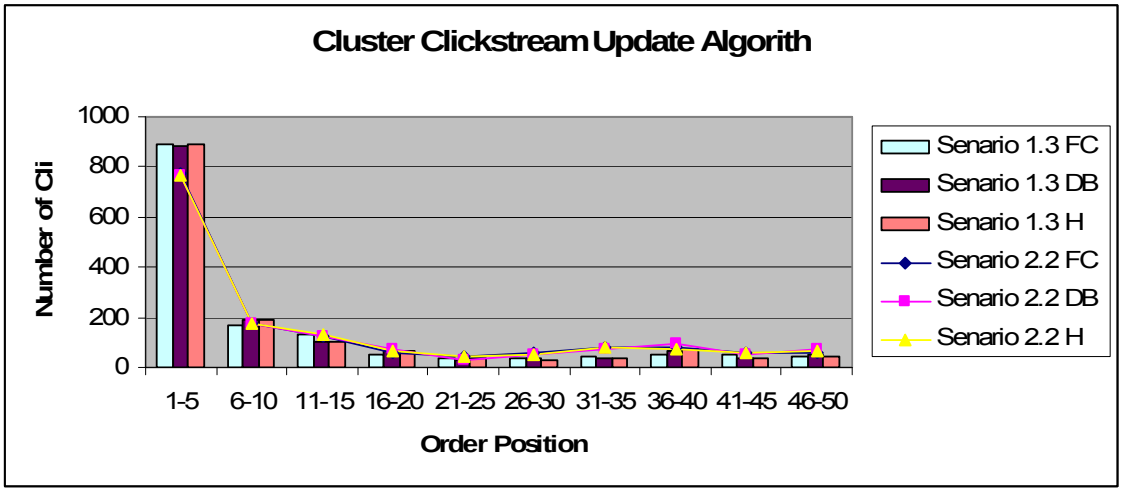
Στο πίνακα που ακολουθεί παρουσιάζονται τα αποτελέσματα στην επιλογή στιγμιοτύπου για κάθε αλγόριθμο και κάθε παραλλαγή του. Στο σύστημα υπάρχουν 50 στιγμιότυπα για κάθε υπηρεσία και κατ' επέκταση 50 πιθανές θέσεις για να παρουσιαστούν τα αποτελέσματα. Στη συνέχεια παρουσιάζονται οι γραφικές παραστάσεις των αποτελεσμάτων, μια για κάθε αλγόριθμο. Κάθε γραφική εμφανίζει τα αποτελέσματα για όλες τις παραλλαγές του αλγορίθμου και παράλληλα τα συγκρίνει με τα αποτελέσματα της τρίτης εκτέλεσης του σεναρίου 1, η οποία μας δίνουν την εκτέλεση με τα καλύτερα αποτελέσματα.

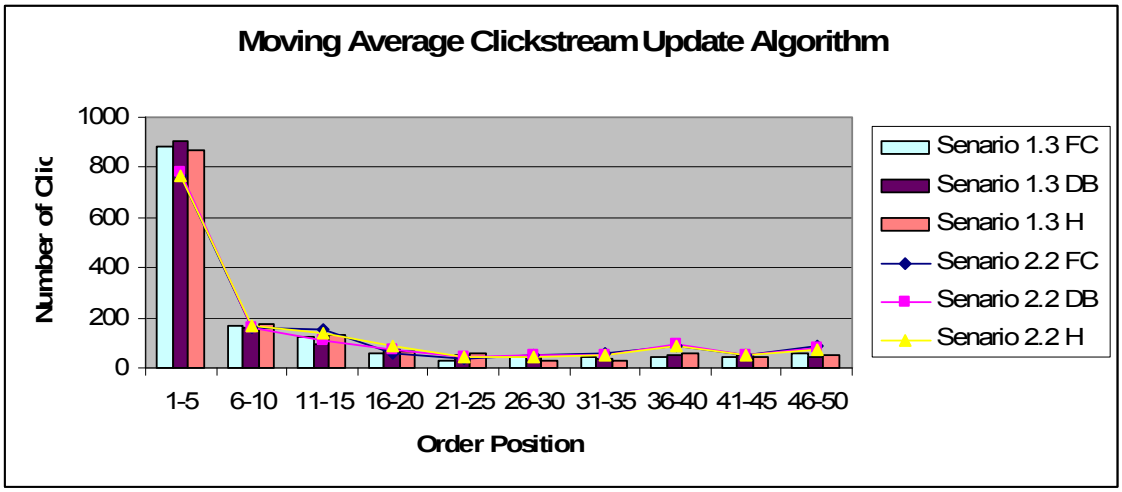
Θέση Εμφάνισης	**Cluster Clickstream Update Algorithm**			**Moving Average Clickstream Update Algorithm**			**Flat Clickstream Update Algorithm**		
	Time Zone Algorithm								
	FC	**DB**	**H**	**FC**	**DB**	**H**	**FC**	**DB**	**H**
1-5	771	766	766	765	781	766	766	779	783
6-10	174	173	172	158	161	167	164	166	172
11-15	122	121	131	150	111	136	139	123	119
16-20	60	70	64	58	75	86	83	57	59
21-25	43	30	42	36	44	41	36	52	62
26-30	57	51	52	49	50	41	42	46	61
31-35	77	73	78	57	52	53	56	74	63
36-40	79	92	72	88	94	85	78	80	70
41-45	57	51	59	54	54	53	58	58	62
46-50	60	73	64	85	78	72	78	65	49

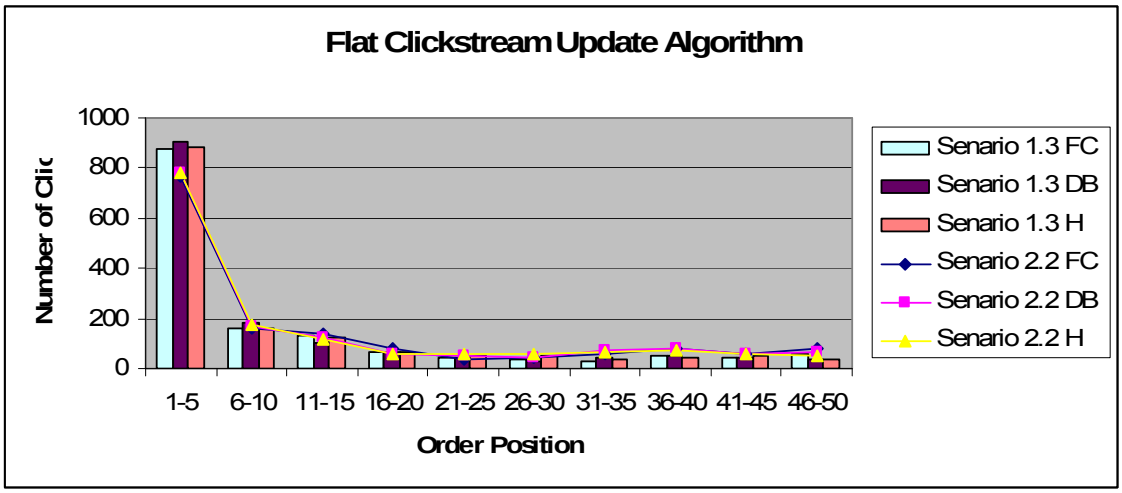
Διαφορά Αποτελεσμάτων από την Τρίτη Εκτέλεση του Σεναρίου 1

Θέση Εμφάνισης	**Cluster Clickstream Update Algorithm**			**Moving Average Clickstream Update Algorithm**			**Flat Clickstream Update Algorithm**		
	Time Zone Algorithm								
	FC	**DB**	**H**	**FC**	**DB**	**H**	**FC**	**DB**	**H**
1-5	-118	-119	-128	-116	-124	-105	-109	-125	-98
6-10	7	-15	-19	-12	4	-5	5	-17	15
11-15	-6	20	30	28	14	7	4	20	-8
16-20	10	3	-1	-3	5	22	15	-7	-6
21-25	6	0	7	4	-6	-14	-9	10	9
26-30	21	17	26	4	7	10	8	20	11
31-35	32	37	43	14	18	22	26	30	23
36-40	25	25	2	42	45	30	25	26	26

41-45	6	11	21	10	5	11	13	17	12
46-50	17	21	19	29	32	22	22	26	16







Όπως παρατηρούμε από τον πιο πάνω η απόδοση των αλγορίθμων μειώνεται σε ότι αφορά τα στιγμιότυπα που εμφανίζονται στις πρώτες θέσεις και αυξάνεται σε ότι αφορά τα στιγμιότυπα που εμφανίζονται στις τελευταίες θέσεις. Πιο συγκεκριμένα τα στιγμιότυπα που εμφανίζονται στις θέσεις 1-5 μειώνονται κατά 10-15 %. Αυτό συνεπάγεται, ότι η επιλογή ενός στιγμιοτύπου μιας υπηρεσίας από τον χρήστη εξαρτάται από την χρονική στιγμή της αναζήτησης και τον τύπο του στιγμιοτύπου.

8.1.2.3 Μηδενισμός όλων των ποσοστών προτίμησης (Σενάριο 2-3)

Στο σενάριο αυτό, μηδενίζουμε όλα τα ποσοστά προτίμησης. Με τον τρόπο αυτό ελέγχουμε πώς θα συμπεριφερόταν το σύστημα εάν δεν κρατούσαμε προφίλ χρηστών.

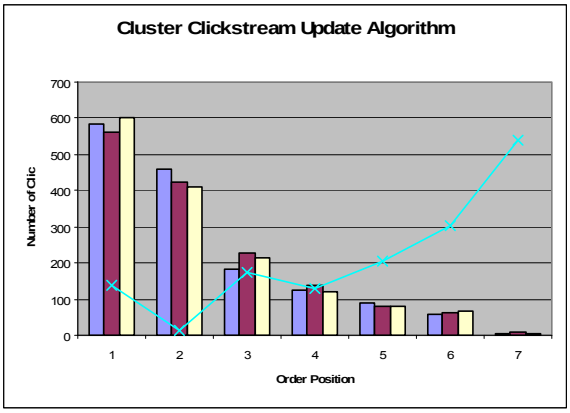
Αποτελέσματα στην Επιλογή Υπηρεσίας

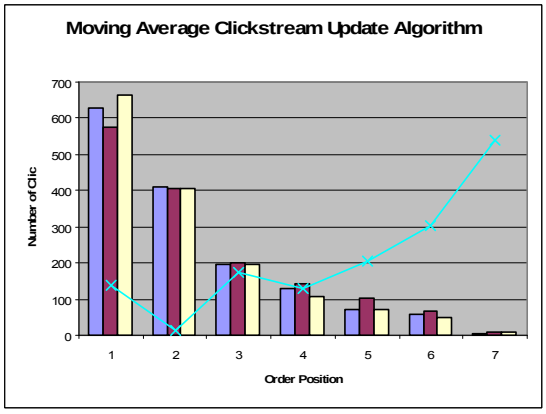
Στο πίνακα που ακολουθεί παρουσιάζονται τα αποτελέσματα στην επιλογή υπηρεσίας για όλους τους αλγορίθμους. Η συμπεριφορά του συστήματος, από τη στιγμή που τα ποσοστά προτίμησης δεν λαμβάνουν μέρος στις μετρήσεις και δεν επηρεάζουν τα αποτελέσματα, είναι η ίδια για κάθε αλγόριθμο. Στο σύστημα υπάρχουν 7 υπηρεσίες και κατ' επέκταση 7 πιθανές θέσεις για να παρουσιαστούν. Στη συνέχεια παρουσιάζονται οι γραφικές παραστάσεις των αποτελεσμάτων, μια για κάθε αλγόριθμο. Κάθε γραφική εμφανίζει τα αποτελέσματα τρίτης εκτέλεσης του σεναρίου 1, η οποία μας δίνουν την εκτέλεση με τα καλύτερα αποτελέσματα σε σύγκριση με τα αποτελέσματα όταν μηδενίσουμε τα ποσοστά προτίμησης.

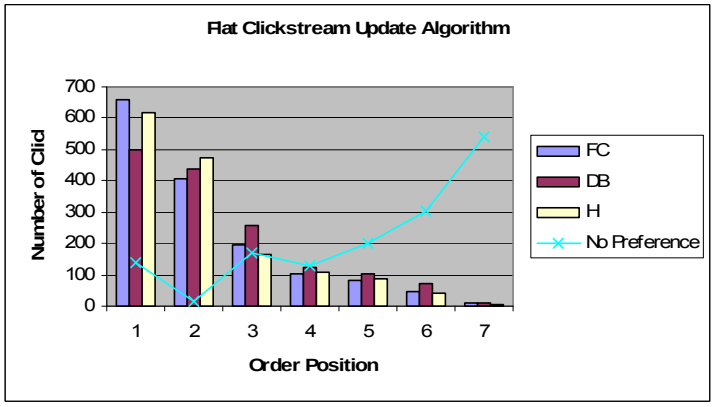
Θέση Εμφάνισης	**Όλοι Αλγόριθμοι**
1	137
2	13
3	172
4	131
5	203
6	305
7	539

Διαφορά Αποτελεσμάτων από την Τρίτη Εκτέλεση του Σεναρίου 1

Θέση Εμφάνισης	Cluster Clickstream Update Algorithm			Moving Average Clickstream Update Algorithm			Flat Clickstream Update Algorithm		
	Time Zone Algorithm								
	FC	DB	H	FC	DB	H	FC	DB	H
1	-447	-425	-466	-493	-437	-526	-523	-364	-482
2	-445	-410	-399	-396	-392	-394	-395	-425	-461
3	-9	-55	-42	-25	-29	-26	-22	-83	5
4	5	-6	12	2	-12	25	28	7	23
5	116	122	121	132	101	131	122	101	116
6	245	242	240	246	240	258	260	234	264
7	535	532	534	534	529	532	530	530	535







Αποτελέσματα στην Επιλογή Στιγμιότυπου Υπηρεσίας

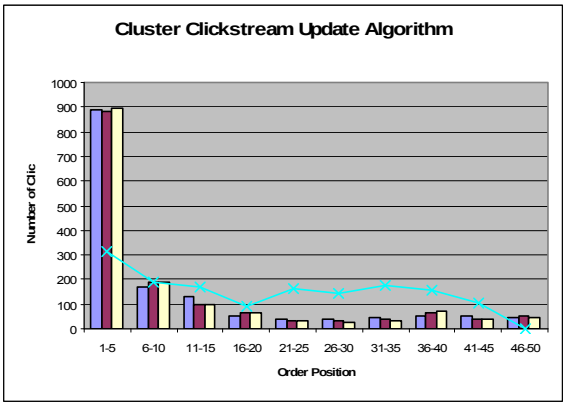
Στο πίνακα που ακολουθεί παρουσιάζονται τα αποτελέσματα στην επιλογή Στιγμιοτύπου για όλους τους αλγορίθμους. Η συμπεριφορά του συστήματος, από τη στιγμή που τα ποσοστά προτίμησης δεν λαμβάνουν μέρος στις μετρήσεις και δεν επηρεάζουν τα αποτελέσματα, είναι η ίδια για κάθε αλγόριθμο. Στο σύστημα υπάρχουν 50 στιγμιότυπα για κάθε υπηρεσία και κατ' επέκταση 50 πιθανές θέσεις για να παρουσιαστούν. Στη συνέχεια παρουσιάζονται οι γραφικές παραστάσεις των

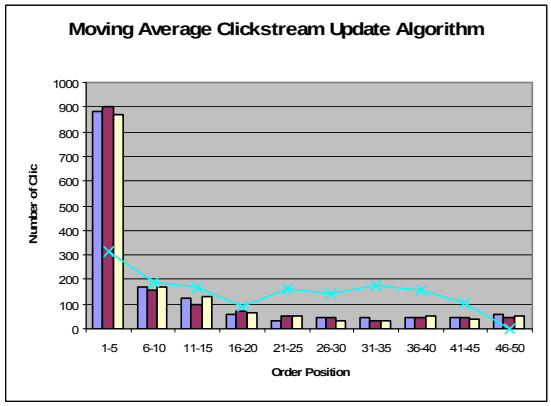
αποτελεσμάτων, μια για κάθε αλγόριθμο. Κάθε γραφική εμφανίζει τα αποτελέσματα τρίτης εκτέλεσης του σεναρίου 1, η οποία μας δίνουν την εκτέλεση με τα καλύτερα αποτελέσματα, σε σύγκριση με τα αποτελέσματα όταν μηδενίσουμε τα ποσοστά προτίμησης.

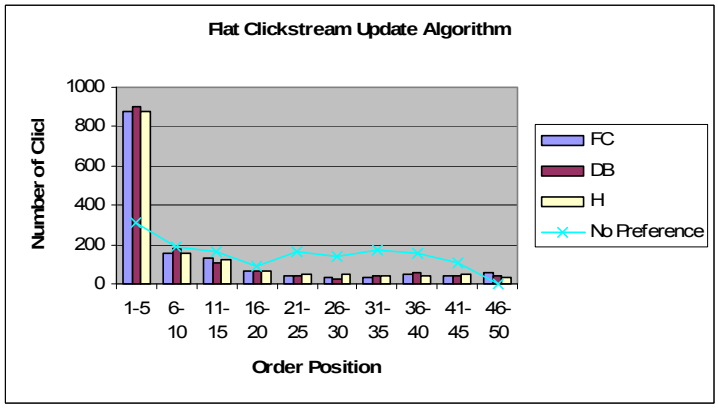
Θέση Εμφάνισης	**Όλοι Αλγόριθμοι**
1-5	311
6-10	187
11-15	167
16-20	93
21-25	161
26-30	141
31-35	176
36-40	157
41-45	107
46-50	0

Διαφορά Αποτελεσμάτων από την Τρίτη Εκτέλεση του Σεναρίου 1

Θέση Εμφάνισης	**Cluster Clickstream Update Algorithm**			**Moving Average Clickstream Update Algorithm**			**Flat Clickstream Update Algorithm**		
	Time Zone Algorithm								
	FC	**DB**	**H**	**FC**	**DB**	**H**	**FC**	**DB**	**H**
1-5	-118	-119	-128	-116	-124	-105	-109	-125	-98
6-10	7	-15	-19	-12	4	-5	5	-17	15
11-15	-6	20	30	28	14	7	4	20	-8
16-20	10	3	-1	-3	5	22	15	-7	-6
21-25	6	0	7	4	-6	-14	-9	10	9
26-30	21	17	26	4	7	10	8	20	11
31-35	32	37	43	14	18	22	26	30	23
36-40	25	25	2	42	45	30	25	26	26
41-45	6	11	21	10	5	11	13	17	12
46-50	17	21	19	29	32	22	22	26	16







Από τα πιο πάνω φαίνεται η σημαντικότητα των ποσοστών προτίμησης στα χαρακτηριστικά των υπηρεσιών και πως μειώνεται η αποτελεσματικότητα ενός συστήματος εάν αγνοήσει τις προτιμήσεις του χρήστη.

Στα πιο πάνω αποτελέσματα, στις θέσεις 1-5 φαίνεται να εμφανίζονται τα περισσότερα αποτελέσματα. Αυτό μπορεί να δικαιολογηθεί, εάν λάβουμε υπόψη ότι τα αποτελέσματα επηρεάζει σημαντικά η θέση με την οποία βρίσκονται στο σύστημα τα στιγμιότυπα. Εάν κάποια στιγμιότυπα τα οποία βρίσκονται καταχωρημένα στις πρώτες θέσεις μέσα στο σύστημα εμφανίστηκαν και στις πρώτες επιλογές του χρήστη, ο μηδενισμός του ποσοστού προτίμησης δεν τα επηρεάζει.

Επιπλέον βλέπουμε ότι κανένα από τα στιγμιότυπα που επιθυμεί ο χρήστης δεν εμφανίζονται στις θέσεις 46-50 όταν μηδενίσουμε τα ποσοστά. Αυτό συμβαίνει γιατί πολύ πιθανό σε κάποια χαρακτηριστικά των στιγμιοτύπων αυτών υπάρχουν ή πολύ χαμηλά ποσοστά ή ακόμη και αρνητικά. Κάτι που μας δείχνει ότι με την διατήρηση του προφίλ υπάρχει και ένα μικρό ποσοστό των υπηρεσιών που αναζητεί ο χρήστης να εμφανιστούν πολύ χαμηλά λόγω ακριβώς των ποσοστών στο προφίλ.

8.1.3 Αποτελέσματα και Σημαντικότητα Experience και Χρονικών Περιόδων (Σενάριο 3)

Στο σενάριο αυτό μελετούμε την σημαντικότητα των σωστών χρονικών περιόδων και experience. Πως αυτά επηρεάζουν την αποτελεσματικότητα του συστήματος και γιατί είναι σημαντική η ύπαρξη των εννοιών αυτών στα συστήματα εξατομίκευσης για κινητούς χρήστες.

8.1.3.1 Χρήση Λανθασμένης Χρονικής Περιόδου (Σενάριο 3-1)

Στο σενάριο αυτό τρέξαμε στους αλγορίθμους το προτεινόμενο clickstream (Preferable Clickstream) μιας χρονικής περιόδου για κάθε χρήστη, σε διαφορετικό χρόνο. Πιο συγκεκριμένα τρέξαμε το προτεινόμενο clickstream για την χρονική περίοδο 21:00 – 00:00, κατά τις 13:30. Πιο κάτω φαίνονται τα αποτελέσματα του σεναρίου αυτού.

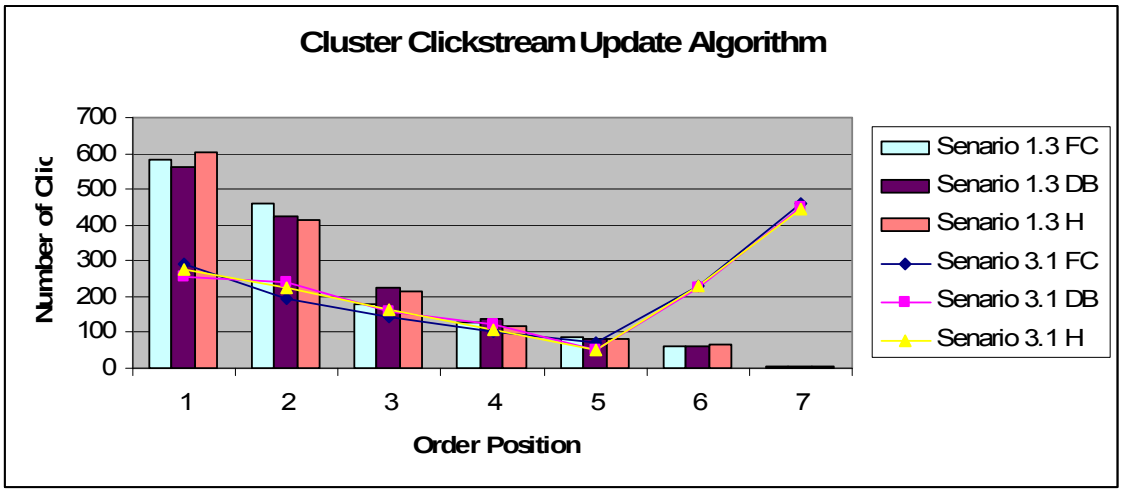
Αποτελέσματα στην Επιλογή Υπηρεσίας

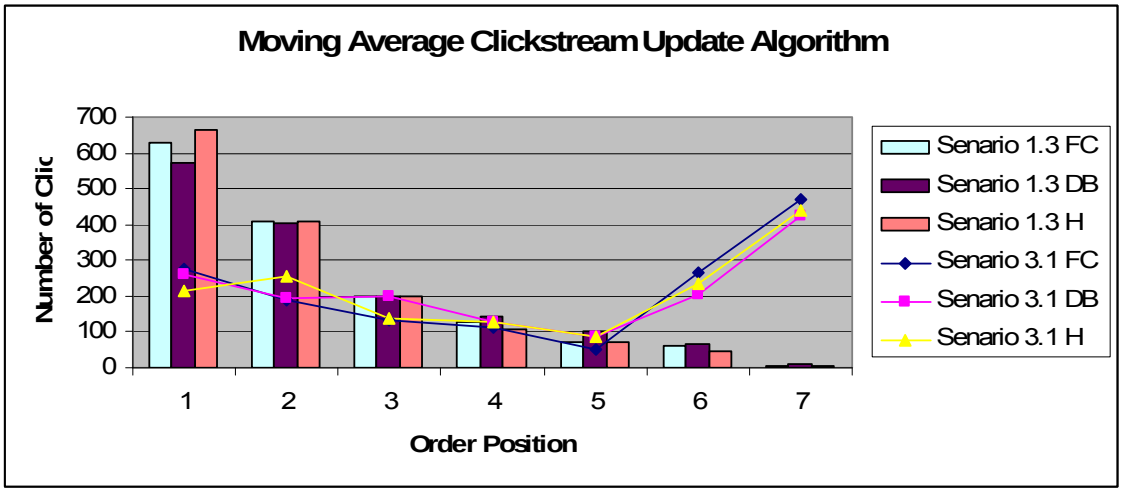
Στο πίνακα που ακολουθεί παρουσιάζονται τα αποτελέσματα στην επιλογή υπηρεσίας για κάθε αλγόριθμο και κάθε παραλλαγή του. Στο σύστημα υπάρχουν 7 υπηρεσίες και κατ' επέκταση 7 πιθανές θέσεις για να παρουσιαστούν. Στη συνέχεια παρουσιάζονται οι γραφικές παραστάσεις των αποτελεσμάτων, μια για κάθε αλγόριθμο. Κάθε γραφική εμφανίζει τα αποτελέσματα για όλες τις παραλλαγές του αλγορίθμου και παράλληλα τα συγκρίνει με τα αποτελέσματα της τρίτης εκτέλεσης του σεναρίου 1, η οποία μας δίνουν την εκτέλεση με τα καλύτερα αποτελέσματα.

