ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ
ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

Skyline Συστήματα Προτίμησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

Λυδίας Κ. Μπαλαφούτη

Επιβλέπων : Δημήτριος Ασκούνης
Καθηγητής Ε.Μ.Π

Αθήνα, Οκτώβριος 2022

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ
ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

Skyline Συστήματα Προτίμησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

Λυδίας Κ. Μπαλαφούτη

Επιβλέπων : Δημήτριος Ασκούνης
Καθηγητής Ε.Μ.Π

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 12/10/2022

.....	.....	.....
Δημήτριος Ασκούνης	Ιωάννης Ψαρράς	Χρυσόστομος Δούκας
Καθηγητής Ε.Μ.Π	Καθηγητής Ε.Μ.Π	Αναπληρωτής Καθηγητής Ε.Μ.Π

Αθήνα, Οκτώβριος 2022

.....

Λυδία, Κ Μπαλαφούτη

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

Περίληψη

Στόχος της παρούσας διπλωματικής εργασίας είναι η μελέτη του βαθμού επίδρασης του skyline συνόλου στα επίπεδα ικανοποίησης των χρηστών από τη χρήση των συστημάτων προτιμήσεων ως προς το προτεινόμενο περιεχόμενο τους. Η διπλωματική αποτελείται από δύο μέρη ένα βιβλιογραφικό και ένα πειραματικό.

Για τις ανάγκες της διπλωματικής εργασίας πραγματοποιήθηκε βιβλιογραφική ανασκόπηση τόσο για τη μελέτη των διάφορων τεχνικών υπολογισμού του skyline συνόλου με χρήση skyline αλγόριθμων, όσο και για τις κατηγορίες των συστημάτων προτιμήσεων που υπάρχουν ανάλογα με την τεχνική που χρησιμοποιούν για να πραγματοποιήσουν προτάσεις νέων αντικειμένων σε ένα χρήστη. Στη συνέχεια, πραγματοποιήθηκε πειραματική διαδικασία για την οποία με εφαρμογή του κατάλληλου skyline αλγόριθμου υπολογίστηκε το skyline σύνολο μιας βάσης δεδομένων με πληροφορίες σχετικά με ενοικιαζόμενα δωμάτια και κατασκευάστηκε σύστημα προτιμήσεων που βασίζεται στο περιεχόμενο (content based). Έπειτα, ο εκάστοτε χρήστης επέλεξε μέσα από μια σειρά φίλτρων το ιδανικό διαμέρισμα σύμφωνα με τις προτιμήσεις του και το σύστημα προτιμήσεων που κατασκευάστηκε, κάνοντας χρήση της ομοιότητας συνημίτονου (cosine similarity), πρότεινε στους χρήστες αντικείμενα σε δύο ομάδες. Στη πρώτη, έκανε χρήση του skyline συνόλου ενώ στη δεύτερη, χρησιμοποίησε ολόκληρη τη βάση χωρίς αυτή να έχει υποστεί κάποια επεξεργασία. Τέλος, οι χρήστες κλήθηκαν να βαθμολογήσουν τις δύο ομάδες προτεινόμενων αντικειμένων ως προς το επίπεδο ικανοποίησης των προτάσεων από τις οποίες αποτελούνται.

Τα αποτελέσματα της έρευνας δείχνουν ότι οι skyline αλγόριθμοι βελτιώνουν σημαντικά τα επίπεδα ικανοποίησης των χρηστών ως προς τις προτάσεις που τους παρέχει ένα σύστημα προτιμήσεων. Το σύστημα προτιμήσεων που έκανε χρήση του skyline συνόλου παρείχε αποτελέσματα που ταίριαζαν περισσότερο στις προτιμήσεις του χρήστη, με το μεγαλύτερο ποσοστό από αυτούς να δηλώνει ότι στο σύνολο των προτάσεων που δέχτηκαν από τα δύο συστήματα το απλό σύστημα προτιμήσεων είχε περισσότερα αποτελέσματα που δεν βρήκαν καθόλου ικανοποιητικά.