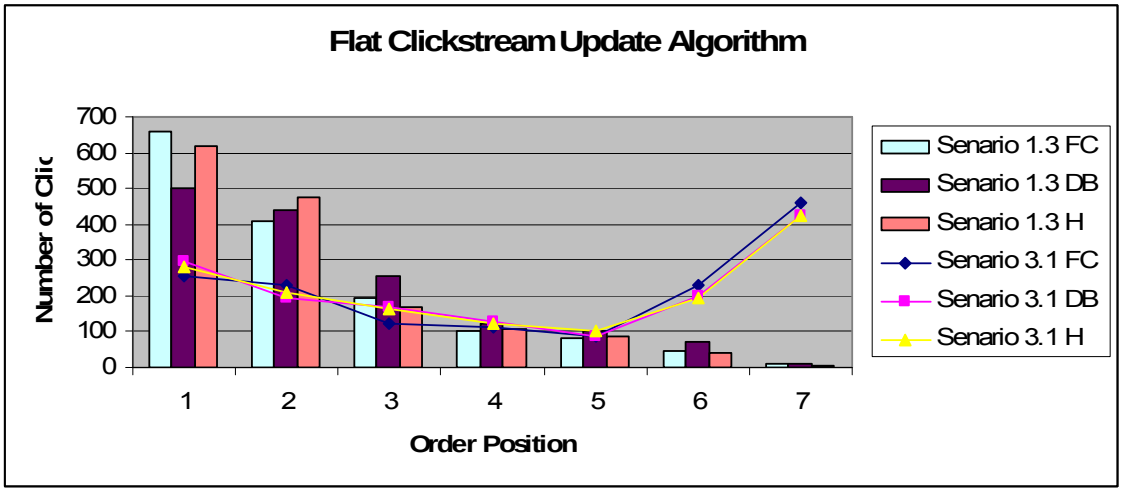
Θέση Εμφάνισης	**Cluster Clickstream Update Algorithm**			**Moving Average Clickstream Update Algorithm**			**Flat Clickstream Update Algorithm**		
	Time Zone Algorithm *								
	FC	**DB**	**H**	**FC**	**DB**	**H**	**FC**	**DB**	**H**
1	290	256	277	275	260	214	253	296	282
2	195	241	227	189	195	257	232	196	207
3	144	157	165	131	199	140	124	170	166
4	103	123	107	113	128	130	112	129	124
5	74	50	49	53	88	85	89	89	102
6	232	223	232	267	206	234	229	197	196
7	462	450	443	472	424	440	461	423	423

Διαφορά Αποτελεσμάτων από την Τρίτη Εκτέλεση του Σεναρίου 1

Θέση Εμφάνισης	Cluster Clickstream Update Algorithm			Moving Average Clickstream Update Algorithm			Flat Clickstream Update Algorithm		
	Time Zone Algorithm								
	FC	DB	H	FC	DB	H	FC	DB	H
1	-294	-306	-326	-355	-314	-449	-407	-205	-337
2	-263	-182	-185	-220	-210	-150	-176	-242	-267
3	-37	-70	-49	-66	-2	-58	-70	-85	-1
4	-23	-14	-12	-16	-15	24	9	5	16
5	-13	-31	-33	-18	-14	13	8	-13	15
6	172	160	167	208	141	187	184	126	155
7	458	443	438	467	414	433	452	414	419







Όπως φαίνεται από τα πιο πάνω αποτελέσματα ο παράγοντας χρόνος επηρεάζει σημαντικά τις επιλογές του χρήστη και άγνοια του παράγοντα αυτού δίνει αποτελέσματα τα οποία δεν είναι καθόλου ικανοποιητικά.

Αποτελέσματα στην Επιλογή Στιγμιοτύπων Υπηρεσίας

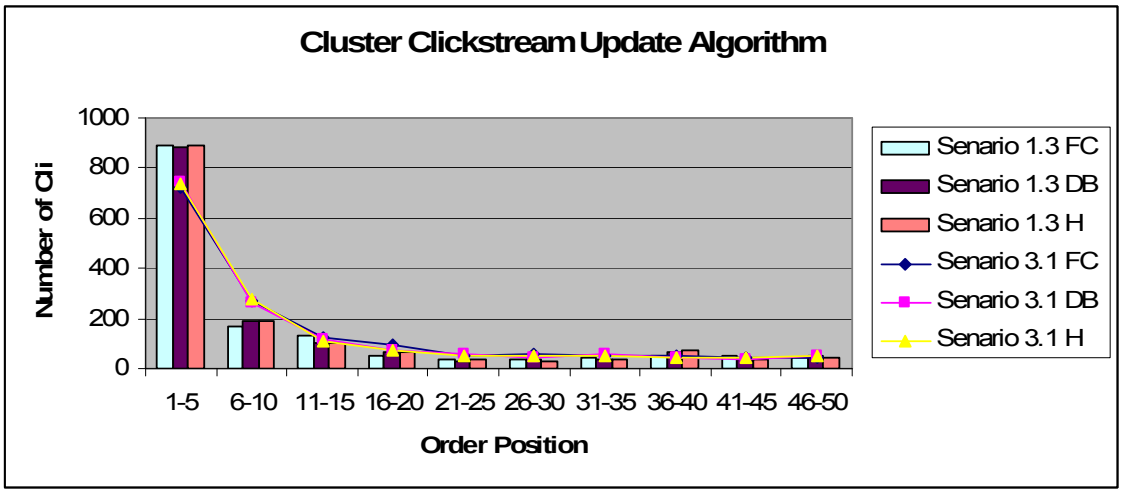
Στο πίνακα που ακολουθεί παρουσιάζονται τα αποτελέσματα στην επιλογή στιγμιοτύπου για κάθε αλγόριθμο και κάθε παραλλαγή του. Στο σύστημα υπάρχουν 50 στιγμιότυπα για κάθε υπηρεσία και κατ' επέκταση 50 πιθανές θέσεις για να παρουσιαστούν τα αποτελέσματα. Στη συνέχεια παρουσιάζονται οι γραφικές παραστάσεις των αποτελεσμάτων, μια για κάθε αλγόριθμο. Κάθε γραφική εμφανίζει τα αποτελέσματα για όλες τις παραλλαγές του αλγορίθμου και παράλληλα τα συγκρίνει με τα αποτελέσματα της τρίτης εκτέλεσης του σεναρίου 1, η οποία μας δίνουν την εκτέλεση με τα καλύτερα αποτελέσματα.

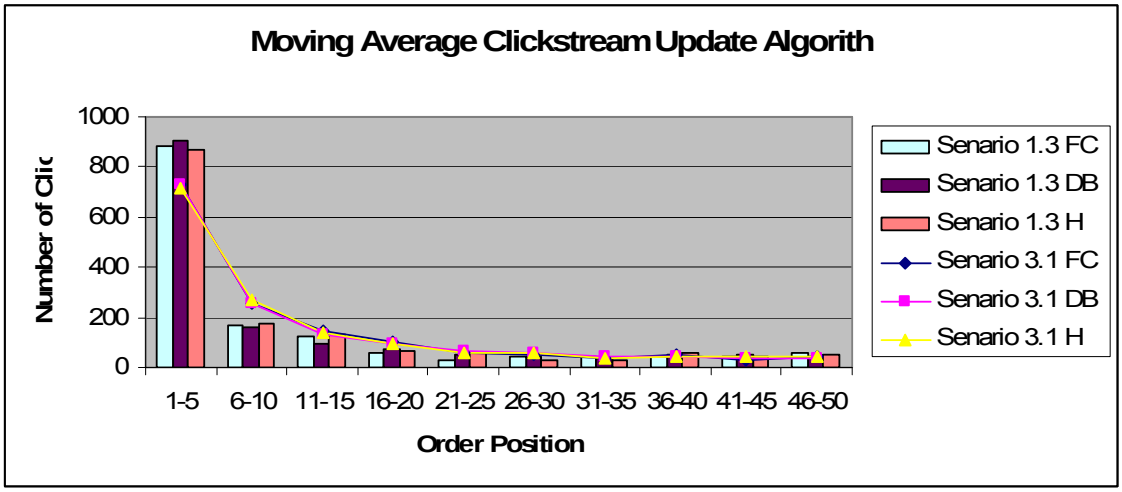
Θέση Εμφάνισης	**Cluster Clickstream Update Algorithm**			**Moving Average Clickstream Update Algorithm**			**Flat Clickstream Update Algorithm**		
	Time Zone Algorithm								
	FC	**DB**	**H**	**FC**	**DB**	**H**	**FC**	**DB**	**H**
1-5	723	746	739	729	727	713	725	648	675
6-10	267	265	275	253	257	273	261	287	304
11-15	122	120	110	149	134	138	140	128	134
16-20	92	71	71	99	97	95	96	103	104
21-25	50	61	54	55	67	59	52	72	54
26-30	58	46	54	54	55	55	57	46	52
31-35	50	55	51	37	42	40	41	44	52

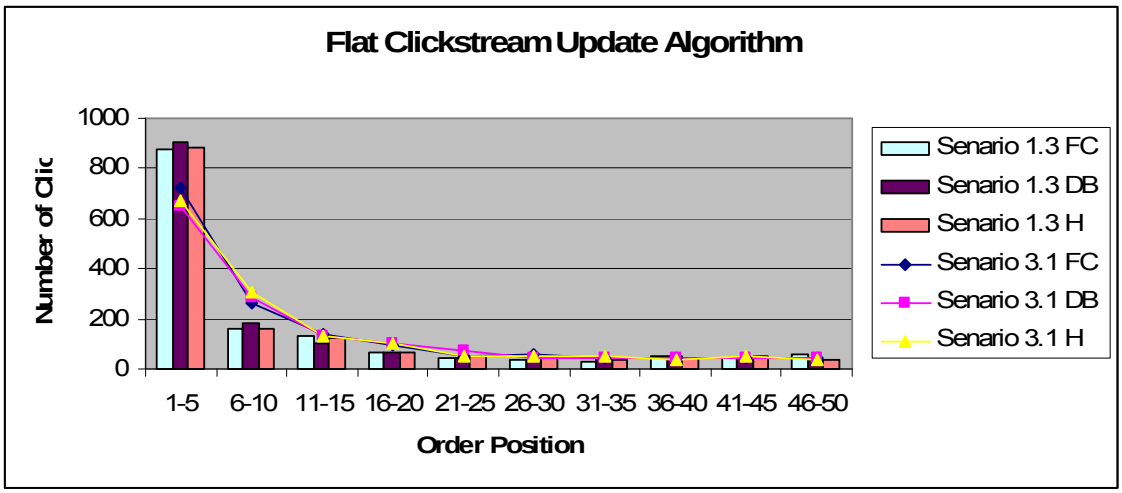
36-40	52	44	46	50	45	41	41	47	39
41-45	45	40	47	31	38	45	43	47	51
46-50	41	52	53	43	38	41	44	42	35

Διαφορά Αποτελεσμάτων από την Τρίτη Εκτέλεση του Σεναρίου 1

Θέση Εμφάνισης	**Cluster Clickstream Update Algorithm**			**Moving Average Clickstream Update Algorithm**			**Flat Clickstream Update Algorithm**		
	Time Zone Algorithm								
	FC	**DB**	**H**	**FC**	**DB**	**H**	**FC**	**DB**	**H**
1-5	-166	-139	-155	-152	-178	-158	-150	-256	-206
6-10	100	77	84	83	100	101	102	104	147
11-15	-6	19	9	27	37	9	5	25	7
16-20	42	4	6	38	27	31	28	39	39
21-25	13	31	19	23	17	4	7	30	1
26-30	22	12	28	9	12	24	23	20	2
31-35	5	19	16	-6	8	9	11	0	12
36-40	-2	-23	-24	4	-4	-14	-12	-7	-5
41-45	-6	0	9	-13	-11	3	-2	6	1
46-50	-2	0	8	-13	-8	-9	-12	3	2







Όπως παρατηρούμε από το πιο πάνω η απόδοση των αλγορίθμων μειώνεται σε ότι αφορά τα στιγμιότυπα που εμφανίζονται στις πρώτες θέσεις και αυξάνεται σε ότι αφορά τα στιγμιότυπα που εμφανίζονται στις τελευταίες θέσεις. Πιο συγκεκριμένα τα στιγμιότυπα που εμφανίζονται στις θέσεις 1-5 μειώνονται κατά 10-15 %.

8.1.3.2 Χρήση Λανθασμένου Experience (Σενάριο 3-2)

Στο σενάριο αυτό, τρέξαμε στους αλγορίθμους το επιθυμητό clickstream (Preferable Clickstream) ενός συγκεκριμένου experience, σε διαφορετικό experience από αυτό στο οποίο αντιστοιχούσε το clickstream.

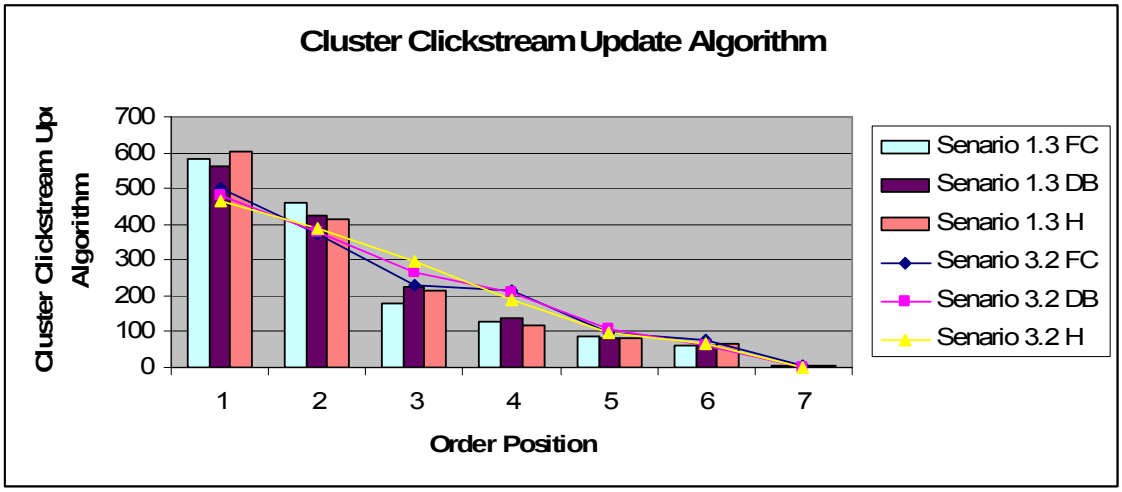
Αποτελέσματα στην Επιλογή Υπηρεσίας

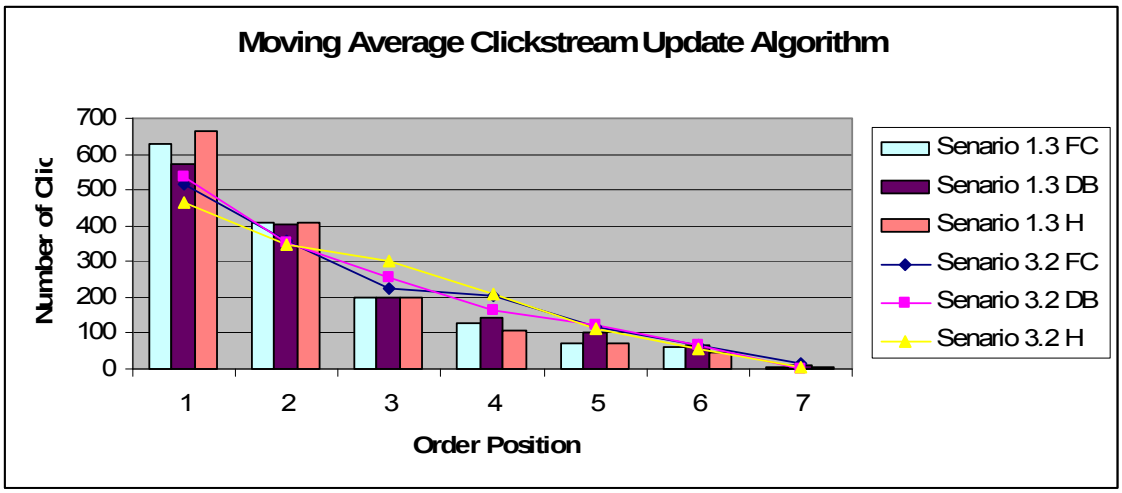
Στο πίνακα που ακολουθεί παρουσιάζονται τα αποτελέσματα στην επιλογή υπηρεσίας για κάθε αλγόριθμο και κάθε παραλλαγή του. Στο σύστημα υπάρχουν 7 υπηρεσίες και κατ' επέκταση 7 πιθανές θέσεις για να παρουσιαστούν. Στη συνέχεια παρουσιάζονται οι γραφικές παραστάσεις των αποτελεσμάτων, μια για κάθε αλγόριθμο. Κάθε γραφική εμφανίζει τα αποτελέσματα για όλες τις παραλλαγές του αλγορίθμου και παράλληλα τα συγκρίνει με τα αποτελέσματα της τρίτης εκτέλεσης του σεναρίου 1, η οποία μας δίνουν την εκτέλεση με τα καλύτερα αποτελέσματα.

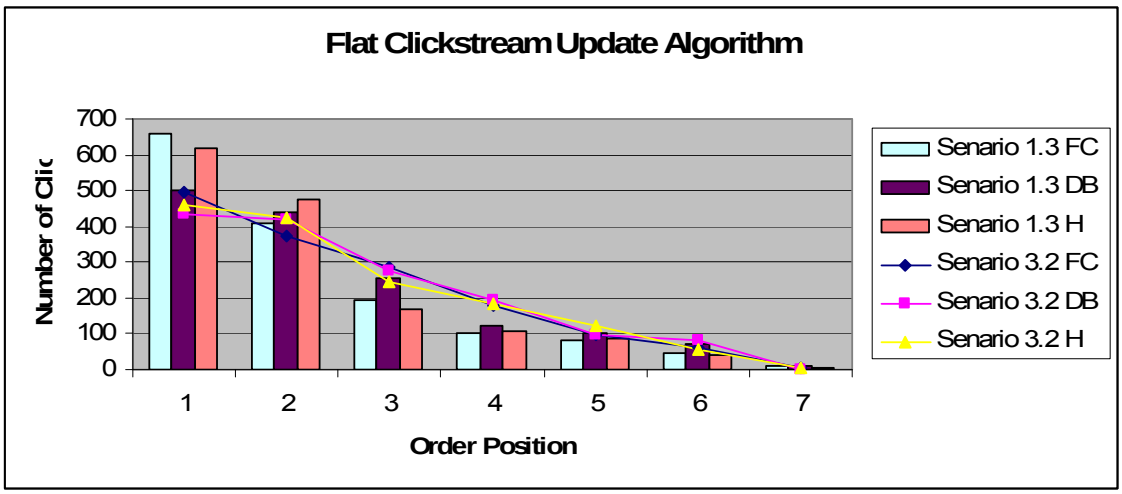
Θέση Εμφάνισης	**Cluster Clickstream Update Algorithm**			**Moving Average Clickstream Update Algorithm**			**Flat Clickstream Update Algorithm**		
	Time Zone Algorithm								
	FC	**DB**	**H**	**FC**	**DB**	**H**	**FC**	**DB**	**H**
1	502	481	465	515	534	467	497	434	462
2	371	377	389	360	355	350	374	421	422
3	232	264	296	226	258	301	288	274	246
4	213	212	188	202	162	212	177	195	185
5	97	105	95	117	125	110	98	96	121
6	79	60	65	66	64	55	61	80	58
7	6	1	2	14	2	5	5	0	6

Διαφορά Αποτελεσμάτων από την Τρίτη Εκτέλεση του Σεναρίου 1

Θέση Εμφάνισης	**Cluster Clickstream Update Algorithm**			**Moving Average Clickstream Update Algorithm**			**Flat Clickstream Update Algorithm**		
	Time Zone Algorithm								
	FC	**DB**	**H**	**FC**	**DB**	**H**	**FC**	**DB**	**H**
1	-82	-81	-138	-115	-40	-196	-163	-67	-157
2	-87	-46	-23	-49	-50	-57	-34	-17	-52
3	51	37	82	29	57	103	94	19	79
4	87	75	69	73	19	106	74	71	77
5	10	24	13	46	23	38	17	-6	34
6	19	-3	0	7	-1	8	16	9	17
7	2	-6	-3	9	-8	-2	-4	-9	2







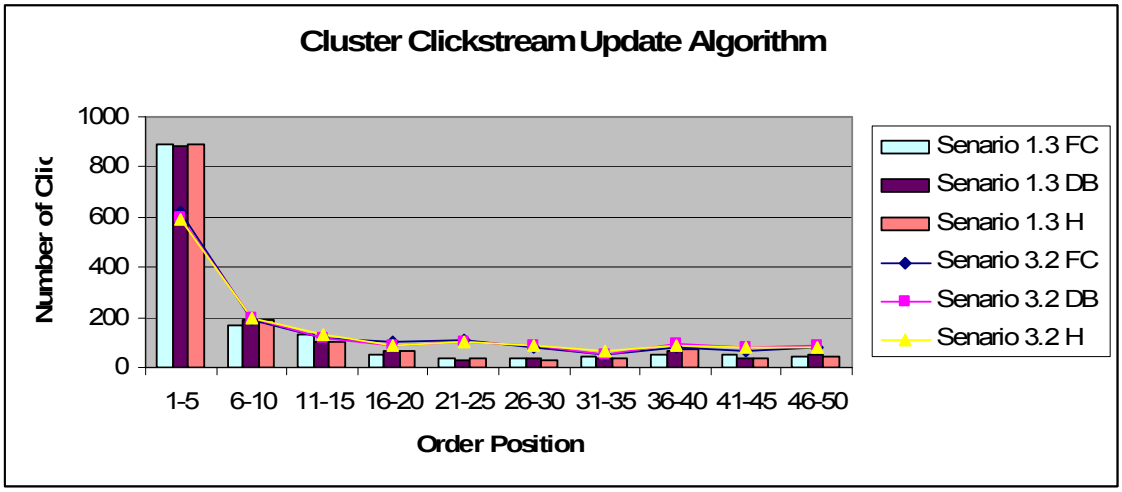
Αποτελέσματα στην Επιλογή Στιγμιοτύπων Υπηρεσίας

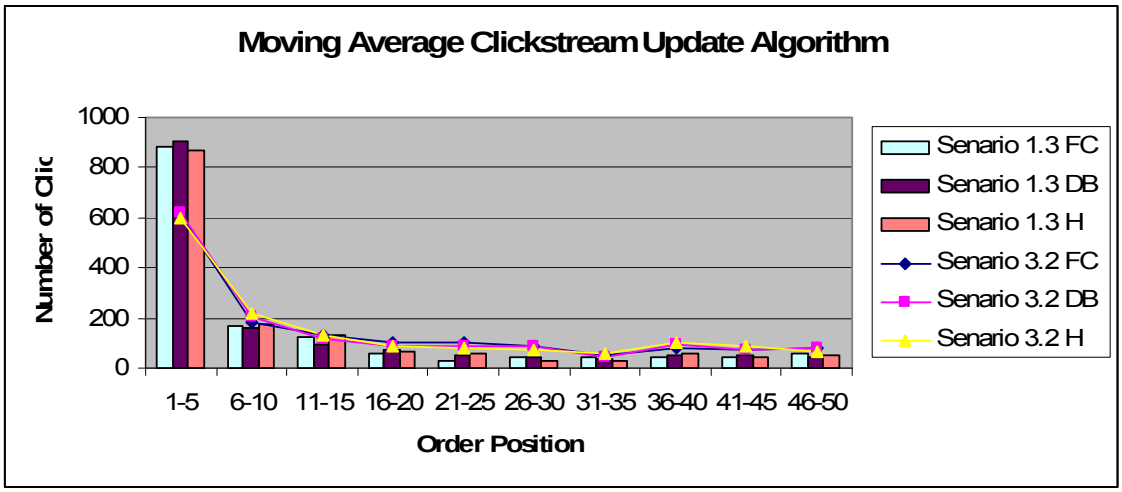
Στο πίνακα που ακολουθεί παρουσιάζονται τα αποτελέσματα στην επιλογή στιγμιοτύπου για κάθε αλγόριθμο και κάθε παραλλαγή του. Στο σύστημα υπάρχουν 50 στιγμιότυπα για κάθε υπηρεσία και κατ' επέκταση 50 πιθανές θέσεις για να παρουσιαστούν τα αποτελέσματα. Στη συνέχεια παρουσιάζονται οι γραφικές παραστάσεις των αποτελεσμάτων, μια για κάθε αλγόριθμο. Κάθε γραφική εμφανίζει τα αποτελέσματα για όλες τις παραλλαγές του αλγορίθμου και παράλληλα τα συγκρίνει με τα αποτελέσματα της τρίτης εκτέλεσης του σεναρίου 1, η οποία μας δίνουν την εκτέλεση με τα καλύτερα αποτελέσματα.

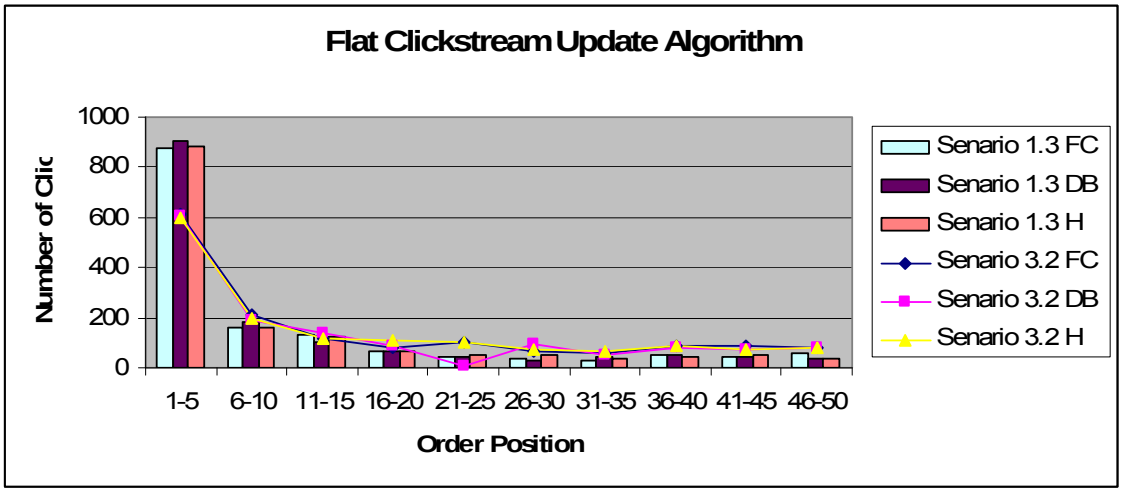
Θέση Εμφάνισης	**Cluster Clickstream Update Algorithm**			**Moving Average Clickstream Update Algorithm**			**Flat Clickstream Update Algorithm**		
	Time Zone Algorithm								
	FC	**DB**	**H**	**FC**	**DB**	**H**	**FC**	**DB**	**H**
1-5	621	600	588	615	620	596	615	605	595
6-10	189	198	197	186	202	216	210	189	200
11-15	118	115	128	129	116	131	118	137	114
16-20	101	85	84	104	88	89	80	89	107
21-25	110	103	101	99	87	83	105	9	99
26-30	82	86	87	86	89	74	63	96	76
31-35	53	54	63	50	46	55	59	50	66
36-40	78	95	90	78	94	99	85	82	90
41-45	69	78	80	73	76	88	86	75	74
46-50	79	86	82	80	82	69	79	80	79

Διαφορά Αποτελεσμάτων από την Τρίτη Εκτέλεση του Σεναρίου 1

Θέση Εμφάνισης	**Cluster Clickstream Update Algorithm**			**Moving Average Clickstream Update Algorithm**			**Flat Clickstream Update Algorithm**		
	Time Zone Algorithm								
	FC	**DB**	**H**	**FC**	**DB**	**H**	**FC**	**DB**	**H**
1-5	-268	-285	-306	-266	-285	-275	-260	-299	-286
6-10	22	10	6	16	45	44	51	6	43
11-15	-10	14	27	7	19	2	-17	34	-13
16-20	51	18	19	43	18	25	12	25	42
21-25	73	73	66	67	37	28	60	-33	46
26-30	46	52	61	41	46	43	29	70	26
31-35	8	18	28	7	12	24	29	6	26
36-40	24	28	20	32	45	44	32	28	46
41-45	18	38	42	29	27	46	41	34	24
46-50	36	34	37	24	36	19	23	41	46







Όπως φαίνεται από τα πιο πάνω αποτελέσματα η τρέχουσα κατάσταση και δραστηριότητα του χρήστη αλλάζει και τα ενδιαφέροντά του τόσο σε ότι αφορά τις υπηρεσίες που τον ενδιαφέρουν, όσο και σε ότι αφορά τον τύπο της υπηρεσίας που τον ενδιαφέρει.

8.1.3.3 Χρήση Λανθασμένης Χρονικής Περιόδου και Experience (Σενάριο 3-3)

Στο σενάριο αυτό συνδυάζουμε τα δύο προηγούμενα.

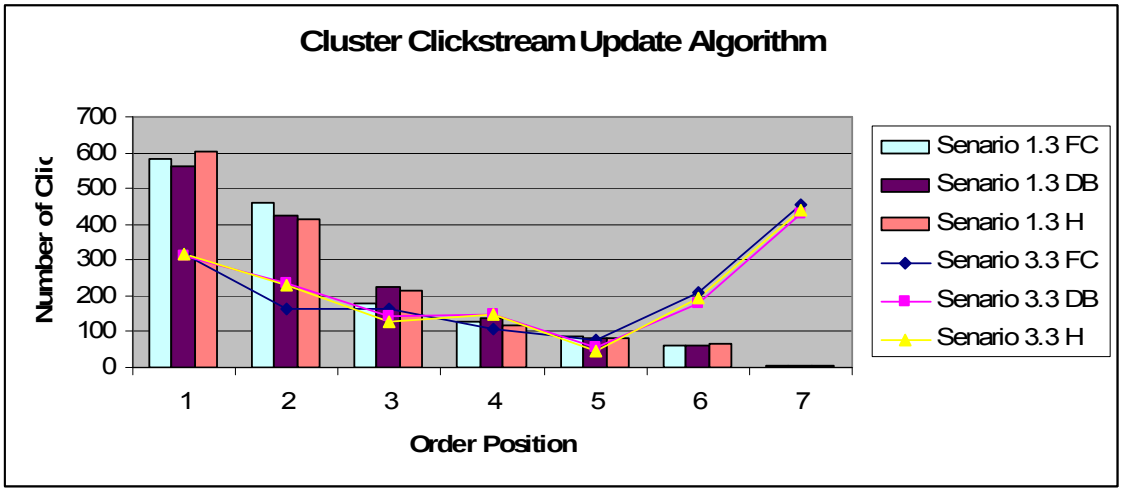
Αποτελέσματα στην Επιλογή Υπηρεσίας

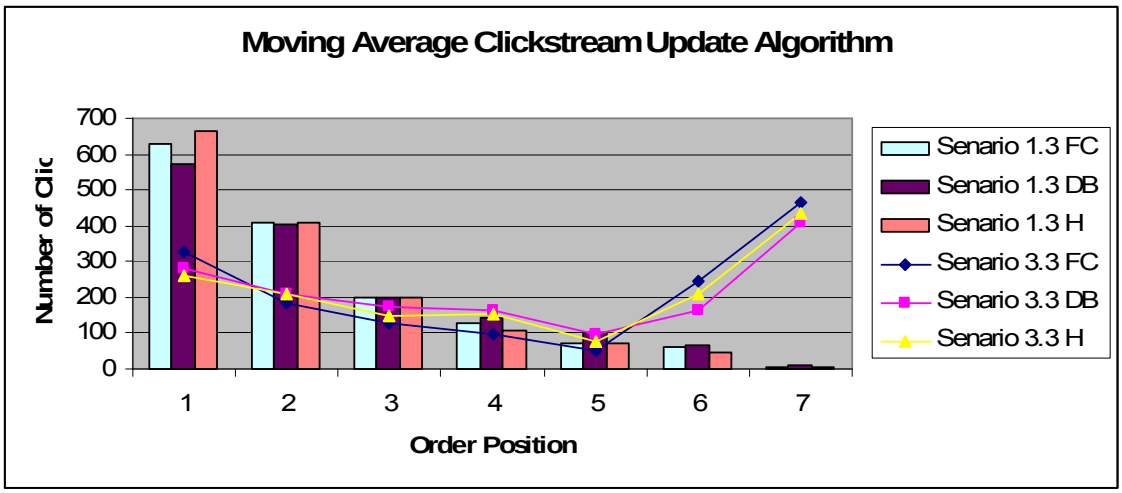
Στο πίνακα που ακολουθεί παρουσιάζονται τα αποτελέσματα στην επιλογή υπηρεσίας για κάθε αλγόριθμο και κάθε παραλλαγή του. Στο σύστημα υπάρχουν 7 υπηρεσίες και κατ' επέκταση 7 πιθανές θέσεις για να παρουσιαστούν. Στη συνέχεια παρουσιάζονται οι γραφικές παραστάσεις των αποτελεσμάτων, μια για κάθε αλγόριθμο. Κάθε γραφική εμφανίζει τα αποτελέσματα για όλες τις παραλλαγές του αλγορίθμου και παράλληλα τα συγκρίνει με τα αποτελέσματα της τρίτης εκτέλεσης του σεναρίου 1, η οποία μας δίνουν την εκτέλεση με τα καλύτερα αποτελέσματα.

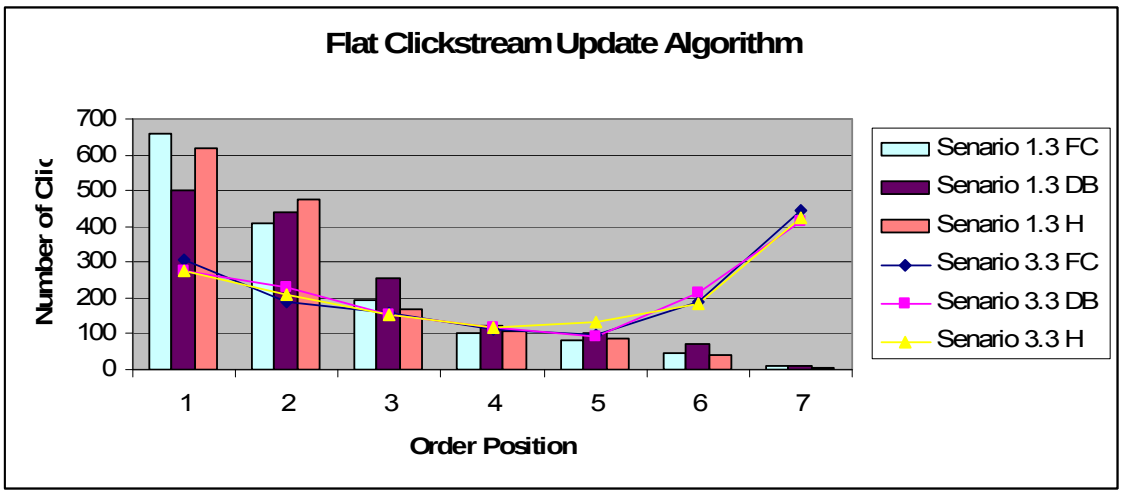
Θέση Εμφάνισης	**Cluster Clickstream Update Algorithm**			**Moving Average Clickstream Update Algorithm**			**Flat Clickstream Update Algorithm**		
	Time Zone Algorithm								
	FC	**DB**	**H**	**FC**	**DB**	**H**	**FC**	**DB**	**H**
1	315	314	317	329	282	261	309	278	276
2	165	236	230	183	207	212	188	228	212
3	164	141	127	129	175	150	160	155	155
4	109	148	150	98	166	155	113	115	118
5	79	54	47	53	96	76	99	93	132
6	212	178	192	244	165	211	187	216	184
7	456	429	437	464	409	435	444	415	423

Διαφορά Αποτελεσμάτων από την Τρίτη Εκτέλεση του Σεναρίου 1

Θέση Εμφάνισης	**Cluster Clickstream Update Algorithm**			**Moving Average Clickstream Update Algorithm**			**Flat Clickstream Update Algorithm**		
	Time Zone Algorithm								
	FC	**DB**	**H**	**FC**	**DB**	**H**	**FC**	**DB**	**H**
1	-269	-248	-286	-301	-292	-402	-351	-223	-343
2	-293	-187	-182	-226	-198	-195	-220	-210	-262
3	-17	-86	-87	-68	-26	-48	-34	-100	-12
4	-17	11	31	-31	23	49	10	-9	10
5	-8	-27	-35	-18	-6	4	18	-9	45
6	152	115	127	185	100	164	142	145	143
7	452	422	432	459	399	428	435	406	419







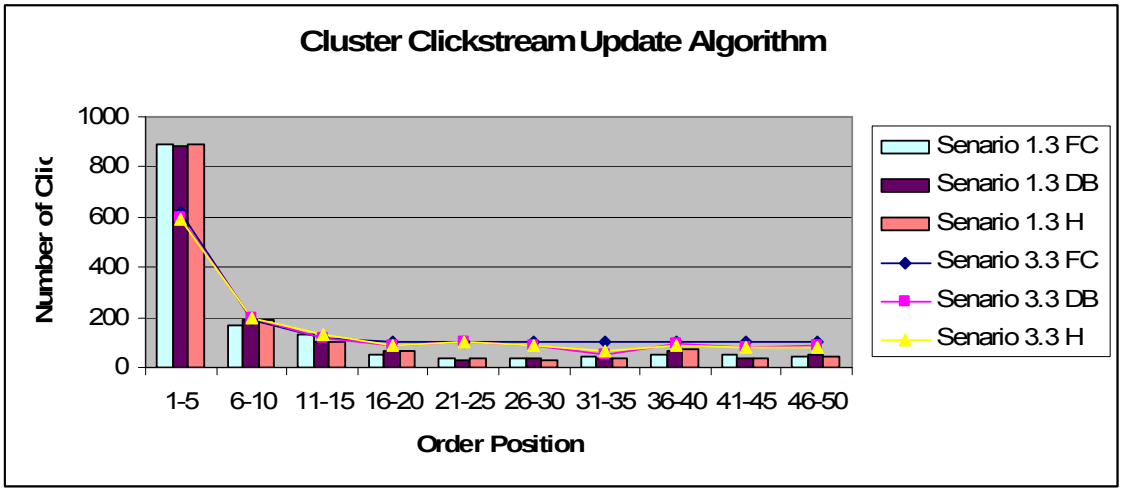
Αποτελέσματα στην Επιλογή Στιγμιοτύπων Υπηρεσίας

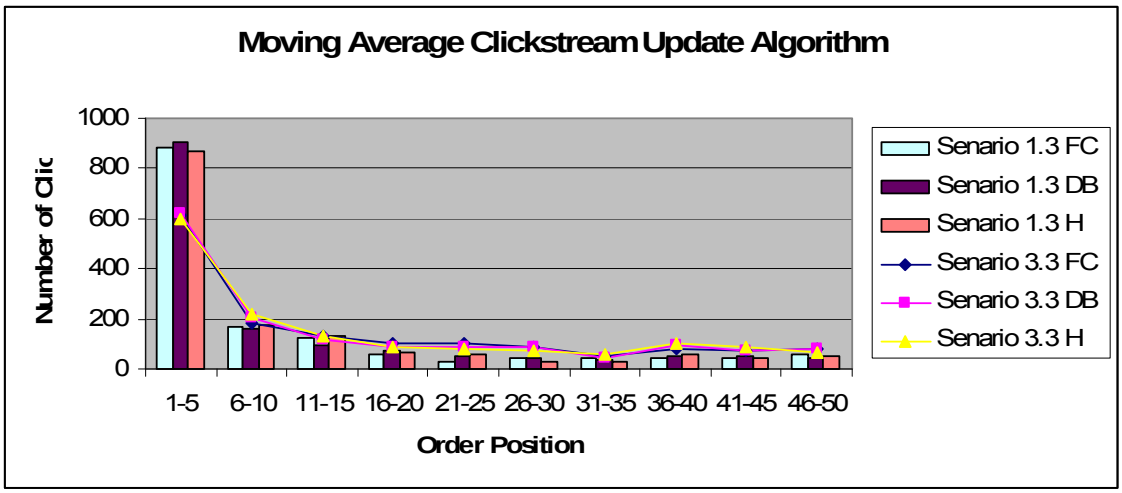
Στο πίνακα που ακολουθεί παρουσιάζονται τα αποτελέσματα στην επιλογή στιγμιοτύπου για κάθε αλγόριθμο και κάθε παραλλαγή του. Στο σύστημα υπάρχουν 50 στιγμιότυπα για κάθε υπηρεσία και κατ' επέκταση 50 πιθανές θέσεις για να παρουσιαστούν τα αποτελέσματα. Στη συνέχεια παρουσιάζονται οι γραφικές παραστάσεις των αποτελεσμάτων, μια για κάθε αλγόριθμο. Κάθε γραφική εμφανίζει τα αποτελέσματα για όλες τις παραλλαγές του αλγορίθμου και παράλληλα τα συγκρίνει με τα αποτελέσματα της τρίτης εκτέλεσης του σεναρίου 1, η οποία μας δίνουν την εκτέλεση με τα καλύτερα αποτελέσματα.

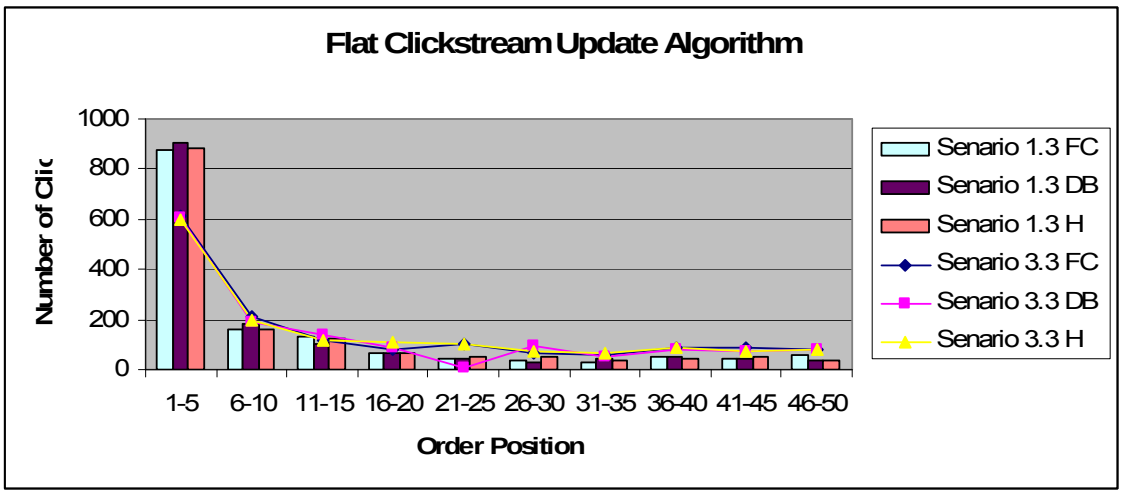
Θέση Εμφάνισης	**Cluster Clickstream Update Algorithm**			**Moving Average Clickstream Update Algorithm**			**Flat Clickstream Update Algorithm**		
	Time Zone Algorithm								
	FC	**DB**	**H**	**FC**	**DB**	**H**	**FC**	**DB**	**H**
1-5	621	600	588	615	620	596	615	605	595
6-10	189	198	197	186	202	216	210	189	200
11-15	118	115	128	129	116	131	118	137	114
16-20	101	85	84	104	88	89	80	89	107
21-25	110	103	101	99	87	83	105	9	99
26-30	82	86	87	86	89	74	63	96	76
31-35	53	54	63	50	46	55	59	50	66
36-40	78	95	90	78	94	99	85	82	90
41-45	69	78	80	73	76	88	86	75	74
46-50	79	86	82	80	82	69	79	80	79

Διαφορά Αποτελεσμάτων από την Τρίτη Εκτέλεση του Σεναρίου 1

Θέση Εμφάνισης	**Cluster Clickstream Update Algorithm**			**Moving Average Clickstream Update Algorithm**			**Flat Clickstream Update Algorithm**		
	Time Zone Algorithm								
	FC	**DB**	**H**	**FC**	**DB**	**H**	**FC**	**DB**	**H**
1-5	-268	-285	-306	-266	-285	-275	-260	-299	-286
6-10	22	10	6	16	45	44	51	6	43
11-15	-10	14	27	7	19	2	-17	34	-13
16-20	51	18	19	43	18	25	12	25	42
21-25	73	73	66	67	37	28	60	-33	46
26-30	46	52	61	41	46	43	29	70	26
31-35	8	18	28	7	12	24	29	6	26
36-40	24	28	20	32	45	44	32	28	46
41-45	18	38	42	29	27	46	41	34	24
46-50	36	34	37	24	36	19	23	41	46


Cluster Clickstream Update Algorithm
Number of Clic
1000
800
600
400
200
0
1-5
6-10
11-15
16-20
21-25
26-30
31-35
36-40
41-45
46-50
Order Position
Senario 1.3 FC
Senario 1.3 DB
Senario 1.3 H
Senario 3.3 FC
Senario 3.3 DB
Senario 3.3 H


Moving Average Clickstream Update Algorithm
Number of Clic
1000
800
600
400
200
0
1-5
6-10
11-15
16-20
21-25
26-30
31-35
36-40
41-45
46-50
Order Position
Senario 1.3 FC
Senario 1.3 DB
Senario 1.3 H
Senario 3.3 FC
Senario 3.3 DB
Senario 3.3 H


Flat Clickstream Update Algorithm
Number of Clic
1000
800
600
400
200
0
1-5
6-10
11-15
16-20
21-25
26-30
31-35
36-40
41-45
46-50
Order Position
Senario 1.3 FC
Senario 1.3 DB
Senario 1.3 H
Senario 3.3 FC
Senario 3.3 DB
Senario 3.3 H

Το πιο πάνω σενάριο επαληθεύει τη σημαντικότητα του χρόνου και της εμπειρίας του χρήστη σαν δύο σημαντικούς παράγοντες που πρέπει να λαμβάνονται υπόψη κατά την εξατομίκευση πληροφοριών που αναφέρονται σε κινητούς χρήστες.

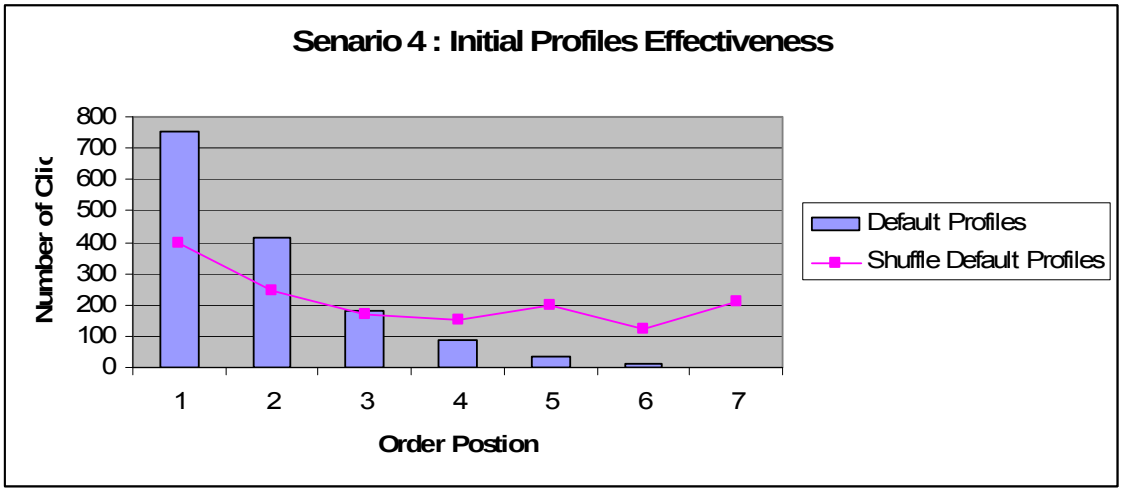
8.1.4 Αποτελέσματα και Σημαντικότητα Αρχικών Προφίλ (Σενάριο 4)

Στο σενάριο αυτό μελετούμε τη σημαντικότητα των αρχικών προφίλ. Αρχικά βλέπουμε τα αποτελέσματα του συστήματος χρησιμοποιώντας τα αρχικά προφίλ και στη συνέχεια τα συγκρίνουμε με τα αποτελέσματα που παίρνουμε από το σύστημα αν δώσουμε λανθασμένα αρχικά προφίλ στους χρήστες.

Αποτελέσματα στην Επιλογή Υπηρεσίας

Στο πίνακα που ακολουθεί παρουσιάζονται τα αποτελέσματα στην επιλογή υπηρεσίας για όλους τους αλγορίθμους. Στο σύστημα υπάρχουν 7 υπηρεσίες και κατ' επέκταση 7 πιθανές θέσεις για να παρουσιαστούν. Στη συνέχεια παρουσιάζεται η γραφική παράσταση των αποτελεσμάτων. Η γραφική εμφανίζει τα αποτελέσματα και τα συγκρίνει με τα αποτελέσματα που παίρνουμε εάν τρέξουμε τους αλγορίθμους πάνω στα Αρχικά Προφίλ.