Λέξεις Κλειδιά: λήψη αποφάσεων με πολλαπλά κριτήρια, συστήματα προτίμησης βασισμένα στο περιεχόμενο, skyline σύνολο, ικανοποίηση χρήστη, μεμονωμένη μελέτη περίπτωσης.

Abstract

The purpose of this research is to study the effect of recommending objects from the skyline set on content based recommender systems as far as the resulted user satisfaction is concerned. The thesis consists of two parts, one bibliographic and one experimental.

For the needs of the thesis, a literature review was initially carried out both for the study of the different techniques for calculating the skyline set, as well as for the categories of recommender systems that exist, depending on the technique used to make suggestions of new objects to a user. Then, an experimental procedure was carried out for which, by applying the appropriate skyline algorithm, the skyline set of a database with information about rooms for rent was calculated and a content-based recommender system was constructed. Each user chose through a series of filters the ideal apartment according to their preferences and the recommender system mentioned above, using the cosine similarity metric, proposed to them objects in two groups. In the first, it made use of the skyline set while in the second it used the whole database without any further processing of the data. Finally, users were asked to rate the two groups of proposed objects in terms of the level of satisfaction of the propositions they comprise.

The results of the research show that skyline algorithms significantly improve user satisfaction when it comes to the suggestions provided by a recommender system. The recommender system using the skyline set provided results that matched more closely the user's preferences, with the largest percentage of users stating that out of all the recommendations received from the two systems, the recommender that used the full database had more results that they did not find satisfactory at all.

Keywords: multiple criteria decision making, content-based recommender systems, skyline set, user satisfaction, single-case study.

Πίνακας Περιεχομένων

Πίνακας Εικόνων

1 Εισαγωγή

Με την αύξηση του όγκου δεδομένων σε όλους τους τομείς έγινε επιτακτική η ανάγκη καλύτερης διαχείρισης της πληροφορίας προκειμένου η αναζήτηση ενός χρήστη να επιστρέφει αποτελέσματα ουσιαστικά προς αυτόν και τις προτιμήσεις του. Για αυτό το σκοπό δημιουργήθηκαν τα συστήματα προτιμήσεων. Πρόκειται για τεχνικές που λαμβάνουν υπ' όψη τις βαθμολογίες ή το περιεχόμενο των αντικειμένων που ο χρήστης έχει ήδη έρθει σε επαφή και τις συμπεριλαμβάνουν στις τελικές προτάσεις που κάνουν προς αυτόν. Με αυτό τον τρόπο η ικανοποίηση του εκάστοτε χρήστη σε μια αναζήτηση του έχει αυξηθεί σημαντικά. Παρ' όλα αυτά, καθώς το πλήθος των δεδομένων συνεχίζει να αυξάνεται, ακόμα και αυτές οι μέθοδοι δεν προσφέρουν τα επίπεδα ικανοποίησης που επιθυμούμε. Αυτό οδηγεί στην εύρεση νέων συνδυαστικών μεθόδων για τη καλύτερη διαχείριση των δεδομένων σε ένα σύστημα προτιμήσεων. Οι skyline αλγόριθμοι επιτυγχάνουν τον διαχωρισμό των σημαντικότερων αντικειμένων μέσα από ένα σύνολο δεδομένων. Το skyline σύνολο υπολογίζεται βάσει γενικευμένων προτιμήσεων χωρίς να καθίσταται απαραίτητη η αυστηρή δήλωση προτιμήσεων από τον χρήστη. Μας ενδιαφέρει λοιπόν να εξετάσουμε κατά πόσο η χρήση αυτών των αλγορίθμων σε μια βάση δεδομένων που χρησιμοποιείται από ένα σύστημα προτιμήσεων επηρεάζει τα επίπεδα ικανοποίησης των χρηστών που το χρησιμοποιούν.

1.1 Στόχος της Διπλωματικής

Στόχος της παρούσας διπλωματικής εργασίας είναι να εξετάσει κατά πόσο η χρήση ενός skyline συνόλου επηρεάζει τα αντικείμενα που παρέχει ένα σύστημα προτιμήσεων στον χρήστη. Πιο συγκεκριμένα μας ενδιαφέρει ο βαθμός ικανοποίησης των χρηστών από τις προτάσεις ενός συστήματος προτιμήσεων που κάνει χρήση του skyline συνόλου και από ένα που χρησιμοποιεί τη βάση δεδομένων χωρίς αυτή να έχει υποστεί κάποια επεξεργασία. Στη συνέχεια θέλουμε να συγκρίνουμε τα δύο αυτά αποτελέσματα για να συμπεράνουμε εάν και κατά πόσο τελικά ένα σύστημα προτιμήσεων επιφέρει πιο ικανοποιητικά αποτελέσματα όταν χρησιμοποιεί το skyline σύνολο.

1.2 Δομή της Διπλωματικής

Η ενότητα 2 αποτελεί το θεωρητικό υπόβαθρο της παρούσας διπλωματικής. Αναλύεται το skyline σύνολο και περιγράφεται μια σειρά από αλγόριθμους που χρησιμοποιούνται για τον υπολογισμό του με ενδεικτικά παραδείγματα λειτουργίας. Στη συνέχεια περιγράφονται τα συστήματα προτιμήσεων, αναλύονται οι κύριες κατηγορίες τους και περιγράφονται οι

αλγόριθμοι που χρησιμοποιούνται. Τέλος γίνεται αναφορά σε μια σειρά από μεθόδους που συνδυάζουν τα συστήματα προτιμήσεων με τους skyline αλγορίθμους. Στην ενότητα 3 αναλύεται το πειραματικό μέρος της διπλωματικής. Περιγράφεται η αρχιτεκτονική του συστήματος η επεξεργασία των δεδομένων και η πειραματική διαδικασία. Τέλος στις ενότητες 4 και 5 αναλύονται τα αποτελέσματα του πειράματος καθώς και οι περιορισμοί και τα όρια του.

2 Θεωρητικό Υπόβαθρο

Στο παρόν κεφάλαιο πραγματοποιείται αρχικά βιβλιογραφική ανασκόπηση των αλγόριθμων υπολογισμού του skyline συνόλου. Στη συνέχεια αναλύονται οι μέθοδοι που υπάρχουν για δημιουργία συστήματος προτιμήσεων και κατατάσσονται σύμφωνα με την προσέγγιση τους. Τέλος αναφέρονται ορισμένες έρευνες που συνδυάζουν τα συστήματα προτιμήσεων με τους skyline αλγορίθμους και τα αποτελέσματα αυτών.

2.1 Skyline

Τα skyline ερωτήματα (skyline queries) γεφυρώνουν το χάσμα ανάμεσα στις παραδοσιακές βάσεις δεδομένων και τις βάσεις δεδομένων πολυμέσων (multimedia databases). Στα πρώτα ένα ερώτημα επιστρέφει ένα σύνολο από μη ταξινομημένα δεδομένα ενώ στις βάσεις δεδομένων πολυμέσων επιστρέφεται ένα βαθμολογημένο σύνολο αντικειμένων. Ο υπολογισμός του skyline συνόλου βασίζεται σε γενικευμένες αποδοχές προτιμήσεων χωρίς να απαιτείται προσδιορισμός προτίμησης από τον χρήστη. Το skyline σύνολο αποτελείται από τα κυρίαρχα αντικείμενα μιας βάσης δεδομένων, περιέχει δηλαδή όλα τα αντικείμενα ενός συνόλου που δεν κυριαρχούνται από κανένα άλλο αντικείμενο σύμφωνα με την Pareto έννοια κυριαρχίας [2], κατά την οποία ένα αντικείμενο είναι κυρίαρχο ως προς ένα άλλο εάν είναι καλύτερο σε τουλάχιστον ένα από τα χαρακτηριστικά τους και ισοδύναμο σε όλα τα υπόλοιπα χαρακτηριστικά. Αρχικά η έννοια της Pareto κυριαρχίας λειτουργούσε μόνο για αριθμητικά δεδομένα, αλλά στη συνέχεια επεκτάθηκε προκειμένου να μπορέσει να διαχειριστεί και κατηγορικά δεδομένα. Προκειμένου να μπορέσουμε να κατανοήσουμε τη σημασία και τη συνεισφορά των skyline συνόλων είναι σημαντικό να αναφέρουμε κάποιους ορισμούς που σχετίζονται με το αντικείμενο.