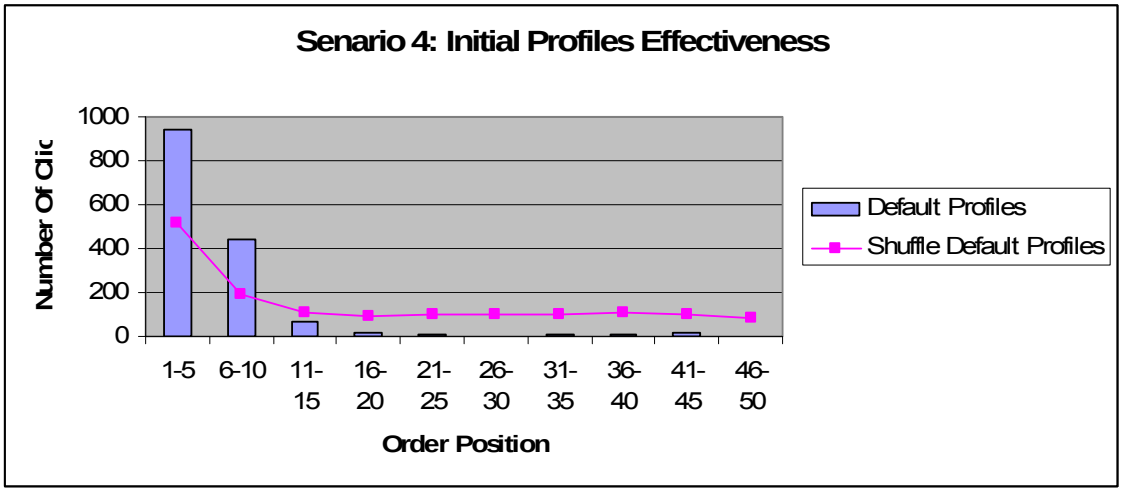
Θέση Εμφάνισης	Default Profiles	Shuffle Default Profiles
1	753	398
2	416	247
3	182	168
4	88	154
5	34	198
6	13	124
7	0	211



Αποτελέσματα στην Επιλογή Στιγμιοτύπων Υπηρεσίας

Στο πίνακα που ακολουθεί παρουσιάζονται τα αποτελέσματα στην επιλογή υπηρεσίας για όλους τους αλγορίθμους. Στο σύστημα υπάρχουν 50 στιγμιότυπα σε κάθε υπηρεσία και κατ' επέκταση 50 πιθανές θέσεις για να παρουσιαστούν τα αποτελέσματα. Στη συνέχεια παρουσιάζεται η γραφική παράσταση των αποτελεσμάτων. Η γραφική εμφανίζει τα αποτελέσματα και τα συγκρίνει με τα αποτελέσματα που παίρνουμε εάν τρέξουμε τους αλγορίθμους πάνω στα Αρχικά Προφίλ.

Θέση Εμφάνισης	**Default Profiles**	**Shuffle Default Profiles**
1-5	939	514
6-10	438	188
11-15	69	109
16-20	16	95
21-25	5	102
26-30	3	98
31-35	8	100
36-40	5	107
41-45	13	103
46-50	3	84



Στα πιο πάνω αποτελέσματα, παρατηρούμε να μειώνεται η απόδοση του συστήματος, εάν δοθούν λανθασμένα αρχικά προφίλ στους χρήστες. Η σωστή ανάθεση των αρχικών προφίλ στους χρήστες είναι κατ' επέκταση μια διαδικασία πολύ σημαντική.

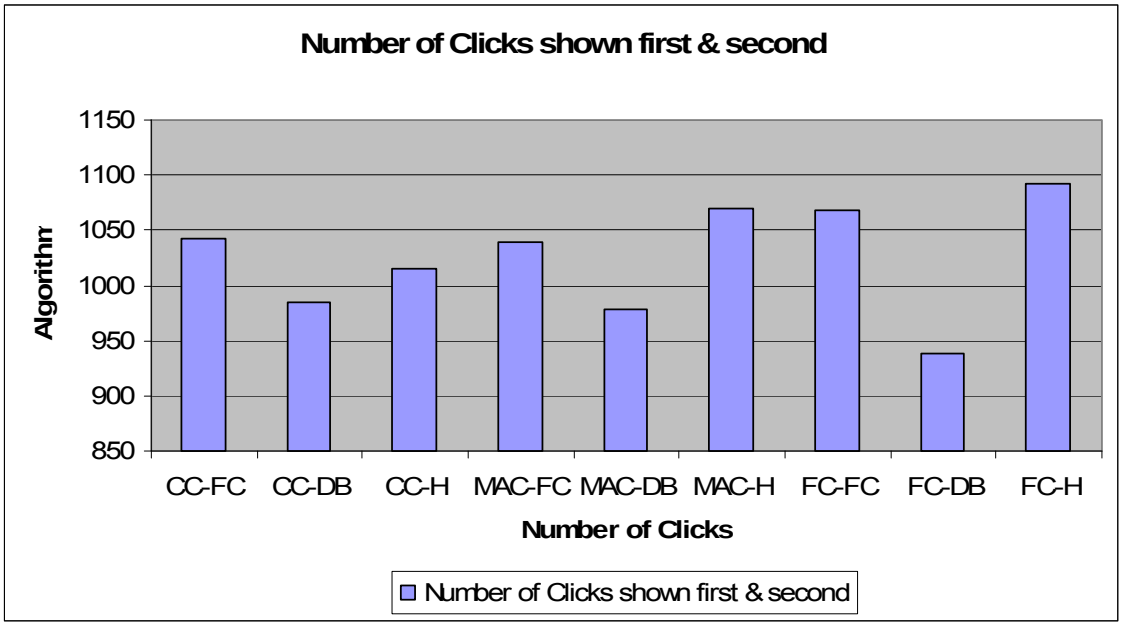
8.2 Αποτελεσματικότητα Αλγορίθμων

Στη παράγραφο αυτή χρησιμοποιούμε τις μετρικές που ορίστηκαν στο υποκεφάλαιο 7.4 για να μπορέσουμε να εξάγουμε πιο περιεκτικά αποτελέσματα με την βοήθεια των μετρικών αυτών.

8.2.1 Μέτρο Επιτυχίας Αλγορίθμου (Order Position)

Με τη μετρική αυτή βλέπουμε συνοπτικά την θέση στην οποία παρουσιάστηκαν τα αποτελέσματα στο χρήστη. Έτσι μπορούμε να συμπεράνουμε αλγορίθμους που δίνουν καλύτερα αποτελέσματα σε ότι αφορά την θέση που εμφανίστηκε η επιθυμητή υπηρεσία ή στιγμιότυπο.

Για την συνοπτική περιγραφή της αποτελεσματικότητας των αλγορίθμων, χρησιμοποιούνται τα αποτελέσματα που πήραμε από το σύστημα στην τρίτη εκτέλεση του σεναρίου 1. Στις γραφικές που ακολουθούν αναπαριστώνται τα αποτελέσματα αυτά και η σημασία τους. Στη συνέχεια συγκρίνουμε τους αλγορίθμους αυτούς με βάση τις μερικές Mean Absolute Error και Μέτρο Αποτελεσματικότητας που περιγράψαμε στα υποκεφάλαια 7.4.1 και 7.4.2.



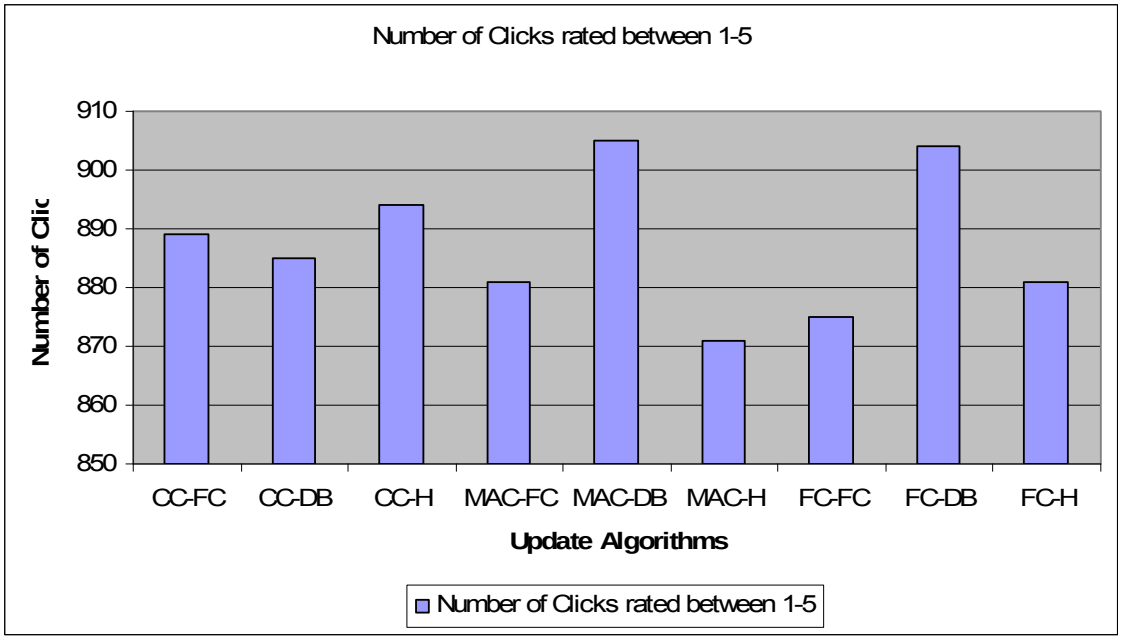
Σχήμα 8-1 : Υπηρεσίες που εμφανίζονται στην 1η και 2η Θέση από το σύστημα

Ο αλγόριθμος Flat Clickstream συνδυασμένος με τον αλγόριθμο Histogram για τις χρονικές περιόδους, φαίνεται να μας δίνει τα καλύτερα αποτελέσματα σε ότι αφορά την εμφάνιση των υπηρεσιών στο χρήστη για επιλογή της επιθυμητής υπηρεσία. Ωστόσο καλά αποτελέσματα φαίνεται να δίνουν και οι άλλοι δύο αλγόριθμοι αρκεί να μην συνδυαστούν με τον αλγόριθμο density based για την προσαρμογή των χρονικών περιόδων.

Ο συνδυασμός οποιουδήποτε αλγορίθμου με τον αλγόριθμο density based φαίνεται να μας δίνει τα χειρότερα αποτελέσματα. Αυτό μπορεί να δικαιολογηθεί εάν λάβουμε υπόψη ότι ο αλγόριθμος αυτός διασπά το χρόνο σε πιο πολλές χρονικές περιόδους απ' ότι οι άλλοι δύο αλγόριθμοι. Με το τρόπο αυτό αποτυγχάνει ανακαλύψει τις χρονικές περιόδους στις οποίες τείνει να αναζητεί ο χρήστης μια συγκεκριμένη υπηρεσία. Ως αποτέλεσμα αποτυγχάνει να αναπαραστήσει τα πραγματικά ποσοστά προτίμησης του χρήστη σε μια χρονική περίοδο.

Επιπλέον ο αλγόριθμος Histogram, φαίνεται να δίνει καλύτερα αποτελέσματα από τους άλλους δύο αλγορίθμους για προσαρμογή των χρονικών περιόδων, σε ότι αφορά την εμφάνιση υπηρεσιών στο χρήστη για επιλογή. Ο αλγόριθμος αυτό προσαρμόζει τις χρονικές περιόδους σε όσο το δυνατό λιγότερες ανάλογα με τις επιλογές του χρήστη στις αντίστοιχες περιόδους. Κατ' επέκταση στην εμφάνιση των υπηρεσιών του χρήστη, είναι καλύτερο να έχουμε όσο το δυνατό πιο μαζεμένες τις χρονικές περιόδους για να μπορούν να έχουν πιο συγκεντρωτικά ποσοστά

προτίμησης για κάθε υπηρεσία ώστε να αναπαριστούν καλύτερα την συμπεριφορά του χρήστη.



Σχήμα 8-2 : Στιγμιότυπα υπηρεσιών που εμφανίζονται στις πρώτες πέντε θέσεις, από το σύστημα

Σε ότι αφορά την εμφάνιση των στιγμιοτύπων των υπηρεσιών, ο αλγόριθμος Moving Average σε συνδυασμό με τον αλγόριθμο Density Based για την προσαρμογή των χρονικών περιόδων, φαίνεται να δίνει τα καλύτερα αποτελέσματα.

Πολύ καλά αποτελέσματα δίνει επίσης ο αλγόριθμος Cluster Clickstream ανεξάρτητα με ποιό αλγόριθμο προσαρμογής των χρονικών περιόδων χρησιμοποιείται.

Και οι δύο πιο πάνω αλγόριθμοι τείνουν να προβλέπουν τις επιλογές του χρήστη, και να δίνουν προτεραιότητα στα χαρακτηριστικά και υπηρεσίες που ο χρήστης φαίνεται να προτιμά περισσότερο.

Επιπλέον, ο αλγόριθμος προσαρμογής χρονικών περιόδων, Density Based, φαίνεται να δίνει πολύ καλά αποτελέσματα στην περίπτωση των στιγμιοτύπων των υπηρεσιών. Ο αλγόριθμος αυτός διαχωρίζει σε περισσότερες περιόδους το 24ωρο. Αυτό μας οδηγεί στο συμπέρασμα ότι για το χαρακτηριστικό Type_By_Time είναι καλύτερα να χρησιμοποιείται ένας αλγόριθμος που διασπά σε μικρότερα κομμάτια τις χρονικές περιόδους. Έτσι μπορεί καλύτερα να διαχωρίσει τους τύπους των

υπηρεσιών στις χρονικές περιόδους δίνοντας πιο αξιόπιστα αποτελέσματα. Από την μελέτη των αποτελεσμάτων φαίνεται ότι ο τύπος μιας υπηρεσίας που προτιμάται από τον χρήστη μπορεί να διαφέρει ακόμη σε γειτονικές χρονικές περιόδους. Έχοντας μικρότερες χρονικές περιόδους μπορούμε να συλλάβουμε πιο αξιόπιστα την συμπεριφορά αυτή.

8.2.2 Mean Absolute Error

Με την μετρική αυτή προσπαθούμε να βρούμε την μέση τιμή σφάλματος που δίνει κάθε αλγόριθμος στην επιλογή υπηρεσίας και στιγμιοτύπου.

Για την μετρική αυτή θεωρούμε σαν επιθυμητή τιμή τη θέση 2 για τις υπηρεσίες και την θέση 5 για τα στιγμιότυπα. Για όσα αποτελέσματα εμφανίστηκαν σε θέση μικρότερη του 2 και του 5 για τις υπηρεσίες και τα στιγμιότυπα αντίστοιχα, χρησιμοποιούμε σαν επιθυμητή θέση την θέση που εμφανίστηκαν.

Αποτελέσματα στην Επιλογή Υπηρεσίας

Πιο κάτω παρουσιάζονται τα αποτελέσματα της τρίτης εκτέλεσης του σεναρίου 1 τα οποία χρησιμοποιούνται για την μετρική αυτή.

Θέση Εμφάνισης	Cluster Clickstream Update Algorithm			Moving Average Clickstream Update Algorithm			Flat Clickstream Update Algorithm		
	Time Zone Algorithm								
	FC	DB	H	FC	DB	H	FC	DB	H
1	584	562	603	630	574	663	660	501	619
2	458	423	412	409	405	407	408	438	474
3	181	227	214	197	201	198	194	255	167
4	126	137	119	129	143	106	103	124	108
5	87	81	82	71	102	72	81	102	87
6	60	63	65	59	65	47	45	71	41
7	4	7	5	5	10	7	9	9	4

Mean Absolute Error

Υπολογίζουμε το Mean Absolute Error για κάθε αλγόριθμο και κάθε παραλλαγή του. Τα αποτελέσματα δίνονται στον πίνακα που ακολουθεί και επιβεβαιώνουν τα αποτελέσματα που αναφέραμε στο προηγούμενο υποκεφάλαιο.

Αλγόριθμος	Time Zone Algorithm		
	FC	DB	H
Cluster Clickstream Update Algorithm	0,636	0,687	0,655
Moving Average Clickstream Update Algorithm	0,619	0,735	0,566
Flat Clickstream Update Algorithm	0,578	0,758	0,552

Από τα πιο πάνω αποτελέσματα επιβεβαιώνουμε ότι ο αλγόριθμος Density Based για προσαρμογή των χρονικών περιόδων δίνει τα χειρότερα αποτελέσματα. Όπως ήδη αναφέραμε ο αλγόριθμος αυτός διασπά τις χρονικές περιόδους ώστε να έχουμε πιο πολλές σε σχέση με τους άλλους δύο. Άρα επαληθεύεται ότι κατά την επιλογή υπηρεσίας απαιτείται να έχουμε όσο το δυνατό πιο μαζεμένα χρονικά διαστήματα.

Επιπλέον από τα πιο πάνω φαίνεται καθαρά ότι αλγόριθμος με το λιγότερο σφάλμα είναι ο Flat Clickstream σε συνδυασμό με τον Histogram και κατ' επέκταση είναι αυτός που δίνει τα καλύτερα αποτελέσματα κατά την επιλογή υπηρεσίας.

Αποτελέσματα στην Επιλογή Στιγμιοτύπου Υπηρεσίας

Πιο κάτω παρουσιάζονται τα αποτελέσματα της τρίτης εκτέλεσης του σεναρίου 1 τα οποία χρησιμοποιούνται για την μετρική αυτή.

Θέση Εμφάνισης	Cluster Clickstream Update Algorithm			Moving Average Clickstream Update Algorithm			Flat Clickstream Update Algorithm		
	Time Zone Algorithm								
	FC	DB	H	FC	DB	H	FC	DB	H
1-5	889	885	894	881	905	871	875	904	881
6-10	167	188	191	170	157	172	159	183	157
11-15	128	101	101	122	97	129	135	103	127
16-20	50	67	65	61	70	64	68	64	65
21-25	37	30	35	32	50	55	45	42	53
26-30	36	34	26	45	43	31	34	26	50
31-35	45	36	35	43	34	31	30	44	40
36-40	54	67	70	46	49	55	53	54	44
41-45	51	40	38	44	49	42	45	41	50
46-50	43	52	45	56	46	50	56	39	33

Mean Absolute Error

Υπολογίζουμε το Mean Absolute Error για κάθε αλγόριθμο και κάθε παραλλαγή του. Τα αποτελέσματα δίνονται στον πίνακα που ακολουθεί και επιβεβαιώνουν τα αποτελέσματα που αναφέραμε στο προηγούμενο υποκεφάλαιο.

Αλγόριθμος	**Time Zone Algorithm**		
	FC	**DB**	**H**
Cluster Clickstream Update Algorithm	7,813	7,846	7,556
Moving Average Clickstream Update Algorithm	7,953	7,763	7,846
Flat Clickstream Update Algorithm	7,993	7,333	7,71

Τα αποτελέσματα της μετρικής αυτής μας δίνουν σαν καλύτερο αλγόριθμο τον Flat Clickstream συνδυασμένο με τον Density Based. Γενικά η παραλλαγή των αλγορίθμων με τον Density Based φαίνεται να δίνει τα καλύτερα αποτελέσματα όπως ακριβώς συμπεράναμε και στο υποκεφάλαιο 8.2.1.1.

8.2.3 Μέτρο αποτελεσματικότητας

Εδώ χρησιμοποιούμε την μετρική που παρουσιάστηκε στο υποκεφάλαιο 7.4.2. Με το Μέτρο αποτελεσματικότητας προσπαθούμε να εξάγουμε την αποτελεσματικότητα των αλγορίθμων σε ότι αφορά την επιλογή υπηρεσίας και στιγμιοτύπου.

Αποτελέσματα στην Επιλογή Υπηρεσίας

Πιο κάτω παρουσιάζονται τα αποτελέσματα της τρίτης εκτέλεσης του σεναρίου 1 τα οποία χρησιμοποιούνται για την μετρική αυτή.

Θέση Εμφάνισης	**Cluster Clickstream Update Algorithm**			**Moving Average Clickstream Update Algorithm**			**Flat Clickstream Update Algorithm**		
	Time Zone Algorithm								
	FC	**DB**	**H**	**FC**	**DB**	**H**	**FC**	**DB**	**H**
1	584	562	603	630	574	663	660	501	619
2	458	423	412	409	405	407	408	438	474
3	181	227	214	197	201	198	194	255	167
4	126	137	119	129	143	106	103	124	108
5	87	81	82	71	102	72	81	102	87
6	60	63	65	59	65	47	45	71	41
7	4	7	5	5	10	7	9	9	4

Μέτρο Αποτελεσματικότητας

Υπολογίζουμε Μέτρο αποτελεσματικότητας για κάθε αλγόριθμο και κάθε παραλλαγή του. Τα αποτελέσματα δίνονται στον πίνακα που ακολουθεί και επιβεβαιώνουν τα αποτελέσματα που αναφέραμε στο προηγούμενο υποκεφάλαιο. Τα αποτελέσματα της μετρικής αυτής μας δείχνουν πόσο αποτελεσματικός είναι ο κάθε αλγόριθμος.

Αλγόριθμος	**Time Zone Algorithm**		
	FC	**DB**	**H**
Cluster Clickstream Update Algorithm	62 %	60%	62%
Moving Average Clickstream Update Algorithm	63%	60%	65%
Flat Clickstream Update Algorithm	65%	57%	64%

Η μετρική αυτή επαληθεύει τις άλλες δύο. Κι από αυτά τα αποτελέσματα φαίνεται καλύτερος αλγόριθμος να είναι ο Flat Clickstream με ποσοστά αποτελεσματικότητας 64% και 65%. Επιπλέον χειρότερος από τους αλγορίθμους προσαρμογής των χρονικών περιόδων και πάλι φαίνεται να είναι ο Density Based με ποσοστά αποτελεσματικότητας 57% και 60%.

Αποτελέσματα στην Επιλογή Στιγμιοτύπου Υπηρεσίας

Πιο κάτω παρουσιάζονται τα αποτελέσματα της τρίτης εκτέλεσης του σεναρίου 1 τα οποία χρησιμοποιούνται για την μετρική αυτή.

Θέση Εμφάνισης	**Cluster Clickstream Update Algorithm**			**Moving Average Clickstream Update Algorithm**			**Flat Clickstream Update Algorithm**		
	Time Zone Algorithm ψ								
	FC	**DB**	**H**	**FC**	**DB**	**H**	**FC**	**DB**	**H**
1-5	889	885	894	881	905	871	875	904	881
6-10	167	188	191	170	157	172	159	183	157
11-15	128	101	101	122	97	129	135	103	127
16-20	50	67	65	61	70	64	68	64	65
21-25	37	30	35	32	50	55	45	42	53
26-30	36	34	26	45	43	31	34	26	50
31-35	45	36	35	43	34	31	30	44	40
36-40	54	67	70	46	49	55	53	54	44
41-45	51	40	38	44	49	42	45	41	50
46-50	43	52	45	56	46	50	56	39	33

Μέτρο αποτελεσματικότητας

Υπολογίζουμε το Μέτρο Αποτελεσματικότητας για κάθε αλγόριθμο και κάθε παραλλαγή του. Τα αποτελέσματα δίνονται στον πίνακα που ακολουθεί και επιβεβαιώνουν τα αποτελέσματα που αναφέραμε στο προηγούμενο υποκεφάλαιο.

Αλγόριθμος	**Time Zone Algorithm**		
	FC	**DB**	**H**
Cluster Clickstream Update Algorithm	63%	63%	63%
Moving Average Clickstream Update Algorithm	62%	63%	62%
Flat Clickstream Update Algorithm	62%	64%	62%

Με βάση τη μετρική αυτή Καλύτερος αλγόριθμος στην επιλογή στιγμιοτύπων φαίνεται να είναι ο Flat Clickstream συνδυασμένος με τον Density Based. Ο αλγόριθμος αυτό έχει την πιο ψηλή αποδοτικότητα με 64%. Επιπλέον και πάλι παρατηρούμε πως για την επιλογή στιγμιοτύπου καλύτερα αποτελέσματα δίνει ο συνδυασμός με τον αλγόριθμο προσαρμογής χρονικών περιόδων Density Based. Καλά αποτελέσματα δίνει επίσης και ο αλγόριθμος Cluster Clickstream με αποτελεσματικότητα 63%.

Κεφάλαιο 9

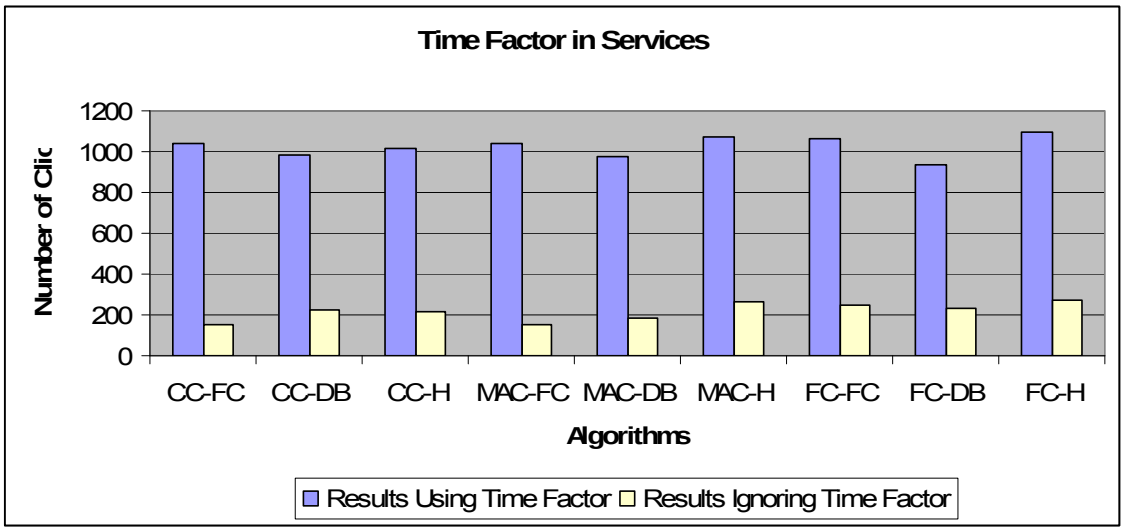
Συμπεράσματα και Μελλοντική εργασία.