2.1.1 Χρήσιμοι ορισμοί

Έστω $A = \{A_1,\ldots,A_d\}$ ένα σύνολο από d χαρακτηριστικά που συσχετίζονται με τα αντικείμενα μιας βάσης D, η οποία περιέχει μια σειρά από αντικείμενα κάθε ένα από τα οποία

αποτελείται από d χαρακτηριστικά σύμφωνα με το A. Η τιμή του αντικειμένου o για το χαρακτηριστικό i θα συμβολίζεται ως $v_i(o)$. Θα λέμε ότι ένα αντικείμενο o_1 είναι κυρίαρχο ως προς ένα άλλο o_2 σε σχέση με το χαρακτηριστικό $i \in A$, όταν η τιμή του χαρακτηριστικού i για το αντικείμενο o_1 είναι προτιμητέα σε σχέση με την τιμή του ίδιου χαρακτηριστικό στο o_2 και θα συμβολίζουμε με $o_1 >_i o_2$. Παρόμοια εάν οι τιμές των o_1 και o_2 είναι ίσες για το χαρακτηριστικό i συμβολίζουμε με $o_1 \approx_i o_2$, ενώ εάν η τιμή του o_1 είναι καλύτερη ή ίση με την τιμή του o_2 για το χαρακτηριστικό i συμβολίζουμε με $o_1 \geq_i o_2$.

Με βάση τους ορισμούς που δώσαμε για την προτίμηση ενός αντικειμένου μπορούμε να ορίσουμε την έννοια της κυριαρχίας όπως φαίνεται παρακάτω.

Σχέση Κυριαρχίας

Έστω ότι έχουμε δύο αντικείμενα της βάσης δεδομένων D, o_1 και o_2. Λέμε ότι το o_1 είναι κυρίαρχο ως προς το o_2 όταν είναι κυρίαρχο τουλάχιστον ως προς ένα από τα χαρακτηριστικά του σε σχέση με το o_2 και ίσο σε όλα τα υπόλοιπα. Πιο συγκεκριμένα:

$$o_1 > o_2 \Leftrightarrow \exists i \in [1,d]: v_i(o_1) >_i v_i(o_2) \wedge \forall j \in [1,d] - i : v_j(o_1) \geq_j v_j(o_1)$$

Η σχέση κυριαρχίας μπορεί να οριστεί τόσο για αριθμητικά όσο και για κατηγορικά χαρακτηριστικά. Ανάλογα με το ποια στοιχεία μας ενδιαφέρει να κρατήσουμε στο τελικό μας σύνολο η σχέση κυριαρχίας μπορεί να οριστεί κάθε φορά με διαφορετικό τρόπο. Στη σχέση κυριαρχίας ισχύει η μεταβατικότητα σύμφωνα με την οποία για τρία αντικείμενα o_1, o_2 και o_3, εάν $o_1 > o_2$ και $o_2 > o_3$, τότε ισχύει ότι $o_1 > o_3$.

Ασύγκριτα αντικείμενα

Λέμε ότι δυο αντικείμενα o_1 και o_2 είναι ασύγκριτα μεταξύ τους, όταν το o_1 δεν κυριαρχεί το o_2 και το o_2 δεν κυριαρχεί το o_1.