Στην εργασία αυτή μελετήσαμε το πρόβλημα τις πληροφοριακής υπερφόρτωσης στο κινητό περιβάλλον σαν ένα διαφοροποιημένο πρόβλημα εξατομίκευσης. Η εξατομίκευση στο κινητό περιβάλλον επεκτείνεται και επαναπροσδιορίζεται έτσι ώστε να λαμβάνει υπόψη τρεις από τους βασικότερους παράγοντες που επηρεάζουν τις επιλογές του χρήστη. Οι παράγοντες αυτοί έχουν άμεση σχέση με τον χρόνο και τον τρόπο που αυτός αλλάζει την κατάσταση και τη θέση του. Έτσι μελετούμε την εξατομίκευση στο κινητό περιβάλλον λαμβάνοντας υπόψη την τοποθεσία του χρήστη, την χρονική στιγμή, και την κατάσταση στην οποία βρίσκεται ο χρήστη κατά την στιγμή της εξατομίκευσης.

Στη συνέχεια θα δούμε συνοπτικά συμπεράσματα που εξάγονται μελετώντας τα αποτελέσματα και τις μετρικές που παρουσιάσαμε στο προηγούμενο κεφάλαιο.

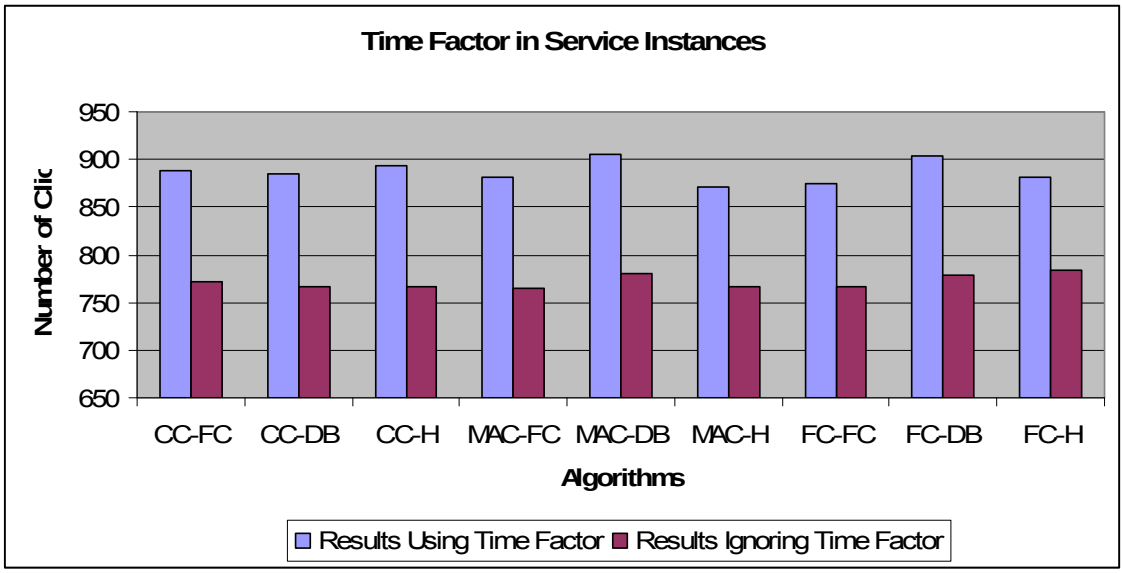
9.1 Συμπεράσματα

9.1.1 Η σημαντικότητα του παράγοντα χρόνου



Σχήμα 8-3: : Η σημαντικότητα του παράγοντα χρόνο στην επιλογή υπηρεσίας

Ο χρόνος είναι ένας πολύ σημαντικός παράγοντας που φαίνεται να επηρεάζει σημαντικά τις υπηρεσίες που αναζητεί ο κινητός χρήστης. Άγνοια του παράγοντα αυτού μπορεί να δώσει πάρα πολύ κακά αποτελέσματα σε ότι αφορά τις υπηρεσίες που προτείνει ένα εξατομικευμένο σύστημα στο χρήστη.



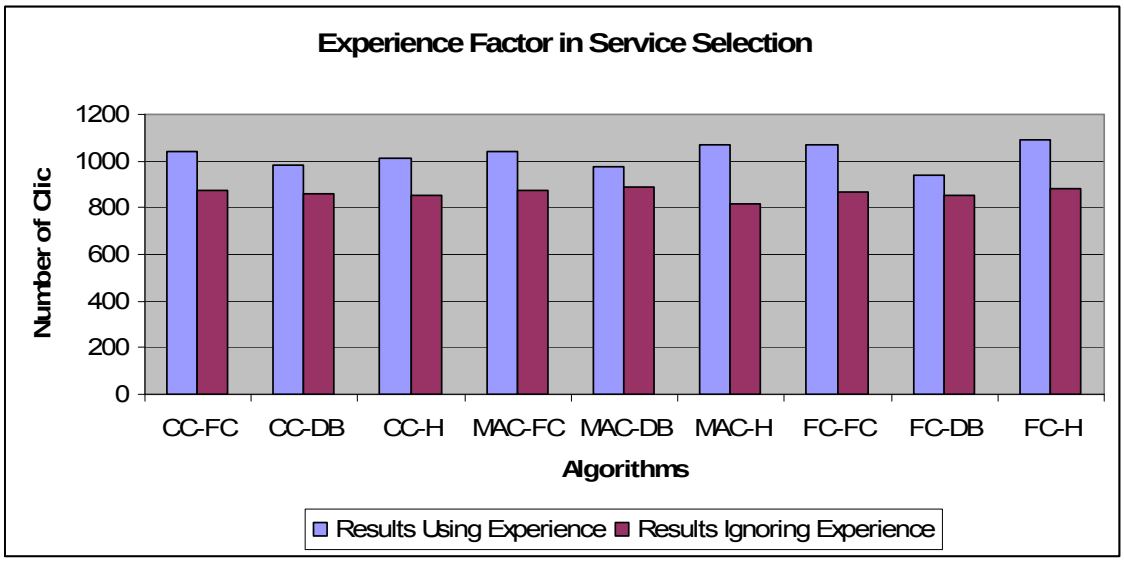
Σχήμα 8-4: Η σημαντικότητα του παράγοντα χρόνο στην επιλογή στιγμιότυπου υπηρεσίας

Ο χρόνος είναι ένας παράγοντας που επηρεάζει σημαντικά ακόμη και την επιλογή στιγμιοτύπου μιας υπηρεσίας. Στο κινητό περιβάλλον οι ανάγκες του χρήστη

αλλάζουν και σε επίπεδο χαρακτηριστικών υπηρεσίας με βάση το χρόνο. Ένας παράγοντας που αν ληφθεί υπόψη μπορεί να δώσει ακόμη καλύτερα αποτελέσματα.

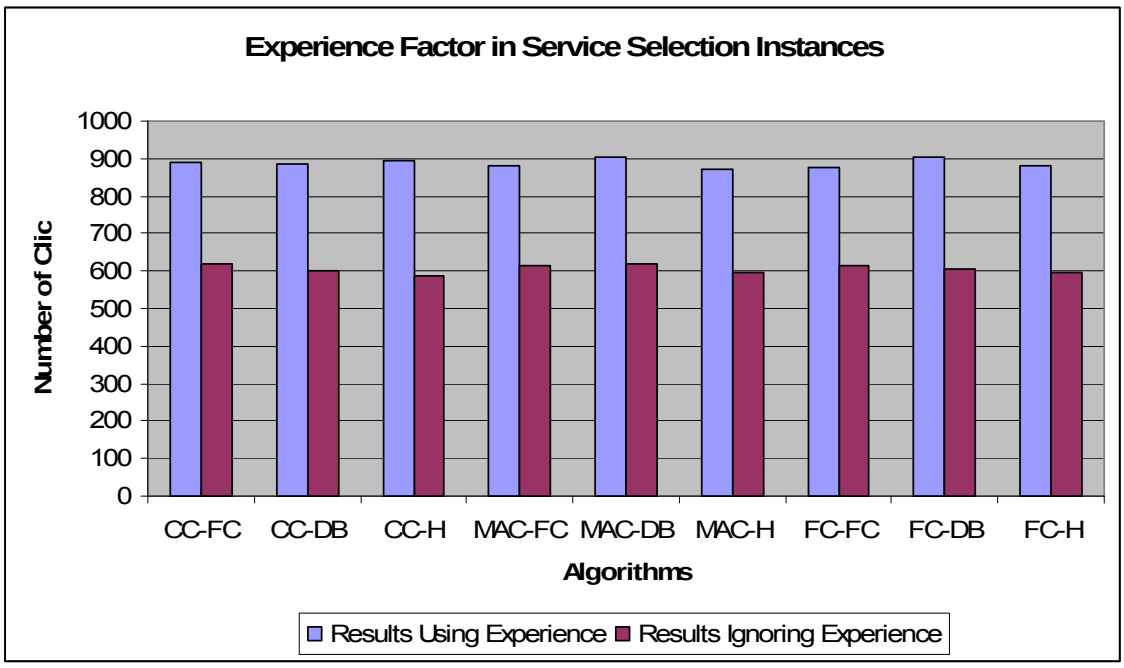
Από τα αποτελέσματα του προηγούμενου κεφαλαίου προκύπτει επίσης το συμπέρασμα ότι για καλύτερα αποτελέσματα θα πρέπει να χρησιμοποιούνται διαφορετικοί αλγόριθμοι κατά την επιλογή υπηρεσίας και στιγμιότυπου υπηρεσίας. Αλγόριθμοι που σπάζουν το χρόνο σε πιο πολλές χρονικές περιόδους είναι πιο αποδοτικοί κατά την επιλογή στιγμιότυπου υπηρεσίας ενώ το αντίθετο συμβαίνει κατά την επιλογή υπηρεσίας. Επιπλέον αλγόριθμοι που λαμβάνουν υπόψη την προηγούμενη τιμή προτιμήσεως ενός χρήστη στα χαρακτηριστικά, δίνουν καλύτερα αποτελέσματα στην επιλογή στιγμιότυπου υπηρεσίας.

9.1.2 Η σημαντικότητα του Experience



Σχήμα 8-5: Η σημαντικότητα του Experience στην Επιλογή Υπηρεσίας

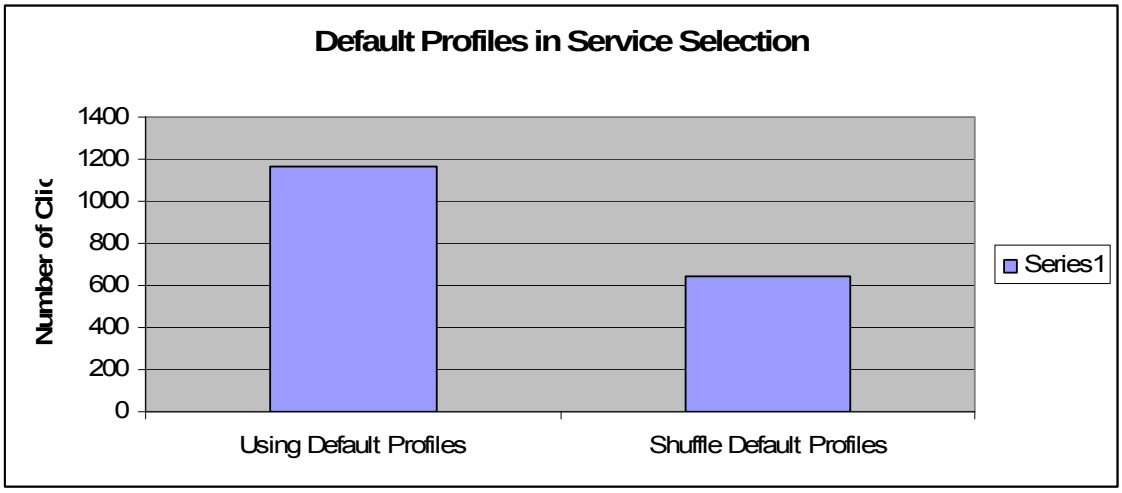
Η τρέχουσα κατάσταση και δραστηριότητα του χρήστη επηρεάζει επίσης σημαντικά τα ενδιαφέροντα του σε ότι αφορά την επιλογή της υπηρεσίας που τον ενδιαφέρει. Η άγνοια μιας κατάστασης της μορφής «Ο χρήστης είναι σε διακοπές» μπορεί να μειώσει σημαντικά την απόδοση ενός εξατομικευμένου συστήματος σε ότι αφορά τις υπηρεσίες που προτείνει ένα εξατομικευμένο σύστημα σε ένα κινητό χρήστη. Τα παραπάνω αποτελέσματα μας οδηγούν επιπλέον στο συμπέρασμα ότι οι χρονικές περίοδοι πιθανότατα αλλάζουν από experience σε experience και για το λόγο αυτό έχουμε την μείωση αυτή.



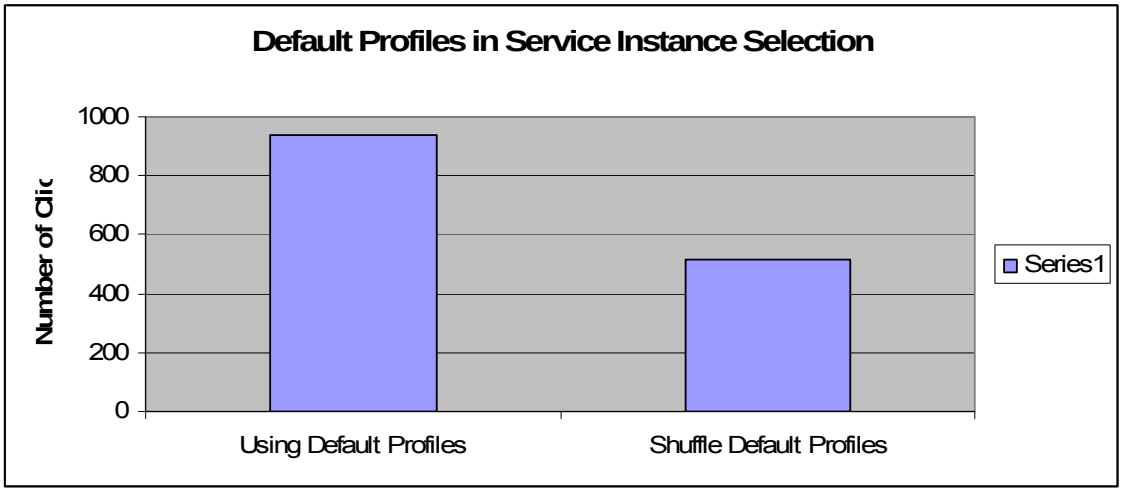
Σχήμα 8-6: Η σημαντικότητα του Experience στην Επιλογή Στιγμιότυπου Υπηρεσίας

Η σημαντικότητα του experience φαίνεται και στην επιλογή στιγμιότυπου μιας υπηρεσίας. Οι προτιμήσεις στα χαρακτηριστικά μιας υπηρεσίας φαίνεται να αλλάζουν ανάλογα με την κατάσταση του χρήστη και για το λόγο αυτό παίρνουμε χειρότερα αποτελέσματα αν αγνοήσουμε τον πιο πάνω παράγοντα.

9.1.3 Η σημαντικότητα των αρχικών προφίλ



Σχήμα 8-7: Η σημαντικότητα των Default Profiles στην Επιλογή Υπηρεσίας



Σχήμα 8-8: Η σημαντικότητα των Default Profiles στην Επιλογή Στιγμιότυπου Υπηρεσίας

Η σημαντικότητα των αρχικών προφίλ φαίνεται στις πρώτες εκτελέσεις του συστήματος. Τα πρώτα αποτελέσματα θα εμφανιστούν στο χρήστη λαμβάνοντας υπόψη μόνο το αρχικό προφίλ. Μέχρι να ενημερωθεί το προφίλ από τις επιλογές του χρήστη είναι σημαντικό το σύστημα να δίνει καλά αποτελέσματα. Το αντίστροφο θα μπορούσε ακόμη και αποτρέψει το χρήστη από το να χρησιμοποιήσει το σύστημα. Η απόδοση του συστήματος λοιπόν στις πρώτες εκτελέσεις είναι πάρα πολύ σημαντική. Όπως φαίνεται και πιο πάνω, η απόδοση αυτή μπορεί να μειωθεί σημαντικά εάν δοθούν λανθασμένα αρχικά προφίλ στους χρήστες.

9.2 Μελλοντική Εργασία

Η εργασία αυτή ανοίγει περαιτέρω προοπτικές για έρευνα σε ότι αφορά την εξατομίκευση σε κινητά συστήματα με βάση αλγορίθμου που συλλέγουν το ιστορικό των clicks του χρήστη για να εξάγουν τα ενδιαφέροντά του.

Η διαδικασία εντοπισμού των ενδιαφερόντων του χρήστη, μπορεί εύκολα να τροποποιηθεί ώστε να επεκταθεί σε συστήματα τα οποία δεν χρησιμοποιούν οντολογίες στην περιγραφή των υπηρεσιών που προσφέρουν. Η διαδικασία συλλογής των ενδιαφερόντων του χρήστη μπορεί να γίνει και σε επεκταθεί σε ιστιακούς πόρους, χρησιμοποιώντας αλγορίθμους και τεχνικές εξατομίκευσης για σελίδες που στο περιεχόμενό τους περιλαμβάνουν απλά κείμενο. Για παράδειγμα μπορούν να χρησιμοποιηθούν αλγόριθμοι όπως TF-IDF και Time-decay και τεχνικές εντοπισμού και εξόρυξης των σημαντικών όρων από ένα κείμενο.

Επιπλέον, ο αλγόριθμος Histogram μπορεί να μελετηθεί και να χρησιμοποιηθεί γενικότερα για την προσαρμογή χαρακτηριστικών πέραν των χρονικών περιόδων. Ο αλγόριθμος αυτός έχει την ικανότητα να διαχωρίζει με τον καλύτερο δυνατό τρόπο χαρακτηριστικά με αριθμητικές τιμές και θα μπορούσε να επεκταθεί και να μελετηθεί γενικότερα η συμπεριφορά του με την προσαρμογή τέτοιων χαρακτηριστικών.

Η εργασία αυτή έχει περιοριστεί στην μελέτη νέων χαρακτηριστικών τα οποία εμφανίζονται στους χρήστες υπολογιστικών μονάδων που αναφέρονται στο κινητό περιβάλλον. Τα χαρακτηριστικά αυτά είναι η τοποθεσία, ο χρόνος και η κατάσταση του χρήστη την στιγμή της αναζήτησης. Η ερώτηση είναι, είναι μόνο αυτά τα νέα χαρακτηριστικά που εμφανίζονται στο κινητό χρήστη και είναι αυτά και μόνο ικανά να κάνουν τον κινητό χρήστη ικανοποιημένο;

Πέραν από τους αλγορίθμους εξατομίκευσης που βοηθούν στο να εμφανιστούν τα επιθυμητά αποτελέσματα στο χρήστη, στο κινητό περιβάλλον υπάρχουν και περιορισμοί στην σύνδεση και στην χωρητικότητα. Λαμβάνοντας αυτά υπόψη οι αλγόριθμοι που μελετήθηκαν στην παρούσα εργασία θα μπορούσαν να μελετηθούν και σαν βάση για αλγορίθμους που προσφέρονται για επίλυση αυτών των προβλημάτων όπως, prefetching, caching.

Βιβλιογραφία

[1] C. Panayiotou, M. Andreou, G. Samaras, A. Pitsillides, Time Based Personalization for the Moving User, Proceedings of International Conference mBusiness (2005), Sydney, Australia, 11-13 July 2005.

[2] C. Panayiotou, M. Andreou, G. Samaras, Using Time and Activity in Personalization for the mobile User, MobiDE, 2006

[3] G. Samaras, C. Panayiotou, Personalized portals for the wireless user based on mobile agents, Workshop Mobile Commerce, 2002

[4] Χριστόφορος Παναγιώτου, Ανάπτυξη ευέλικτης αρχιτεκτονικής συστημάτων προσωπικοποίησης: Πρωτότυπο σύστημα προσωπικοποίησης ασύρματου διαδικτύου (WAP), Διπλωματική Εργασία Εξειδίκευσης Πανεπιστήμιο Κύπρου, Τμήμα Πληροφορικής, Δεκέμβριος 2001

[5] Andrea Goldsmith, Overview of Wireless Communications, Cambridge University Press

[6] International Engineering Consortium, Global System for Mobile Communication, www.iec.org

[7] Ric Howell, WAP Overview, http://www.topxml.com

[8] W3C Communications Team, XML In 10 Points, www.w3.org

[9] http://www.w3schools.com/xml/xml_whatis.asp

[10] World Wide Web Consortium , Extensible Markup Language (XML) 1.0, www.w3.org

[11] Brett McLaughlin. "Java and XML", First Edition, O'Reilly

[12] Michael Jervis, XML DTDs Vs XML Schema can be found at: http://www.sitepoint.com/article/xml-dtds-xml-schema/3

[13] World Wide Web Consortium, XML Schema, http://www.w3.org/XML/Schema

[14] Harvey M. Deitel, Paul J. Deitel, T. R. Nieto, Ted Lin, Praveen Sadhu, "XML How to Program", First Edition

[15] James Gosling, Henry McGilton, The Java Language Environment, White Paper, May 1996, www.java.sun.com

[16] StatSoft, Statistics: Methods and Applications, http://www.statsoft.com

[17] Jiawei Han, Micheline Kamber, Data Mining: Concepts and Techniques, First Edition 2001

[18] Decision Point - Complete Market Research at a Glance, Moving Averages, http://www.decisionpoint.com/TAcourse/MovingAve.html

[19] Giorgio Ingargiola, Building Classification Models: ID3 and C4.5, http://www.cis.temple.edu/~ingargio/cis587/readings/id3-c45.html

[20] Haym Hirsh, Chumki Basu and Brian D. Davidson, *Learning to Personalize*, Communications Of the ACM Vol43, No8, August 2000

[21] Willy Chiu, *Web site Personalization*, http://www.ibm.com

[22] Cyrus Shahabi and Yi-Shin Chen, *Web Information Personalization: Challenges and Approaches*, In proceedings of *Third International Workshop on Databases in Networked Information Systems*, 2003, 5--15.

[23] Bonett, *M. Personalization of Web Services: Opportunities and Challenges* ARIADNE, 28, 2001.

[24] Kyung-Sam Choi, Chi-Hoon Lee and Phill-Kyu Rhee, *Document Ontology Based Personalized Filtering System*, International Multimedia Conference 2000

[25] Stuart E. Middleton, Niger R. Shadbolt and David C de Roure, *Ontological User Profiling in Recommender Systems*, ACM Transactions on Information Systems (TOIS) Volume 22 , Issue 1 (January 2004)

[26] Tsvi Kufflik and Petetz Shoval, *Generation of User Profile for Information Filtering – Research Agenda*, Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, 2000

[27] Lalitha Suryanarayana and Johan Hjelm, *Profiles for the Situated Web*, ACM Digital Library, Proceedings of the 11th international conference on World Wide Web, 2002

[28] Μαρία Αντρέου, *Εξατομίκευση για κινητούς και ασύρματους χρήστες, βασισμένη στην ηλικία το χρόνο και την εμπειρία*, Ατομική Διπλωματική Εργασία Πανεπιστήμιο Κύπρου, Τμήμα Πληροφορικής, Ιούλιος 2004

[29] Dong-Ho Kim, Vijayalakshmi Atluri, Il Im, Michael Bieber, Nabil Adam, Yelena Yesha, *A Clickstream-Based Collaborative Filtering Personalization Model: Towards A Better Performance*, Proceedings of the 6th annual ACM international workshop on Web information and data management, 2004

[30] Magdalini Eirinaki, Michalis Vazirgiannis, *Web Mining for Web Personalization*, ACM Transactions on Internet Technology (TOIT), Volume 3 , Issue 1 (February 2003)

[31] Stuart E. Middleton, David C. De Roure and Nigel R. Shadbolt, *Capturing knowledge of user preferences: ontologies in recommender systems*, Proceedings of the 1st international conference on Knowledge capture, 2001

[32] C. R. Anderson, P. Domingos, and D. S. Weld. *Personalizing web sites for mobile users*, In Proc. of the 10th Intl. WWW Conf., 2001

[33] Eunshil Lee, Jinbeom Kang, Joongmin Choi, Jaeyoung Yang, *Topic-SpecificWeb Content Adaptation to Mobile Devices*, Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, 2006

[34] M.M. Lankhorst, H. van Kranenburg, A. Salden, and A.J.H. Peddemors, *Enabling Technology for Personalizing Mobile Services*, Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02)-Volume 3 - Volume 3

[35] S. Bull, *User Modelling and Mobile Learning*, User Modeling 2003: 9 the International Conference, SpringerVerlag, Berlin Heidelberg, 2003, pp. 383-387

[36] Lalitha Suryanarayana and Johan Hjelm, *Profiles for the Situated Web*, International World Wide Web Conference, Proceedings of the 11th international conference on World Wide Web, 2002

[37] Sarabjot Singh Anand, Patricia Kearney, Mary Shapcott, *Generating semantically enriched user profiles for Web personalization*, ACM Transactions on Internet Technology (TOIT), Volume 7 , Issue 4 (October 2007)

[38] Georgia Koutrika, Yiannis Ioannidis, *Personalization of Queries Based on User Preference,* Proceedings of the 21st International Conference on Data Engineering (ICDE 2005)

[39] Susan Gauch, Jason Chaffee, Alexander Pretschner, *Ontology-Based Personalized Search and Browsing*. Web Intelligence and Agent Systems archive Volume 1 , Issue 3-4 (March 2003)

[40] Corin R. Andreson, Pedro Domingos, Daniel S. Weld, *Adaptive Web Navigation for Wireless Devices,* In Proceedings of the 17th International Joint Conference on Artificial Intelligence, pp. 879–884.

Παράρτημα Α

Στο παράρτημα αυτό δίνονται οι βασικές XML γραμματικές που ορίστηκαν και χρησιμοποιήθηκαν κατά την ανάπτυξη του συστήματος

- userProfile-schema.xsd
- restaurant-schema.xsd
- hotel-schema.xsd
- bookshop-schema.xsd
- bar-schema.xsd
- cafe-schema.xsd
- copycenter-schema.xsd

userProfile-schema.xsd

```
    <?xml version="1.0" encoding="UTF-8"?>
    <!-- edited with XMLSPY v2004 rel. 3 U (http://www.xmlspy.com) by Maria Andreou (UCY) -->
    <xs:schema xmlns:xs=http://www.w3.org/2001/XMLSchema elementFormDefault="qualified"
attributeFormDefault="unqualified">
        <!--
**********************************USER_PROFILE*********************************--
>
        <!--
*****************************************************************************--
>
        <xs:element name="userProfile">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element name="userName" type="userInfo" />
                                <xs:element name="userLocation" type="location" minOccurs="0"/>
                                <xs:element name="servicesInfo" type="servicesInfo"/>
                        </xs:sequence>
                        <xs:attribute name="IMEI" type="IMEIType" use="required"/>
                </xs:complexType>
        </xs:element>
        <!--
**********************************USER_INFO***********************************-
->
        <!--
*****************************************************************************--
>
        <xs:complexType name="userInfo">
                <xs:sequence>
                        <xs:element name="ID" type="IDType"/>
                        <xs:element name="Name" type="nameType"/>
                        <xs:element name="Middle" type="middleNameType" minOccurs="0"/>
                        <xs:element name="Surname" type="nameType"/>
                </xs:sequence>
        </xs:complexType>
        <!--
**********************************LOCATION************************************-
->
        <!--
*****************************************************************************--
>
        <!--prostoparwn den ginetai xrisi autou tou xaraktiristikou-apla topotheteitaigia mellontikous
skopous-->
        <xs:complexType name="location">
                <xs:sequence>
                        <xs:element name="postalCode" type="xs:integer" minOccurs="0"/>
                        <xs:element name="area" type="xs:string"/>
                        <xs:element name="town" type="provincesAndTowns"/>
                        <xs:element name="province" type="provincesAndTowns"/>
                        <xs:element name="country" type="xs:string" fixed="Cyprus"/>
                </xs:sequence>
        </xs:complexType>
        <!--
**********************************SERVICES_INFO*******************************-
->
        <!--
*****************************************************************************--
>
        <xs:complexType name="servicesInfo">
                <xs:sequence>
                        <xs:element name="restaurantsInfo" type="restaurantsInfo"/>
                        <xs:element name="hotelsInfo" type="hotelsInfo"/>
                        <xs:element name="pharmaciesInfo" type="pharmaciesInfo"/>
                        <xs:element name="cafesInfo" type="cafesInfo"/>
                        <xs:element name="bookshopsInfo" type="bookshopsInfo"/>
                        <xs:element name="barsInfo" type="barsInfo"/>
                        <xs:element name="copycentersInfo" type="copycentersInfo"/>
```

```
                </xs:sequence>
                <xs:attribute name="lastUpdated"/>
        </xs:complexType>
        <!--
**************************************BARSINFO**********************************-
->
        <xs:complexType name="barsInfo">
                <xs:sequence>
                        <xs:element name="barServices" type="barServicesType"/>
                        <xs:element name="timeZones" type="timeZones"/>
                </xs:sequence>
                <xs:attribute name="accesses" type="xs:integer" use="required"/>
        </xs:complexType>
        <!--ccServicesType-->
        <xs:complexType name="barServicesType">
                <xs:sequence>
                        <xs:element  name="barServiceType"  type="barServiceType"  minOccurs="0"
maxOccurs="unbounded"/>
                </xs:sequence>
                <xs:attribute name="categoryWeight" type="weightType" use="required"/>
        </xs:complexType>
        <!--ccServiceType-->
        <xs:complexType name="barServiceType">
                <xs:attribute name="name" type="bSType" use="required"/>
                <xs:attribute name="weight" type="weightType" use="required"/>
                <xs:attribute name="previousTimeDecay" type="floatTimeDecay" use="required"/>
                <xs:attribute name="currentTimeDecay" type="xs:integer" use="required"/>
                <xs:attribute name="lastUpdated"/>
        </xs:complexType>
        <!--
**************************************COPYCENTERINFO*****************************-
->
        <xs:complexType name="copycentersInfo">
                <xs:sequence>
                        <xs:element name="ccServices" type="ccServicesType"/>
                        <xs:element name="timeZones" type="timeZones"/>
                </xs:sequence>
                <xs:attribute name="accesses" type="xs:integer" use="required"/>
        </xs:complexType>
        <!--ccServicesType-->
        <xs:complexType name="ccServicesType">
                <xs:sequence>
                        <xs:element   name="ccServiceType"   type="ccServiceType"   minOccurs="0"
maxOccurs="unbounded"/>
                </xs:sequence>
                <xs:attribute name="categoryWeight" type="weightType" use="required"/>
        </xs:complexType>
        <!--ccServiceType-->
        <xs:complexType name="ccServiceType">
                <xs:attribute name="name" type="ccSType" use="required"/>
                <xs:attribute name="weight" type="weightType" use="required"/>
                <xs:attribute name="previousTimeDecay" type="floatTimeDecay" use="required"/>
                <xs:attribute name="currentTimeDecay" type="xs:integer" use="required"/>
                <xs:attribute name="lastUpdated"/>
        </xs:complexType>
        <!--
**************************************BOOKSHOPS_INFO****************************-
->
        <xs:complexType name="bookshopsInfo">
                <xs:sequence>
                        <xs:element name="bsServices" type="bsServicesType"/>
                        <xs:element name="bookTypes" type="bookTypes"/>
                        <xs:element name="timeZones" type="timeZones"/>
                </xs:sequence>
                <xs:attribute name="accesses" type="xs:integer" use="required"/>
        </xs:complexType>
        <!--ccServicesType-->
        <xs:complexType name="bsServicesType">
                <xs:sequence>
```

```
                        <xs:element name="bsServiceType" type="bsServiceType" minOccurs="0"
maxOccurs="unbounded"/>
                </xs:sequence>
                <xs:attribute name="categoryWeight" type="weightType" use="required"/>
        </xs:complexType>
        <!--ccServicesType-->
        <xs:complexType name="bookTypes">
                <xs:sequence>
                        <xs:element name="bookType" type="bookType" minOccurs="0"
maxOccurs="unbounded"/>
                </xs:sequence>
                <xs:attribute name="categoryWeight" type="weightType" use="required"/>
        </xs:complexType>
        <!--ccServiceType-->
        <xs:complexType name="bsServiceType">
                <xs:attribute name="name" type="bsSType" use="required"/>
                <xs:attribute name="weight" type="weightType" use="required"/>
                <xs:attribute name="previousTimeDecay" type="floatTimeDecay" use="required"/>
                <xs:attribute name="currentTimeDecay" type="xs:integer" use="required"/>
                <xs:attribute name="lastUpdated"/>
        </xs:complexType>
        <!--ccServiceType-->
        <xs:complexType name="bookType">
                <xs:attribute name="name" type="bType" use="required"/>
                <xs:attribute name="weight" type="weightType" use="required"/>
                <xs:attribute name="previousTimeDecay" type="floatTimeDecay" use="required"/>
                <xs:attribute name="currentTimeDecay" type="xs:integer" use="required"/>
                <xs:attribute name="lastUpdated"/>
        </xs:complexType>
        <!--
**************************************CAFES**************************************-
->
        <xs:complexType name="cafesInfo">
                <xs:sequence>
                        <xs:element name="foodCategories" type="foodCategories"/>
                        <xs:element name="drinkCategories" type="drinkCategories"/>
                        <xs:element name="timeZones" type="timeZones"/>
                </xs:sequence>
                <xs:attribute name="accesses" type="xs:integer" use="required"/>
        </xs:complexType>
        <!--FOOD_CATEGORIES-->
        <xs:complexType name="foodCategories">
                <xs:sequence>
                        <xs:element name="foodCategoryType" type="foodCategoryType"
minOccurs="0" maxOccurs="3"/>
                </xs:sequence>
                <xs:attribute name="categoryWeight" type="weightType" use="required"/>
        </xs:complexType>
        <!--DRINK_CATEGORIES-->
        <xs:complexType name="drinkCategories">
                <xs:sequence>
                        <xs:element name="drinkCategoryType" type="drinkCategoryType"
minOccurs="0" maxOccurs="3"/>
                </xs:sequence>
                <xs:attribute name="categoryWeight" type="weightType" use="required"/>
        </xs:complexType>
        <!--FOOD_CATEGORY_TYPE-->
        <xs:complexType name="foodCategoryType">
                <xs:attribute name="name" type="cafeFoodCategory" use="required"/>
                <xs:attribute name="weight" type="weightType" use="required"/>
                <xs:attribute name="previousTimeDecay" type="floatTimeDecay" use="required"/>
                <xs:attribute name="currentTimeDecay" type="xs:integer" use="required"/>
                <xs:attribute name="lastUpdated"/>
        </xs:complexType>
        <!--DRINK_CATEGORY_TYPE-->
        <xs:complexType name="drinkCategoryType">
                <xs:attribute name="name" type="cafeDrinkCategory" use="required"/>
                <xs:attribute name="weight" type="weightType" use="required"/>
                <xs:attribute name="previousTimeDecay" type="floatTimeDecay" use="required"/>
```

```
                <xs:attribute name="currentTimeDecay" type="xs:integer" use="required"/>
                <xs:attribute name="lastUpdated"/>
        </xs:complexType>
        <!--
**************************************HOTELS************************************-
->
        <xs:complexType name="hotelsInfo">
                <xs:sequence>
                        <xs:element name="hotelCategoryInfo" type="hotelCategories"/>
                        <xs:element name="hotelFacilities" type="hotelFacilities"/>
                        <xs:element name="locationCategoryInfo" type="locationCategories"/>
                        <xs:element name="averagePrice" type="averagePrice"/>
                        <xs:element name="hotelsTypeByTime" type="hotelsTypeByTime"/>
                        <xs:element name="timeZones" type="timeZones"/>
                </xs:sequence>
                <xs:attribute name="accesses" type="xs:integer" use="required"/>
        </xs:complexType>
        <!--
***********************************HOTELS_TYPE_BY_TIME**************************-
->
        <xs:complexType name="hotelsTypeByTime">
                <xs:sequence>
                        <xs:element              name="timeZone"              type="HTypeByTime"
maxOccurs="unbounded"/>
                </xs:sequence>
                <xs:attribute name="categoryWeight" type="weightType" use="required"/>
        </xs:complexType>
        <!--
***********************************HTypeByTime**********************************--
>
        <xs:complexType name="HTypeByTime">
                <xs:sequence>
                        <xs:element name="hotelCategory" type="hotelCategoryInfo" minOccurs="0"
maxOccurs="5"/>
                </xs:sequence>
                <xs:attribute name="time" use="required"/>
        </xs:complexType>
        <!--************LoCATION_CATEGORIES***************-->
        <xs:complexType name="locationCategories">
                <xs:sequence>
                        <xs:element  name="locationCategory"  type="locationType"  minOccurs="0"
maxOccurs="5"/>
                </xs:sequence>
                <xs:attribute name="categoryWeight" type="weightType" use="required"/>
        </xs:complexType>
        <!--LOCATION TYPE-->
        <xs:complexType name="locationType">
                <xs:attribute name="name" type="locationCategory" use="required"/>
                <xs:attribute name="weight" type="weightType" use="required"/>
                <xs:attribute name="previousTimeDecay" type="floatTimeDecay" use="required"/>
                <xs:attribute name="currentTimeDecay" type="xs:integer" use="required"/>
                <xs:attribute name="lastUpdated"/>
        </xs:complexType>
        <!--*********HOTEL_CATEGORIES*********-->
        <xs:complexType name="hotelCategories">
                <xs:sequence>
                        <xs:element          name="hotelCategory"          type="hotelCategoryInfo"
maxOccurs="5"/>
                </xs:sequence>
                <xs:attribute name="categoryWeight" type="weightType" use="required"/>
        </xs:complexType>
        <!--HOTEL_CATEGORY_INFO-->
        <xs:complexType name="hotelCategoryInfo">
                <xs:attribute name="name" type="hotelCategory" use="required"/>
                <xs:attribute name="weight" type="weightType" use="required"/>
                <xs:attribute name="previousTimeDecay" type="floatTimeDecay" use="required"/>
                <xs:attribute name="currentTimeDecay" type="xs:integer" use="required"/>
                <xs:attribute name="lastUpdated"/>
        </xs:complexType>
```

```
	<!--********HOTEL_FACILITIES**********-->
	<xs:complexType name="hotelFacilities">
		<xs:sequence>
			<xs:element name="generalFacilities" type="generalFacilities"/>
			<xs:element name="hotelActivities" type="hotelActivities"/>
			<xs:element name="hotelServices" type="hotelServices"/>
		</xs:sequence>
		<xs:attribute name="categoryWeight" type="weightType" use="required"/>
	</xs:complexType>
	<!--GENERAL_FACILITIES-->
	<xs:complexType name="generalFacilities">
		<xs:sequence>
			<xs:element name="generalFacility" type="generalFacilitiesType"
minOccurs="0" maxOccurs="unbounded"/>
		</xs:sequence>
		<xs:attribute name="weightCategory" type="weightType" use="required"/>
	</xs:complexType>
	<!--GENERAL_FACILITIES_TYPE-->
	<xs:complexType name="generalFacilitiesType">
		<xs:attribute name="name" type="generalTypeFacility" use="required"/>
		<xs:attribute name="weight" type="weightType" use="required"/>
		<xs:attribute name="previousTimeDecay" type="floatTimeDecay" use="required"/>
		<xs:attribute name="currentTimeDecay" type="xs:integer" use="required"/>
		<xs:attribute name="lastUpdated"/>
	</xs:complexType>
	<!--HOTEL_ACTIVITIES-->
	<xs:complexType name="hotelActivities">
		<xs:sequence>
			<xs:element name="hotelActivity" type="hotelActivity" minOccurs="0"
maxOccurs="unbounded"/>
		</xs:sequence>
		<xs:attribute name="weightCategory" type="weightType" use="required"/>
	</xs:complexType>
	<!--HOTEL_ACTIVITY-->
	<xs:complexType name="hotelActivity">
		<xs:attribute name="name" type="activityType" use="required"/>
		<xs:attribute name="weight" type="weightType" use="required"/>
		<xs:attribute name="previousTimeDecay" type="floatTimeDecay" use="required"/>
		<xs:attribute name="currentTimeDecay" type="xs:integer" use="required"/>
		<xs:attribute name="lastUpdated"/>
	</xs:complexType>
	<!--HOTEL_SERVICES-->
	<xs:complexType name="hotelServices">
		<xs:choice>
			<xs:element name="hotelService" type="hotelService" minOccurs="0"
maxOccurs="unbounded"/>
		</xs:choice>
		<xs:attribute name="weightCategory" type="weightType" use="required"/>
	</xs:complexType>
	<!--HOTEL_SERVICE-->
	<xs:complexType name="hotelService">
		<xs:attribute name="name" type="serviceType" use="required"/>
		<xs:attribute name="weight" type="weightType" use="required"/>
		<xs:attribute name="previousTimeDecay" type="floatTimeDecay" use="required"/>
		<xs:attribute name="currentTimeDecay" type="xs:integer" use="required"/>
		<xs:attribute name="lastUpdated"/>
	</xs:complexType>
	<!--
**********************************PHARMACIES**********************************-
->
	<xs:complexType name="pharmaciesInfo">
		<xs:sequence>
			<xs:element name="timeZones" type="timeZones"/>
		</xs:sequence>
	</xs:complexType>
	<!--
**********************************RESTAURANTS*********************************-
->
	<!--RESTAURANTS_INFO-->
```

```
        <xs:complexType name="restaurantsInfo">
                <xs:sequence>
                        <xs:element name="restaurantTypeInfo" type="restaurantTypeInfo"/>
                        <xs:element name="foodCategory" type="foodCategory"/>
                        <xs:element name="restaurantServiceInfo" type="restaurantServices"/>
                        <xs:element name="averagePrice" type="averagePrice"/>
                        <xs:element                                    name="restaurantsTypeByTime"
type="restaurantsTypeByTime"/>
                        <xs:element name="timeZones" type="timeZones"/>
                </xs:sequence>
                <xs:attribute name="accesses" type="xs:integer" use="required"/>
        </xs:complexType>
        <!--
**************************RESTAURANT_TYPE_BY_TIME*********************************-
->
        <xs:complexType name="restaurantsTypeByTime">
                <xs:sequence>
                        <xs:element              name="timeZone"              type="RTypeByTime"
maxOccurs="unbounded"/>
                </xs:sequence>
                <xs:attribute name="categoryWeight" type="weightType" use="required"/>
        </xs:complexType>
        <!--
************************************RTypeByTime**********************************--
>
        <xs:complexType name="RTypeByTime">
                <xs:sequence>
                        <xs:element  name="restaurantType"  type="restaurantType"  minOccurs="0"
maxOccurs="unbounded"/>
                </xs:sequence>
                <xs:attribute name="time" type="xs:string" use="required"/>
        </xs:complexType>
        <!--***********RESTAURANT_TYPE_INFO*************-->
        <xs:complexType name="restaurantTypeInfo">
                <xs:sequence>
                        <xs:element  name="restaurantType"  type="restaurantType"  minOccurs="0"
maxOccurs="unbounded"/>
                </xs:sequence>
                <xs:attribute name="categoryWeight" type="weightType" use="required"/>
        </xs:complexType>
        <!--RESTAURANT_TYPE-->
        <xs:complexType name="restaurantType">
                <xs:attribute name="name" type="restaurantTypeName" use="required"/>
                <xs:attribute name="weight" type="weightType" use="required"/>
                <xs:attribute name="previousTimeDecay" type="floatTimeDecay" use="required"/>
                <xs:attribute name="currentTimeDecay" type="xs:integer" use="required"/>
                <xs:attribute name="lastUpdated" type="xs:string"/>
        </xs:complexType>
        <!--***************FOOD_CATEGORY****************-->
        <xs:complexType name="foodCategory">
                <xs:all>
                        <xs:element name="poultry" type="poultryCategory"/>
                        <xs:element name="meat" type="meatCategory"/>
                        <xs:element name="fish" type="fishCategory"/>
                        <xs:element name="seafood" type="seafoodCategory"/>
                        <xs:element name="other" type="otherCategory"/>
                </xs:all>
                <xs:attribute name="categoryWeight" use="required"/>
        </xs:complexType>
        <!--POULTRY_CATEGORY:poulerika-->
        <xs:complexType name="poultryCategory">
                <xs:sequence>
                        <xs:element    name="poultryType"    type="poultryType"    minOccurs="0"
maxOccurs="unbounded"/>
                </xs:sequence>
                <xs:attribute name="weightCategory" type="weightType" use="required"/>
        </xs:complexType>
        <!--POULTRY_TYPE-->
        <xs:complexType name="poultryType">
```

```
                <xs:attribute name="name" type="poultryTypeName" use="required"/>
                <xs:attribute name="weight" type="weightType" use="required"/>
                <xs:attribute name="previousTimeDecay" type="floatTimeDecay" use="required"/>
                <xs:attribute name="currentTimeDecay" type="xs:integer" use="required"/>
                <xs:attribute name="lastUpdated" type="xs:string"/>
        </xs:complexType>
        <!--MEAT_CATEGORY-->
        <xs:complexType name="meatCategory">
                <xs:sequence>
                        <xs:element      name="meatType"      type="meatType"      minOccurs="0"
maxOccurs="unbounded"/>
                </xs:sequence>
                <xs:attribute name="weightCategory" type="weightType" use="required"/>
        </xs:complexType>
        <!--MEAT_TYPE DISHES-->
        <xs:complexType name="meatType">
                <xs:attribute name="name" type="meatTypeName" use="required"/>
                <xs:attribute name="weight" type="weightType" use="required"/>
                <xs:attribute name="previousTimeDecay" type="floatTimeDecay" use="required"/>
                <xs:attribute name="currentTimeDecay" type="xs:integer" use="required"/>
                <xs:attribute name="lastUpdated"/>
        </xs:complexType>
        <!--FISH_CATEGORY-->
        <xs:complexType name="fishCategory">
                <xs:sequence>
                        <xs:element       name="fishType"       type="fishType"       minOccurs="0"
maxOccurs="unbounded"/>
                </xs:sequence>
                <xs:attribute name="weightCategory" type="weightType" use="required"/>
        </xs:complexType>
        <!--FISH_TYPE-->
        <xs:complexType name="fishType">
                <xs:attribute name="name" type="fishTypeName" use="required"/>
                <xs:attribute name="weight" type="weightType" use="required"/>
                <xs:attribute name="previousTimeDecay" type="floatTimeDecay" use="required"/>
                <xs:attribute name="currentTimeDecay" type="xs:integer" use="required"/>
                <xs:attribute name="lastUpdated"/>
        </xs:complexType>
        <!--SEAFOOD_CATEGORY-->
        <xs:complexType name="seafoodCategory">
                <xs:sequence>
                        <xs:element    name="seafoodType"    type="seafoodType"    minOccurs="0"
maxOccurs="unbounded"/>
                </xs:sequence>
                <xs:attribute name="weightCategory" type="weightType" use="required"/>
        </xs:complexType>
        <!--SEAFOOD-->
        <xs:complexType name="seafoodType">
                <xs:attribute name="name" type="seafoodTypeName" use="required"/>
                <xs:attribute name="weight" type="weightType" use="required"/>
                <xs:attribute name="previousTimeDecay" type="floatTimeDecay" use="required"/>
                <xs:attribute name="currentTimeDecay" type="xs:integer" use="required"/>
                <xs:attribute name="lastUpdated"/>
        </xs:complexType>
        <!--OTHER_CATEGORY-->
        <xs:complexType name="otherCategory">
                <xs:sequence>
                        <xs:element      name="otherType"      type="otherType"      minOccurs="0"
maxOccurs="unbounded"/>
                </xs:sequence>
                <xs:attribute name="weightCategory" type="weightType" use="required"/>
        </xs:complexType>
        <!--OTHER-->
        <xs:complexType name="otherType">
                <xs:attribute name="name" type="otherTypeName" use="required"/>
                <xs:attribute name="weight" type="weightType" use="required"/>
                <xs:attribute name="previousTimeDecay" type="floatTimeDecay" use="required"/>
                <xs:attribute name="currentTimeDecay" type="xs:integer" use="required"/>
                <xs:attribute name="lastUpdated"/>
```

```
	</xs:complexType>
	<!--*****************RESTAURANT_SERVICES*****************-->
	<!--Exw mono3 eidi restaurant services: delivery, take away, and both-->
	<xs:complexType name="restaurantServices">
		<xs:sequence>
			<xs:element	name="restaurantService"	type="restaurantServiceType"
minOccurs="0" maxOccurs="3"/>
		</xs:sequence>
		<xs:attribute name="categoryWeight" type="weightType" use="required"/>
	</xs:complexType>
	<!--RESTAURANT_SERVICES_TYPE-->
	<xs:complexType name="restaurantServiceType">
		<xs:attribute name="name" type="restaurantServicesTypeName" use="required"/>
		<xs:attribute name="weight" type="weightType" use="required"/>
		<xs:attribute name="previousTimeDecay" type="floatTimeDecay" use="required"/>
		<xs:attribute name="currentTimeDecay" type="xs:integer" use="required"/>
		<xs:attribute name="lastUpdated"/>
	</xs:complexType>
	<!--
*******************************GENERAL_ELEMENTS*******************************--
>
	<!--
*****************************************************************************--
>
	<!--********************TIME_ZONES********************-->
	<xs:complexType name="timeZones">
		<xs:sequence>
			<xs:element		name="timeZone"		type="timeZoneType"
maxOccurs="unbounded"/>
		</xs:sequence>
	</xs:complexType>
	<!--TIME_ZONE_TYPE-->
	<xs:complexType name="timeZoneType">
		<xs:attribute name="time" use="required"/>
		<xs:attribute name="weight" use="required"/>
		<xs:attribute name="requestNum" use="required"/>
		<xs:attribute name="errorRequestNum" use="required"/>
	</xs:complexType>
	<!--******************AVERAGE_PRICE******************-->
	<xs:complexType name="averagePrice">
		<xs:choice>
			<xs:element	name="preferableAveragePrice"	type="preferableAveragePrice"
minOccurs="0" maxOccurs="unbounded"/>
			<xs:element		name="preferableAveragePriceH"
type="preferableHotelsAveragePrice" minOccurs="0" maxOccurs="unbounded"/>
		</xs:choice>
		<xs:attribute name="categoryWeight" type="weightType" use="required"/>
	</xs:complexType>
	<!--PREFERABLE_AVERAGE_PRICE-->
	<xs:complexType name="preferableAveragePrice">
		<xs:attribute name="serviceType" type="restaurantTypeName" use="required"/>
		<xs:attribute name="preferablePrice" type="xs:decimal" use="required"/>
		<xs:attribute name="weight" type="weightType" use="required"/>
		<xs:attribute name="lastUpdated"/>
	</xs:complexType>
	<!--PREFERABLE_HOTEL_AVERAGE_PRICE-->
	<xs:complexType name="preferableHotelsAveragePrice">
		<xs:attribute name="serviceType" type="hotelCategory" use="required"/>
		<xs:attribute name="preferablePrice" type="xs:decimal" use="required"/>
		<xs:attribute name="weight" type="weightType" use="required"/>
		<xs:attribute name="lastUpdated"/>
	</xs:complexType>
	<!--
*******************************GENERAL_RESTRICTION_TYPES************************-
->
	<!--
*****************************************************************************--
>
	<!--IMEI type: to IMEI einai enas integer arithmos 15 psifiwn-->
```

```
	<xs:simpleType name="IMEIType">
		<xs:restriction base="xs:integer">
			<xs:pattern value="[0-9]{15}"/>
		</xs:restriction>
	</xs:simpleType>
	<!--IDType: ID is an 6digit integer number-->
	<xs:simpleType name="IDType">
		<xs:restriction base="xs:integer">
			<xs:pattern value="[0-9]{6}"/>
		</xs:restriction>
	</xs:simpleType>
	<!--NAME_TYPE: To onoma arxizei me kefalaio kai sinexizei me mikra grammata-->
	<xs:simpleType name="nameType">
		<xs:restriction base="xs:string">
			<xs:pattern value="[A-Z]([a-z])+"/>
		</xs:restriction>
	</xs:simpleType>
	<!--MIDDLE_NAME_TYPE:To middle Name type einai ena gramma kefalaio-->
	<xs:simpleType name="middleNameType">
		<xs:restriction base="xs:string">
			<xs:pattern value="[A-Z]"/>
		</xs:restriction>
	</xs:simpleType>
	<!--WEIGHT_TYPE:To weight se mia ipiresia einai enas akeraios apo to 0-100-->
	<xs:simpleType name="weightType">
		<xs:restriction base="xs:integer">
			<xs:minInclusive value="-1"/>
			<xs:maxInclusive value="100"/>
		</xs:restriction>
	</xs:simpleType>
	<!--FLOAT_TIME_DECAY:To weight se mia ipiresia einai enas akeraios apo to 0-100-->
	<xs:simpleType name="floatTimeDecay">
		<xs:restriction base="xs:float">
			<xs:minInclusive value="0"/>
			<xs:maxInclusive value="1"/>
		</xs:restriction>
	</xs:simpleType>
	<!--**************************************ENUMERATIONS*****************-->
	<!--*********************************************************************-->
	<!--********************RESTAURANTS**********************-->
	<!--RESTAURANT_TYPE:orizw enumeration gia to sinolo twn restaurantType-->
	<xs:simpleType name="restaurantTypeName">
		<xs:restriction base="xs:string">
			<xs:enumeration value="African"/>
			<xs:enumeration value="American"/>
			<xs:enumeration value="Armenian"/>
			<xs:enumeration value="Austrian"/>
			<xs:enumeration value="Barbecue and Take Away Shops"/>
			<xs:enumeration value="Brazilian"/>
			<xs:enumeration value="Chinese"/>
			<xs:enumeration value="Creperies"/>
			<xs:enumeration value="Cypriot"/>
			<xs:enumeration value="Eating Cafes"/>
			<xs:enumeration value="Fast Food"/>
			<xs:enumeration value="Fish Taverns"/>
			<xs:enumeration value="French"/>
			<xs:enumeration value="Greek"/>
			<xs:enumeration value="Indian"/>
			<xs:enumeration value="International"/>
			<xs:enumeration value="Irish"/>
			<xs:enumeration value="Italian"/>
			<xs:enumeration value="Japanese"/>
			<xs:enumeration value="Kebab House"/>
			<xs:enumeration value="Lebanese/Syrian/Arabic"/>
			<xs:enumeration value="Mexican"/>
			<xs:enumeration value="Pizzarias"/>
			<xs:enumeration value="Polynesian"/>
			<xs:enumeration value="Pubs with restaurants"/>
			<xs:enumeration value="Russian"/>
```

```
                    <xs:enumeration value="Snack Bar"/>
                    <xs:enumeration value="Spanish"/>
                    <xs:enumeration value="Steak House"/>
                    <xs:enumeration value="Taverns"/>
                    <xs:enumeration value="Taverns with mousic"/>
                    <xs:enumeration value="Thai"/>
                    <xs:enumeration value="Yugoslavia"/>
            </xs:restriction>
    </xs:simpleType>
    <!--POULTRY_TYPE-->
    <xs:simpleType name="poultryTypeName">
            <xs:restriction base="xs:string">
                    <xs:enumeration value="Chicken"/>
                    <xs:enumeration value="Duck"/>
                    <xs:enumeration value="Turkey"/>
                    <xs:enumeration value="Cock"/>
            </xs:restriction>
    </xs:simpleType>
    <!--FISH_TYPE-->
    <xs:simpleType name="fishTypeName">
            <xs:restriction base="xs:string">
                    <xs:enumeration value="Salmon"/>
                    <xs:enumeration value="Mullet"/>
                    <xs:enumeration value="Herring"/>
                    <xs:enumeration value="Sardines"/>
                    <xs:enumeration value="Sole"/>
                    <xs:enumeration value="Cod"/>
                    <xs:enumeration value="ColdFish"/>
                    <xs:enumeration value="Tope"/>
                    <xs:enumeration value="Tuna"/>
                    <xs:enumeration value="Swordfish"/>
                    <xs:enumeration value="Trout"/>
                    <xs:enumeration value="RedMullet"/>
                    <xs:enumeration value="Eel"/>
            </xs:restriction>
    </xs:simpleType>
    <!--MEAT_TYPE-->
    <xs:simpleType name="meatTypeName">
            <xs:restriction base="xs:string">
                    <xs:enumeration value="Beef"/>
                    <xs:enumeration value="Pork"/>
                    <xs:enumeration value="Lamp"/>
                    <xs:enumeration value="Rabit"/>
            </xs:restriction>
    </xs:simpleType>
    <!--SEAFOOD_TYPE-->
    <xs:simpleType name="seafoodTypeName">
            <xs:restriction base="xs:string">
                    <xs:enumeration value="Anchovies"/>
                    <xs:enumeration value="Cuttlefish"/>
                    <xs:enumeration value="Prowns"/>
                    <xs:enumeration value="Squid"/>
                    <xs:enumeration value="Mussels"/>
                    <xs:enumeration value="Octapus"/>
            </xs:restriction>
    </xs:simpleType>
    <!--OTHER_TYPE-->
    <xs:simpleType name="otherTypeName">
            <xs:restriction base="xs:string">
                    <xs:enumeration value="Vegetarian"/>
                    <xs:enumeration value="Pasta"/>
                    <xs:enumeration value="Pizzas"/>
                    <xs:enumeration value="Salads"/>
            </xs:restriction>
    </xs:simpleType>
    <!--RESTAURANT_SERVICES-->
    <!--orizw enumeration gia to sinolo twn restaurantServices-->
    <xs:simpleType name="restaurantServicesTypeName">
            <xs:restriction base="xs:string">
```

```
                    <xs:enumeration value="delivery"/>
                    <xs:enumeration value="takeAway"/>
                    <xs:enumeration value="deliveryAndTakeAway"/>
            </xs:restriction>
    </xs:simpleType>
    <!--*****************HOTELS*****************-->
    <!--HOTEL_CATEGORY-->
    <xs:simpleType name="hotelCategory">
            <xs:restriction base="xs:string">
                    <xs:enumeration value="*"/>
                    <xs:enumeration value="**"/>
                    <xs:enumeration value="***"/>
                    <xs:enumeration value="****"/>
                    <xs:enumeration value="*****"/>
            </xs:restriction>
    </xs:simpleType>
    <!--LOCATION_CATEGORY-->
    <xs:simpleType name="locationCategory">
            <xs:restriction base="xs:string">
                    <xs:enumeration value="Coastal"/>
                    <xs:enumeration value="Mountain"/>
                    <xs:enumeration value="InTown"/>
            </xs:restriction>
    </xs:simpleType>
    <!--GENERAL_TYPE_FACILITIES-->
    <xs:simpleType name="generalTypeFacility">
            <xs:restriction base="xs:string">
                    <xs:enumeration value="Parking"/>
                    <xs:enumeration value="Restaurant"/>
                    <xs:enumeration value="Bar"/>
                    <xs:enumeration value="Shops at Hotel"/>
                    <xs:enumeration value="Heeting"/>
            </xs:restriction>
    </xs:simpleType>
    <!--ACTIVITY_TYPE-->
    <xs:simpleType name="activityType">
            <xs:restriction base="xs:string">
                    <xs:enumeration value="Swimming Pool"/>
                    <xs:enumeration value="Tennis-court"/>
                    <xs:enumeration value="Sauna"/>
                    <xs:enumeration value="Fitness Center"/>
                    <xs:enumeration value="Game Room"/>
                    <xs:enumeration value="Billiard"/>
                    <xs:enumeration value="Jacuzzi"/>
            </xs:restriction>
    </xs:simpleType>
    <!--SERVICE_TYPE-->
    <xs:simpleType name="serviceType">
            <xs:restriction base="xs:string">
                    <xs:enumeration value="Room-service"/>
                    <xs:enumeration value="Babysitting"/>
                    <xs:enumeration value="Laundry"/>
                    <xs:enumeration value="Barber/Beauty Shop"/>
                    <xs:enumeration value="Internet Services"/>
            </xs:restriction>
    </xs:simpleType>
    <!--******************CAFES**********************-->
    <!--CAFE_DRINK_CATEGORY-->
    <xs:simpleType name="cafeDrinkCategory">
            <xs:restriction base="xs:string">
                    <xs:enumeration value="hots"/>
                    <xs:enumeration value="colds"/>
                    <xs:enumeration value="alcoholic"/>
            </xs:restriction>
    </xs:simpleType>
    <!--CAFE_FOOD_CATEGORY-->
    <xs:simpleType name="cafeFoodCategory">
            <xs:restriction base="xs:string">
                    <xs:enumeration value="Sandwiches"/>
```

```
                        <xs:enumeration value="Crepes"/>
                        <xs:enumeration value="Tosts"/>
                </xs:restriction>
        </xs:simpleType>
        <!--******************BAR**********************-->
        <xs:simpleType name="bSType">
                <xs:restriction base="xs:string">
                        <xs:enumeration value="FoodServed"/>
                        <xs:enumeration value="SmokeFreeArea"/>
                        <xs:enumeration value="OpenAtNoon"/>
                        <xs:enumeration value="TV"/>
                        <xs:enumeration value="OutDoorsSeating"/>
                </xs:restriction>
        </xs:simpleType>
        <!--******************COPYCENTER**********************-->
        <xs:simpleType name="ccSType">
                <xs:restriction base="xs:string">
                        <xs:enumeration value="Typing"/>
                        <xs:enumeration value="Stationery"/>
                        <xs:enumeration value="Bindings"/>
                        <xs:enumeration value="StudentNotes"/>
                        <xs:enumeration value="Slides"/>
                        <xs:enumeration value="GraphicDesign"/>
                        <xs:enumeration value="FullColorPrinting"/>
                        <xs:enumeration value="DigitalPrinting"/>
                        <xs:enumeration value="MailingLabelingAndStuffing"/>
                </xs:restriction>
        </xs:simpleType>
        <!--******************BOOKSHOP**********************-->
        <xs:simpleType name="bsSType">
                <xs:restriction base="xs:string">
                        <xs:enumeration value="Typing"/>
                        <xs:enumeration value="Stationery"/>
                        <xs:enumeration value="Photocopies"/>
                        <xs:enumeration value="Bindings"/>
                        <xs:enumeration value="StudentNodes"/>
                </xs:restriction>
        </xs:simpleType>
        <xs:simpleType name="bType">
                <xs:restriction base="xs:string">
                        <xs:enumeration value="ComputerBooks"/>
                        <xs:enumeration value="ChildrenBooks"/>
                        <xs:enumeration value="Literature"/>
                        <xs:enumeration value="Science"/>
                        <xs:enumeration value="History"/>
                        <xs:enumeration value="StudentBooks"/>
                        <xs:enumeration value="HealthMindAndBody"/>
                        <xs:enumeration value="Cooking"/>
                        <xs:enumeration value="Calendars"/>
                        <xs:enumeration value="Magazines"/>
                </xs:restriction>
        </xs:simpleType>
        <!--******************GENERAL**********************-->
        <!--PROVINCES_AND_TOWNS-->
        <!--Orizw enumeration gia tin eparxia kai poli: oi poleis tis Kiprou-->
        <xs:simpleType name="provincesAndTowns">
                <xs:restriction base="xs:string">
                        <xs:enumeration value="Nicosia"/>
                        <xs:enumeration value="Limassol"/>
                        <xs:enumeration value="Famagusta"/>
                        <xs:enumeration value="Larnaca"/>
                        <xs:enumeration value="Pafos"/>
                        <xs:enumeration value="Kerynia"/>
                </xs:restriction>
        </xs:simpleType>
</xs:schema>
```