2.1.2 Skyline Σύνολο

Το skyline σύνολο αποτελείται από όλα τα αντικείμενα μιας βάσης δεδομένων D τα οποία δεν κυριαρχούνται από κανένα άλλο αντικείμενο. Πιο συγκεκριμένα:

$$Sky_{Set} = \{o_1 \in D | \neg \exists\, o_2 \in D : o_2 > o_1\}$$

Παρατηρούμε δηλαδή ότι όλα τα αντικείμενα που περιέχονται στο skyline σύνολο είναι ασύγκριτα μεταξύ τους και συνεπώς δεν μπορούμε να γνωρίζουμε ποιο από αυτά είναι καλύτερο για έναν χρήστη. Μια επιπλέον ενδιαφέρουσα ιδιότητα των συστημάτων συστάσεων είναι ότι οποιοδήποτε σύνολο κριτηρίων αξιολόγησης που προκύπτει από τις προτιμήσεις του εκάστοτε χρήστη μπορεί να δημιουργήσει μια μονότονη συνάρτηση. Συνεπώς ανεξάρτητα από τη σημασία που έχουν για έναν χρήστη οι τιμές των διάφορων χαρακτηριστικών ενός αντικειμένου πάντα θα υπάρχουν στο skyline σύνολο τα πιο ευνοϊκά αντικείμενα σε κάθε ερώτημα που θέτει.

Ένα από τα σημαντικότερα μειονεκτήματα κατά τον υπολογισμό του skyline συνόλου είναι ότι καταλήγει να περιέχει μεγάλο αριθμό αντικειμένων, ένα φαινόμενο που είναι γνωστό ως η κατάρα των πολλών διαστάσεων (curse of dimensionality). Ο μεγάλος αριθμός δεδομένων ότι τα σύνολα ορίζοντα συνήθως καταλήγουν να είναι όμορφα μεγάλο λόγω της επίδρασης αυτού που είναι γνωστό ως κατάρα της διάστασης. Με την αύξηση του αριθμού των διαστάσεων που θέλουμε ο skyline αλγόριθμος να λάβει υπ' όψη του, όλο και περισσότερα αντικείμενα γίνονται ασύγκριτα και καταλήγουν στον skyline σύνολο. Στο [3] παρουσιάζονται τα ποσοστά των αντικειμένων που ανήκουν στο skyline σύνολο σε σχέση με ολόκληρη τη βάση δεδομένων. Για ανεξάρτητες βάσεις δεδομένων όταν ο αριθμός των διαστάσεων είναι μεγαλύτερος του 20, πάνω από το 90% των αντικειμένων του συνόλου αντιστοιχούν στο skyline σύνολο, καθιστώντας το μη διαχειρίσιμο για τους περισσότερους χρήστες.

2.1.3 Skyline Αλγόριθμοι

Παρακάτω γίνεται αναφορά στους πιο χαρακτηριστικούς αλγορίθμους που χρησιμοποιούνται για τον υπολογισμό του skyline συνόλου που περιεγράφηκε στην προηγούμενη ενότητα. Οι skyline αλγόριθμοι χωρίζονται σε δύο κύριες κατηγορίες. Η πρώτη αποτελείται από τους αλγόριθμους που δεν βασίζονται σε δείκτες για τον υπολογισμό του skyline συνόλου (non-index based) και περιλαμβάνουν μεταξύ άλλων τους Block Nested Loop, Divide and Conquer και Bitmap. Η δεύτερη κατηγόρια αποτελείται από αλγόριθμους που κάνουν χρήση δεικτών για τον υπολογισμό του skyline συνόλου όπως οι Nearest Neigbour και Branch and bound που θα αναλυθούν παρακάτω. Στη συνέχεια θα αναλυθούν και κάποιες επιπλεόν κατηγορίες αλγορίθμων για τον υπολογισμό ενός skyline συνόλου.

Block Nested Loop

Ο αλγόριθμος Block Nested Loop [4] είναι από τους πιο απλούς αλγορίθμους για τον υπολογισμό ενός skyline συνόλου. Η βασική ιδέα αυτού του αλγορίθμου είναι να συγκρίνουμε κάθε αντικείμενο p της βάσης με όλα τα άλλα αντικείμενα. Εάν το p δεν κυριαρχείται από κανένα άλλο αντικείμενο τότε προστίθεται στη λίστα με τα skyline αντικείμενα διαφορετικά απορρίπτεται. Ο συγκεκριμένος αλγόριθμος ανήκει στη κατηγορία των αφελών αλγορίθμων (naive algorithms) καθώς αποτελείται από διπλό for-loop. Αναλυτικά ο αλγόριθμος αποτελείται από τα εξής βήματα.