restaurant-schema.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v2004 rel. 3 U (http://www.xmlspy.com) by Maria Andreou (UCY) -->
<!--
**************************************RESTAURANTS_SCHEMA*************************
***************-->
<!--
*******************************************************************************
*****************************-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
	<!--**************RESTAURANT_ELEMENT**************-->
	<xs:element name="restaurant">
		<xs:complexType>
			<xs:sequence>
				<xs:element name="restaurantType" type="restaurantType"
minOccurs="0"/>
				<xs:element name="foodCategory" type="foodCategory"
minOccurs="0"/>
				<xs:element name="restaurantLocation" type="location"
minOccurs="0"/>
				<xs:element name="restaurantService" type="restaurantServices"
minOccurs="0"/>
				<xs:element name="distance" type="xs:decimal" minOccurs="0"/>
				<xs:element name="links" type="linksType"/>
				<xs:element name="details" type="serviceDetails" minOccurs="0"/>
				<xs:element name="averagePrice" type="xs:decimal"
minOccurs="0"/>
			</xs:sequence>
			<xs:attribute name="type" type="fileType" use="required"/>
		</xs:complexType>
	</xs:element>
	<!--******************LINKS_TYPE**************************-->
	<xs:complexType name="linksType">
		<xs:sequence>
			<xs:element name="link" type="linkType" maxOccurs="unbounded"/>
		</xs:sequence>
	</xs:complexType>
	<!--******************LIK_TYPE**************************-->
	<xs:complexType name="linkType">
		<xs:attribute name="address" type="xs:string" use="required"/>
		<xs:attribute name="position" type="xs:string"/>
	</xs:complexType>
	<!--******************INFO**********************************-->
	<xs:complexType name="serviceDetails">
		<xs:sequence>
			<xs:element name="serviceName" type="xs:string"/>
			<xs:element name="phones" type="phonesType"/>
		</xs:sequence>
	</xs:complexType>
	<!--******************PHONES_TYPE*****************-->
	<xs:complexType name="phonesType">
		<xs:sequence>
			<xs:element name="phone" type="phoneType" maxOccurs="unbounded"/>
		</xs:sequence>
	</xs:complexType>
	<!--********************FOOD_CATEGORY**********************-->
	<!--ta elements pou akolouthoun mporoun na emfanistoun me opoiandipote seira apla prepei na
emfanistoun to poli mia fora-->
	<xs:complexType name="foodCategory">
		<xs:all>
			<xs:element name="poultry" type="poultry" minOccurs="0"/>
			<xs:element name="meat" type="meat" minOccurs="0"/>
			<xs:element name="fish" type="fish" minOccurs="0"/>
			<xs:element name="seafood" type="seafood" minOccurs="0"/>
			<xs:element name="other" type="other" minOccurs="0"/>
		</xs:all>
```

```
	</xs:complexType>
	<!--POULTRY:poulerika-->
	<xs:complexType name="poultry">
		<xs:sequence>
			<xs:element name="poultryType" type="poultryType"
maxOccurs="unbounded"/>
		</xs:sequence>
	</xs:complexType>
	<!--MEAT DISHES-->
	<xs:complexType name="meat">
		<xs:sequence>
			<xs:element name="meatType" type="meatType"
maxOccurs="unbounded"/>
		</xs:sequence>
	</xs:complexType>
	<!--SEAFOOD-->
	<xs:complexType name="fish">
		<xs:sequence>
			<xs:element name="fishType" type="fishType" maxOccurs="unbounded"/>
		</xs:sequence>
	</xs:complexType>
	<!--SEAFOOD-->
	<xs:complexType name="seafood">
		<xs:sequence>
			<xs:element name="seafoodType" type="seafoodType"
maxOccurs="unbounded"/>
		</xs:sequence>
	</xs:complexType>
	<!--OTHER-->
	<xs:complexType name="other">
		<xs:sequence>
			<xs:element name="otherType" type="otherType"
maxOccurs="unbounded"/>
		</xs:sequence>
	</xs:complexType>
	<!--*****************LOCATION***************-->
	<!--orizw to xaraktiristiko location pou mas dinei tin perioxi pou vrisketai ena stigmiotipo
ipiresias-->
	<xs:complexType name="location">
		<xs:sequence>
			<xs:element name="postalCode" type="xs:integer" minOccurs="0"/>
			<xs:element name="area" type="xs:string"/>
			<xs:element name="town" type="provincesAndTowns"/>
			<xs:element name="province" type="provincesAndTowns"/>
			<xs:element name="country" type="xs:string" fixed="Cyprus"/>
		</xs:sequence>
	</xs:complexType>
	<!--
************************************ENUMERATIONS*********************************
********-->
	<!--
*****************************************************************************
****************-->
	<!--FILE_TYPE-->
	<xs:simpleType name="fileType">
		<xs:restriction base="xs:string">
			<xs:enumeration value="linkFile"/>
			<xs:enumeration value="serviceFile"/>
		</xs:restriction>
	</xs:simpleType>
	<!--***********************RESTAURANT_TYPE:orizw enumeration gia to sinolo twn
restaurantType-->
	<xs:simpleType name="restaurantType">
		<xs:restriction base="xs:string">
			<xs:enumeration value="African"/>
			<xs:enumeration value="American"/>
			<xs:enumeration value="Armenian"/>
			<xs:enumeration value="Austrian"/>
			<xs:enumeration value="Barbecue and Take Away Shops"/>
```

```
                        <xs:enumeration value="Brazilian"/>
                        <xs:enumeration value="Chinese"/>
                        <xs:enumeration value="Creperies"/>
                        <xs:enumeration value="Cypriot"/>
                        <xs:enumeration value="Eating Cafes"/>
                        <xs:enumeration value="Fast Food"/>
                        <xs:enumeration value="Fish Taverns"/>
                        <xs:enumeration value="French"/>
                        <xs:enumeration value="Greek"/>
                        <xs:enumeration value="Indian"/>
                        <xs:enumeration value="International"/>
                        <xs:enumeration value="Irish"/>
                        <xs:enumeration value="Italian"/>
                        <xs:enumeration value="Japanese"/>
                        <xs:enumeration value="Kebab House"/>
                        <xs:enumeration value="Lebanese/Syrian/Arabic"/>
                        <xs:enumeration value="Mexican"/>
                        <xs:enumeration value="Pizzarias"/>
                        <xs:enumeration value="Polynesian"/>
                        <xs:enumeration value="Pubs with restaurants"/>
                        <xs:enumeration value="Russian"/>
                        <xs:enumeration value="Snack Bar"/>
                        <xs:enumeration value="Spanish"/>
                        <xs:enumeration value="Steak House"/>
                        <xs:enumeration value="Taverns"/>
                        <xs:enumeration value="Taverns with mousic"/>
                        <xs:enumeration value="Thai"/>
                        <xs:enumeration value="Yugoslavia"/>
                </xs:restriction>
        </xs:simpleType>
        <!--POULTRY_TYPE-->
        <xs:simpleType name="poultryType">
                <xs:restriction base="xs:string">
                        <xs:enumeration value="Chicken"/>
                        <xs:enumeration value="Duck"/>
                        <xs:enumeration value="Turkey"/>
                        <xs:enumeration value="Cock"/>
                </xs:restriction>
        </xs:simpleType>
        <!--FISH_TYPE-->
        <xs:simpleType name="fishType">
                <xs:restriction base="xs:string">
                        <xs:enumeration value="Salmon"/>
                        <xs:enumeration value="Mullet"/>
                        <xs:enumeration value="Herring"/>
                        <xs:enumeration value="Sardines"/>
                        <xs:enumeration value="Sole"/>
                        <xs:enumeration value="Cod"/>
                        <xs:enumeration value="ColdFish"/>
                        <xs:enumeration value="Tope"/>
                        <xs:enumeration value="Tuna"/>
                        <xs:enumeration value="Swordfish"/>
                        <xs:enumeration value="Trout"/>
                        <xs:enumeration value="RedMullet"/>
                        <xs:enumeration value="Eel"/>
                </xs:restriction>
        </xs:simpleType>
        <!--MEAT_TYPE-->
        <xs:simpleType name="meatType">
                <xs:restriction base="xs:string">
                        <xs:enumeration value="Beef"/>
                        <xs:enumeration value="Pork"/>
                        <xs:enumeration value="Lamp"/>
                        <xs:enumeration value="Rabit"/>
                </xs:restriction>
        </xs:simpleType>
        <!--SEAFOOD_TYPE-->
        <xs:simpleType name="seafoodType">
                <xs:restriction base="xs:string">
```

```
                        <xs:enumeration value="Anchovies"/>
                        <xs:enumeration value="Cuttlefish"/>
                        <xs:enumeration value="Prowns"/>
                        <xs:enumeration value="Squid"/>
                        <xs:enumeration value="Mussels"/>
                        <xs:enumeration value="Octapus"/>
                </xs:restriction>
        </xs:simpleType>
        <!--OTHER_TYPE-->
        <xs:simpleType name="otherType">
                <xs:restriction base="xs:string">
                        <xs:enumeration value="Vegetarian"/>
                        <xs:enumeration value="Pasta"/>
                        <xs:enumeration value="Pizzas"/>
                        <xs:enumeration value="Salads"/>
                </xs:restriction>
        </xs:simpleType>
        <!--********************RESTAURANT_SERVICES***********************-->
        <!--orizw enumeration gia to sinolo twn restaurantServices-->
        <xs:simpleType name="restaurantServices">
                <xs:restriction base="xs:string">
                        <xs:enumeration value="delivery"/>
                        <xs:enumeration value="takeAway"/>
                        <xs:enumeration value="deliveryAndTakeAway"/>
                </xs:restriction>
        </xs:simpleType>
        <!--
**********************************PHONE_TYPE*************************************
********-->
        <xs:simpleType name="phoneType">
                <xs:restriction base="xs:string">
                        <xs:pattern value="[0-9]{8}"/>
                </xs:restriction>
        </xs:simpleType>
        <!--********************PROVINCES_AND_TOWNS**************************-->
        <!--Orizw enumeration gia tin eparxia kai poli: oi poleis tis Kiprou-->
        <xs:simpleType name="provincesAndTowns">
                <xs:restriction base="xs:string">
                        <xs:enumeration value="Nicosia"/>
                        <xs:enumeration value="Limassol"/>
                        <xs:enumeration value="Famagusta"/>
                        <xs:enumeration value="Larnaca"/>
                        <xs:enumeration value="Pafos"/>
                        <xs:enumeration value="Kerynia"/>
                </xs:restriction>
        </xs:simpleType>
    </xs:schema>
```

hotel-schema.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
<!--
**************************************HOTELS**************************************-->
<!--
*************************************************************************************-->
<xs:element name="hotel">
<xs:complexType mixed="true">
<xs:sequence>
<xs:element name="hotelCategory" type="hotelCategory"
minOccurs="0"/>
<xs:element name="hotelFacilities" type="hotelFacilities"
minOccurs="0"/>
<xs:element name="locationCategory" type="locationCategory"
minOccurs="0"/>
<xs:element name="hotelLocation" type="location"
minOccurs="0"/>
<xs:element name="distance" type="xs:decimal" minOccurs="0"/>

<xs:element name="links" type="linksType"/>
<xs:element name="details" type="serviceDetails" minOccurs="0"/>
<xs:element name="averagePrice" type="xs:decimal"
minOccurs="0"/>
</xs:sequence>
<xs:attribute name="type" type="fileType" use="required"/>
</xs:complexType>
</xs:element>
<!--******************LINKS_TYPE*************************-->
<xs:complexType name="linksType">
<xs:sequence>
<xs:element name="link" type="linkType" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
<!--******************LIK_TYPE*************************-->
<xs:complexType name="linkType">
<xs:attribute name="address" type="xs:string" use="required"/>
<xs:attribute name="position" type="xs:string"/>
</xs:complexType>
<!--******************INFO*********************************-->
<xs:complexType name="serviceDetails">
<xs:sequence>
<xs:element name="serviceName" type="xs:string"/>
<xs:element name="phones" type="phonesType"/>
</xs:sequence>
</xs:complexType>
<!--******************PHONES_TYPE*****************-->
<xs:complexType name="phonesType">
<xs:sequence>
<xs:element name="phone" type="phoneType" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
<!--******************HOTEL_FACILITIES*****************-->
<xs:complexType name="hotelFacilities">
<xs:sequence>
<xs:element name="generalFacilities" type="generalFacilities"
minOccurs="0"/>
<xs:element name="hotelActivities" type="hotelActivities" minOccurs="0"/>
<xs:element name="hotelServices" type="hotelServices" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
<!--GENERAL_FACILITIES-->
<xs:complexType name="generalFacilities">
<xs:choice>
<xs:element name="generalFacility" type="generalTypeFacility"
maxOccurs="unbounded"/>
</xs:choice>
```

```
	</xs:complexType>
	<!--HOTEL_ACTIVITIES-->
	<xs:complexType name="hotelActivities">
		<xs:choice>
			<xs:element name="hotelActivity" type="activityType"
maxOccurs="unbounded"/>
		</xs:choice>
	</xs:complexType>
	<!--HOTEL_SERVICES-->
	<xs:complexType name="hotelServices">
		<xs:choice>
			<xs:element name="hotelService" type="serviceType"
maxOccurs="unbounded"/>
		</xs:choice>
	</xs:complexType>
	<!--****************LOCATION***************-->
	<!--orizw to xaraktiristiko location pou mas dinei tin perioxi pou vrisketai ena stigmiotipo
ipiresias-->
	<xs:complexType name="location">
		<xs:sequence>
			<xs:element name="postalCode" type="xs:integer" minOccurs="0"/>
			<xs:element name="area" type="xs:string"/>
			<xs:element name="town" type="provincesAndTowns"/>
			<xs:element name="province" type="provincesAndTowns"/>
			<xs:element name="country" type="xs:string" fixed="Cyprus"/>
		</xs:sequence>
	</xs:complexType>
	<!--*****************HOTEL_CATEGORY*****************-->
	<xs:simpleType name="hotelCategory">
		<xs:restriction base="xs:string">
			<xs:enumeration value="*"/>
			<xs:enumeration value="**"/>
			<xs:enumeration value="***"/>
			<xs:enumeration value="****"/>
			<xs:enumeration value="*****"/>
		</xs:restriction>
	</xs:simpleType>
	<!--
**********************************PHONE_TYPE**************************************
********-->
	<xs:simpleType name="phoneType">
		<xs:restriction base="xs:string">
			<xs:pattern value="[0-9]{8}"/>
		</xs:restriction>
	</xs:simpleType>
	<!--
************************************ENUMERATIONS*********************************
***********-->
	<!--
*******************************************************************************
*********-->
	<!--FILE_TYPE-->
	<xs:simpleType name="fileType">
		<xs:restriction base="xs:string">
			<xs:enumeration value="linkFile"/>
			<xs:enumeration value="serviceFile"/>
		</xs:restriction>
	</xs:simpleType>
	<!--LOCATION_CATEGORY-->
	<xs:simpleType name="locationCategory">
		<xs:restriction base="xs:string">
			<xs:enumeration value="Coastal"/>
			<xs:enumeration value="Mountain"/>
			<xs:enumeration value="InTown"/>
		</xs:restriction>
	</xs:simpleType>
	<!--GENERAL_TYPE_FACILITIES-->
	<xs:simpleType name="generalTypeFacility">
		<xs:restriction base="xs:string">
```

```
                        <xs:enumeration value="Parking"/>
                        <xs:enumeration value="Restaurant"/>
                        <xs:enumeration value="Bar"/>
                        <xs:enumeration value="Shops at Hotel"/>
                        <xs:enumeration value="Heeting"/>
                </xs:restriction>
        </xs:simpleType>
        <!--ACTIVITY_TYPE-->
        <xs:simpleType name="activityType">
                <xs:restriction base="xs:string">
                        <xs:enumeration value="Swimming Pool"/>
                        <xs:enumeration value="Tennis-court"/>
                        <xs:enumeration value="Sauna"/>
                        <xs:enumeration value="Fitness Center"/>
                        <xs:enumeration value="Game Room"/>
                        <xs:enumeration value="Billiard"/>
                        <xs:enumeration value="Jacuzzi"/>
                </xs:restriction>
        </xs:simpleType>
        <!--SERVICE_TYPE-->
        <xs:simpleType name="serviceType">
                <xs:restriction base="xs:string">
                        <xs:enumeration value="Room-service"/>
                        <xs:enumeration value="Babysitting"/>
                        <xs:enumeration value="Laundry"/>
                        <xs:enumeration value="Barber/Beauty Shop"/>
                        <xs:enumeration value="Internet Services"/>
                </xs:restriction>
        </xs:simpleType>
        <!--PROVINCES_AND_TOWNS-->
        <!--Orizw enumeration gia tin eparxia kai poli: oi poleis tis Kiprou-->
        <xs:simpleType name="provincesAndTowns">
                <xs:restriction base="xs:string">
                        <xs:enumeration value="Nicosia"/>
                        <xs:enumeration value="Limassol"/>
                        <xs:enumeration value="Famagusta"/>
                        <xs:enumeration value="Larnaca"/>
                        <xs:enumeration value="Pafos"/>
                        <xs:enumeration value="Kerynia"/>
                </xs:restriction>
        </xs:simpleType>
</xs:schema>
```

bookshop-schema.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">
	<!--**********PHARMACY_SCHEMA**********-->
	<!--**********-->
	<xs:element name="bookshop">
		<xs:complexType mixed="true">
			<xs:sequence>
				<xs:element name="distance" type="xs:decimal" minOccurs="0"/>
				<xs:element name="links" type="linksType"/>
				<xs:element name="details" type="serviceDetails" minOccurs="0"/>
				<xs:element name="bookshopLocation" type="location" minOccurs="0"/>
				<xs:element name="bsServices" type="bsServicesType" minOccurs="0"/>
				<xs:element name="bookTypes" type="bookTypes" minOccurs="0"/>
			</xs:sequence>
			<xs:attribute name="type" type="fileType" use="required"/>
		</xs:complexType>
	</xs:element>
	<!--**********LINKS_TYPE**********-->
	<xs:complexType name="bsServicesType">
		<xs:sequence>
			<xs:element name="bsServiceType" type="bsServiceType" maxOccurs="5"/>
		</xs:sequence>
	</xs:complexType>
	<!--**********LINKS_TYPE**********-->
	<xs:complexType name="bookTypes">
		<xs:sequence>
			<xs:element name="bookType" type="bookType" maxOccurs="10"/>
		</xs:sequence>
	</xs:complexType>
	<!--**********LINKS_TYPE**********-->
	<xs:complexType name="linksType">
		<xs:sequence>
			<xs:element name="link" type="linkType" maxOccurs="unbounded"/>
		</xs:sequence>
	</xs:complexType>
	<!--**********LIK_TYPE**********-->
	<xs:complexType name="linkType">
		<xs:attribute name="address" type="xs:string" use="required"/>
		<xs:attribute name="position" type="xs:string"/>
	</xs:complexType>
	<!--**********INFO**********-->
	<xs:complexType name="serviceDetails">
		<xs:sequence>
			<xs:element name="serviceName" type="xs:string"/>
			<xs:element name="phones" type="phonesType"/>
		</xs:sequence>
	</xs:complexType>
	<!--**********PHONES_TYPE**********-->
	<xs:complexType name="phonesType">
		<xs:sequence>
			<xs:element name="phone" type="phoneType" maxOccurs="unbounded"/>
		</xs:sequence>
	</xs:complexType>
	<!--**********LOCATION**********-->
	<!--orizw to xaraktiristiko location pou mas dinei tin perioxi pou vrisketai ena stigmiotipo ipiresias-->
	<xs:complexType name="location">
		<xs:sequence>
			<xs:element name="postalCode" type="xs:integer" minOccurs="0"/>
```

```
                        <xs:element name="area" type="xs:string"/>
                        <xs:element name="town" type="provincesAndTowns"/>
                        <xs:element name="province" type="provincesAndTowns"/>
                        <xs:element name="country" type="xs:string" fixed="Cyprus"/>
                </xs:sequence>
        </xs:complexType>
        <!--
**********************************PHONE_TYPE**************************************
********-->
        <xs:simpleType name="phoneType">
                <xs:restriction base="xs:string">
                        <xs:pattern value="[0-9]{8}"/>
                </xs:restriction>
        </xs:simpleType>
        <!--FILE_TYPE-->
        <xs:simpleType name="fileType">
                <xs:restriction base="xs:string">
                        <xs:enumeration value="linkFile"/>
                        <xs:enumeration value="serviceFile"/>
                </xs:restriction>
        </xs:simpleType>
        <!--*********************PROVINCES_AND_TOWNS**************************-->
        <!--Orizw enumeration gia tin eparxia kai poli: oi poleis tis Kiprou-->
        <xs:simpleType name="provincesAndTowns">
                <xs:restriction base="xs:string">
                        <xs:enumeration value="Nicosia"/>
                        <xs:enumeration value="Limassol"/>
                        <xs:enumeration value="Famagusta"/>
                        <xs:enumeration value="Larnaca"/>
                        <xs:enumeration value="Pafos"/>
                        <xs:enumeration value="Kerynia"/>
                </xs:restriction>
        </xs:simpleType>
        <!--ENUMERATION: YesNo-->
        <xs:simpleType name="bsServiceType">
                <xs:restriction base="xs:string">
                        <xs:enumeration value="Typing"/>
                        <xs:enumeration value="Stationery"/>
                        <xs:enumeration value="Photocopies"/>
                        <xs:enumeration value="Bindings"/>
                        <xs:enumeration value="StudentNodes"/>
                </xs:restriction>
        </xs:simpleType>
        <!--ENUMERATION: YesNo-->
        <xs:simpleType name="bookType">
                <xs:restriction base="xs:string">
                        <xs:enumeration value="ComputerBooks"/>
                        <xs:enumeration value="ChildrenBooks"/>
                        <xs:enumeration value="Literature"/>
                        <xs:enumeration value="History"/>
                        <xs:enumeration value="StudentBooks"/>
                        <xs:enumeration value="HealthMindAndBody"/>
                        <xs:enumeration value="Science"/>
                        <xs:enumeration value="Cooking"/>
                        <xs:enumeration value="Calendars"/>
                        <xs:enumeration value="Magazines"/>
                </xs:restriction>
        </xs:simpleType>
</xs:schema>
```

bar-schema.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v2004 rel. 3 U (http://www.xmlspy.com) by Maria (UCY) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">
	<!--
**********PHARMACY_SCHEMA**********-->
	<!--
**********-->
	<xs:element name="bar">
		<xs:complexType mixed="true">
			<xs:sequence>
				<xs:element name="distance" type="xs:decimal" minOccurs="0"/>
				<xs:element name="links" type="linksType"/>
				<xs:element name="details" type="serviceDetails" minOccurs="0"/>
				<xs:element name="barLocation" type="location" minOccurs="0"/>
				<xs:element name="barServices" type="barServicesType" minOccurs="0"/>
			</xs:sequence>
			<xs:attribute name="type" type="fileType" use="required"/>
		</xs:complexType>
	</xs:element>
	<!--**********LINKS_TYPE**********-->
	<xs:complexType name="barServicesType">
		<xs:sequence>
			<xs:element name="barServiceType" type="barServiceType" maxOccurs="9"/>
		</xs:sequence>
	</xs:complexType>
	<!--**********LINKS_TYPE**********-->
	<xs:complexType name="linksType">
		<xs:sequence>
			<xs:element name="link" type="linkType" maxOccurs="unbounded"/>
		</xs:sequence>
	</xs:complexType>
	<!--**********LIK_TYPE**********-->
	<xs:complexType name="linkType">
		<xs:attribute name="address" type="xs:string" use="required"/>
		<xs:attribute name="position" type="xs:string"/>
	</xs:complexType>
	<!--**********INFO**********-->
	<xs:complexType name="serviceDetails">
		<xs:sequence>
			<xs:element name="serviceName" type="xs:string"/>
			<xs:element name="phones" type="phonesType"/>
		</xs:sequence>
	</xs:complexType>
	<!--**********PHONES_TYPE**********-->
	<xs:complexType name="phonesType">
		<xs:sequence>
			<xs:element name="phone" type="phoneType" maxOccurs="unbounded"/>
		</xs:sequence>
	</xs:complexType>
	<!--**********LOCATION**********-->
	<!--orizw to xaraktiristiko location pou mas dinei tin perioxi pou vrisketai ena stigmiotipo ipiresias-->
	<xs:complexType name="location">
		<xs:sequence>
			<xs:element name="postalCode" type="xs:integer" minOccurs="0"/>
			<xs:element name="area" type="xs:string"/>
			<xs:element name="town" type="provincesAndTowns"/>
			<xs:element name="province" type="provincesAndTowns"/>
			<xs:element name="country" type="xs:string" fixed="Cyprus"/>
		</xs:sequence>
	</xs:complexType>
	<!--
```

```
**********************************PHONE_TYPE**************************************
********-->
	<xs:simpleType name="phoneType">
		<xs:restriction base="xs:string">
			<xs:pattern value="[0-9]{8}"/>
		</xs:restriction>
	</xs:simpleType>
	<!--FILE_TYPE-->
	<xs:simpleType name="fileType">
		<xs:restriction base="xs:string">
			<xs:enumeration value="linkFile"/>
			<xs:enumeration value="serviceFile"/>
		</xs:restriction>
	</xs:simpleType>
	<!--**********************PROVINCES_AND_TOWNS**************************-->
	<!--Orizw enumeration gia tin eparxia kai poli: oi poleis tis Kiprou-->
	<xs:simpleType name="provincesAndTowns">
		<xs:restriction base="xs:string">
			<xs:enumeration value="Nicosia"/>
			<xs:enumeration value="Limassol"/>
			<xs:enumeration value="Famagusta"/>
			<xs:enumeration value="Larnaca"/>
			<xs:enumeration value="Pafos"/>
			<xs:enumeration value="Kerynia"/>
		</xs:restriction>
	</xs:simpleType>
	<!--ENUMERATION: YesNo-->
	<xs:simpleType name="barServiceType">
		<xs:restriction base="xs:string">
			<xs:enumeration value="FoodServed"/>
			<xs:enumeration value="SmokeFreeArea"/>
			<xs:enumeration value="OpenAtNoon"/>
			<xs:enumeration value="TV"/>
			<xs:enumeration value="OutDoorsSeating"/>
		</xs:restriction>
	</xs:simpleType>
</xs:schema>
```

cafe-schema.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v2004 rel. 3 U (http://www.xmlspy.com) by Maria (UCY) -->
<!--
**************************************RESTAURANTS_SCHEMA*************************
**************-->
<!--
*****************************************************************************************
*************************-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
	<!--***************RESTAURANT_ELEMENT**************-->
	<xs:element name="cafe">
		<xs:complexType>
			<xs:sequence>
				<xs:element name="foodCategories" type="foodCategories"
minOccurs="0"/>
				<xs:element name="cafeLocation" type="location" minOccurs="0"/>
				<xs:element name="drinkCategories" type="drinkCategories"
minOccurs="0"/>
				<xs:element name="distance" type="xs:decimal" minOccurs="0"/>
				<xs:element name="links" type="linksType"/>
				<xs:element name="details" type="serviceDetails" minOccurs="0"/>
			</xs:sequence>
			<xs:attribute name="type" type="fileType" use="required"/>
		</xs:complexType>
	</xs:element>
	<!--******************LINKS_TYPE**************************-->
	<xs:complexType name="linksType">
		<xs:sequence>
			<xs:element name="link" type="linkType" maxOccurs="unbounded"/>
		</xs:sequence>
	</xs:complexType>
	<!--******************LIK_TYPE**************************-->
	<xs:complexType name="linkType">
		<xs:attribute name="address" type="xs:string" use="required"/>
		<xs:attribute name="position" type="xs:string"/>
	</xs:complexType>
	<!--******************INFO**********************************-->
	<xs:complexType name="serviceDetails">
		<xs:sequence>
			<xs:element name="serviceName" type="xs:string"/>
			<xs:element name="phones" type="phonesType"/>
		</xs:sequence>
	</xs:complexType>
	<!--******************PHONES_TYPE*****************-->
	<xs:complexType name="phonesType">
		<xs:sequence>
			<xs:element name="phone" type="phoneType" maxOccurs="unbounded"/>
		</xs:sequence>
	</xs:complexType>
	<!--******************LOCATION***************-->
	<!--orizw to xaraktiristiko location pou mas dinei tin perioxi pou vrisketai ena stigmiotipo
ipiresias-->
	<xs:complexType name="location">
		<xs:sequence>
			<xs:element name="postalCode" type="xs:integer" minOccurs="0"/>
			<xs:element name="area" type="xs:string"/>
			<xs:element name="town" type="provincesAndTowns"/>
			<xs:element name="province" type="provincesAndTowns"/>
			<xs:element name="country" type="xs:string" fixed="Cyprus"/>
		</xs:sequence>
	</xs:complexType>
	<!--******************DRINK_CATEGORY*********************-->
	<xs:complexType name="drinkCategories">
		<xs:sequence>
			<xs:element name="drinkCategoryType" type="drinkCategoryType"
maxOccurs="3"/>
```

```
                </xs:sequence>
        </xs:complexType>
        <!--******************DRINK_CATEGORY*********************-->
        <xs:complexType name="foodCategories">
                <xs:sequence>
                        <xs:element         name="foodCategoryType"         type="foodCategoryType"
maxOccurs="3"/>
                </xs:sequence>
        </xs:complexType>
        <!--
************************************ENUMERATIONS*********************************
*******-->
        <!--
*****************************************************************************************
**************-->
        <!--FILE_TYPE-->
        <xs:simpleType name="fileType">
                <xs:restriction base="xs:string">
                        <xs:enumeration value="linkFile"/>
                        <xs:enumeration value="serviceFile"/>
                </xs:restriction>
        </xs:simpleType>
        <!--
*********************************PHONE_TYPE*************************************
*******-->
        <xs:simpleType name="phoneType">
                <xs:restriction base="xs:string">
                        <xs:pattern value="[0-9]{8}"/>
                </xs:restriction>
        </xs:simpleType>
        <!--********************PROVINCES_AND_TOWNS**************************-->
        <!--Orizw enumeration gia tin eparxia kai poli: oi poleis tis Kiprou-->
        <xs:simpleType name="provincesAndTowns">
                <xs:restriction base="xs:string">
                        <xs:enumeration value="Nicosia"/>
                        <xs:enumeration value="Limassol"/>
                        <xs:enumeration value="Famagusta"/>
                        <xs:enumeration value="Larnaca"/>
                        <xs:enumeration value="Pafos"/>
                        <xs:enumeration value="Kerynia"/>
                </xs:restriction>
        </xs:simpleType>
        <!--********************FOOD_CATEGORY_TYPE**************************-->
        <xs:simpleType name="foodCategoryType">
                <xs:restriction base="xs:string">
                        <xs:enumeration value="Sandwiches"/>
                        <xs:enumeration value="Crepes"/>
                        <xs:enumeration value="Tosts"/>
                </xs:restriction>
        </xs:simpleType>
        <!--********************DRINK_CATEGORY_TYPE**************************-->
        <xs:simpleType name="drinkCategoryType">
                <xs:restriction base="xs:string">
                        <xs:enumeration value="hots"/>
                        <xs:enumeration value="colds"/>
                        <xs:enumeration value="alcoholic"/>
                </xs:restriction>
        </xs:simpleType>
</xs:schema>
```

copycenter-schema.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v2004 rel. 3 U (http://www.xmlspy.com) by Maria (UCY) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
	<!--
**********PHARMACY_SCHEMA**********
-->
	<!--
**********
-->
	<xs:element name="copycenter">
		<xs:complexType mixed="true">
			<xs:sequence>
				<xs:element name="distance" type="xs:decimal" minOccurs="0"/>
				<xs:element name="links" type="linksType"/>
				<xs:element name="details" type="serviceDetails" minOccurs="0"/>
				<xs:element name="copycenterLocation" type="location"
minOccurs="0"/>
				<xs:element name="ccServices" type="ccServicesType"
minOccurs="0"/>
			</xs:sequence>
			<xs:attribute name="type" type="fileType" use="required"/>
		</xs:complexType>
	</xs:element>
	<!--**********LINKS_TYPE**********-->
	<xs:complexType name="ccServicesType">
		<xs:sequence>
			<xs:element name="ccServiceType" type="ccServiceType" maxOccurs="9"/>
		</xs:sequence>
	</xs:complexType>
	<!--**********LINKS_TYPE**********-->
	<xs:complexType name="linksType">
		<xs:sequence>
			<xs:element name="link" type="linkType" maxOccurs="unbounded"/>
		</xs:sequence>
	</xs:complexType>
	<!--**********LIK_TYPE**********-->
	<xs:complexType name="linkType">
		<xs:attribute name="address" type="xs:string" use="required"/>
		<xs:attribute name="position" type="xs:string"/>
	</xs:complexType>
	<!--**********INFO**********-->
	<xs:complexType name="serviceDetails">
		<xs:sequence>
			<xs:element name="serviceName" type="xs:string"/>
			<xs:element name="phones" type="phonesType"/>
		</xs:sequence>
	</xs:complexType>
	<!--**********PHONES_TYPE**********-->
	<xs:complexType name="phonesType">
		<xs:sequence>
			<xs:element name="phone" type="phoneType" maxOccurs="unbounded"/>
		</xs:sequence>
	</xs:complexType>
	<!--**********LOCATION**********-->
	<!--orizw to xaraktiristiko location pou mas dinei tin perioxi pou vrisketai ena stigmiotipo
ipiresias-->
	<xs:complexType name="location">
		<xs:sequence>
			<xs:element name="postalCode" type="xs:integer" minOccurs="0"/>
			<xs:element name="area" type="xs:string"/>
			<xs:element name="town" type="provincesAndTowns"/>
			<xs:element name="province" type="provincesAndTowns"/>
			<xs:element name="country" type="xs:string" fixed="Cyprus"/>
		</xs:sequence>
	</xs:complexType>
	<!--
```

```
**********************************PHONE_TYPE**************************************
*******-->
        <xs:simpleType name="phoneType">
                <xs:restriction base="xs:string">
                        <xs:pattern value="[0-9]{8}"/>
                </xs:restriction>
        </xs:simpleType>
        <!--FILE_TYPE-->
        <xs:simpleType name="fileType">
                <xs:restriction base="xs:string">
                        <xs:enumeration value="linkFile"/>
                        <xs:enumeration value="serviceFile"/>
                </xs:restriction>
        </xs:simpleType>
        <!--**********************PROVINCES_AND_TOWNS**************************-->
        <!--Orizw enumeration gia tin eparxia kai poli: oi poleis tis Kiprou-->
        <xs:simpleType name="provincesAndTowns">
                <xs:restriction base="xs:string">
                        <xs:enumeration value="Nicosia"/>
                        <xs:enumeration value="Limassol"/>
                        <xs:enumeration value="Famagusta"/>
                        <xs:enumeration value="Larnaca"/>
                        <xs:enumeration value="Pafos"/>
                        <xs:enumeration value="Kerynia"/>
                </xs:restriction>
        </xs:simpleType>
        <!--ENUMERATION: YesNo-->
        <xs:simpleType name="ccServiceType">
                <xs:restriction base="xs:string">
                        <xs:enumeration value="Typing"/>
                        <xs:enumeration value="Stationery"/>
                        <xs:enumeration value="Bindings"/>
                        <xs:enumeration value="StudentNotes"/>
                        <xs:enumeration value="Slides"/>
                        <xs:enumeration value="GraphicDesign"/>
                        <xs:enumeration value="FullColorPrinting"/>
                        <xs:enumeration value="DigitalPrinting"/>
                        <xs:enumeration value="MailingLabelingAndStuffing"/>
                </xs:restriction>
        </xs:simpleType>
    </xs:schema>
```