1. Αρχικά η λίστα είναι κενή. Προστίθεται σε αυτή το πρώτο αντικείμενο της βάσης.
2. Σύμφωνα με τη σχέση κυριαρχίας που έχουμε ορίσει συγκρίνουμε όλα τα στοιχεία της βάσης με αυτά της λίστας. Για κάθε αντικείμενο p υπάρχουν τρείς περιπτώσεις:
    - Εάν το αντικείμενο p κυριαρχείται από οποιοδήποτε αντικείμενο της λίστας απορρίπτεται καθώς δεν μπορεί να ανήκει στο skyline σύνολο.
    - Εάν το αντικείμενο p είναι κυρίαρχο σε ένα ή περισσότερα αντικείμενα της λίστας, τότε αυτά τα αντικείμενα αφαιρούνται από τη λίστα και απορρίπτονται καθώς δεν μπορούν πλέον να ανήκουν στο skyline σύνολο και έπειτα το αντικείμενο p προστίθεται στη λίστα.
    - Εάν το αντικείμενο p δεν μπορεί να συγκριθεί με κανένα άλλο αντικείμενο της λίστας ως προς τη σχέση κυριαρχίας προστίθεται στη λίστα.
3. Η διαδικασία επαναλαμβάνεται μέχρι να έχουν συγκριθεί όλα τα αντικείμενα της βάσης.

Στο τέλος του αλγορίθμου επιστρέφεται η λίστα η οποία περιέχει όλα τα ασύγκριτα μεταξύ τους στοιχεία δηλαδή το skyline σύνολο. Η πολυπλοκότητα του αλγορίθμου στη χειρότερη περίπτωση είναι $O(N^2)$ εξαιτίας του διπλού for-loop.

Sort Filter Skyline

Ο συγκεκριμένος αλγόριθμος [5] αποτελεί επέκταση του Block Nested Loop. Πρόκειται για έναν προοδευτικό (progressive) αλγόριθμο καθώς πραγματοποιείται στην αρχή η ταξινόμηση των αντικειμένων σε αύξουσα σειρά με βάση κάποια μονότονη συνάρτηση προτίμησης (preference function). Η ταξινόμηση εξασφαλίζει ότι στη περίπτωση που ένα αντικείμενο κυριαρχεί ένα άλλο θα χρησιμοποιηθεί πρώτο για σύγκριση της κυριαρχίας με κάποιο άλλο αντικείμενο και έτσι θα μειωθεί σημαντικά ο αριθμός των συγκρίσεων που πραγματοποιούνται. Η πιο χαρακτηριστική μονότονη συνάρτηση που χρησιμοποιείται για

την ταξινόμηση είναι το άθροισμα των κανονικοποιημένων συντεταγμένων όλων των διαστάσεων του αντικειμένου.

Divide and Conquer

Ο αλγόριθμος Διαίρει και Βασίλευε (Divide and Conquer) [4] βασίζεται στον διαχωρισμού των αντικειμένων σε μικρότερα τμήματα αναδρομικά και τον επιμέρους υπολογισμό των skyline αντικειμένων. Ο διαχωρισμός πραγματοποιείται υπολογίζοντας τον μέσο (median) κάθε διάστασης. Το τελικό skyline σύνολο υπολογίζεται από τη συνένωση των επιμέρους skyline συνόλων που έχουν υπολογιστεί. Στις εικόνες εμφανίζονται δυο παραδείγματα εκτέλεσης του αλγορίθμου όπως παρουσιάζονται στο [1]. Στη πρώτη εικόνα υπολογίζεται ο μέσος για δύο διαστάσεις. Βλέπουμε ότι τα αντικείμενα στο τμήμα $S_{2,2}$ δεν θα συμπεριληφθούν στον υπολογισμό του skyline συνόλου καθώς όλα κυριαρχούνται από τα αντικείμενα που βρίσκονται στο τμήμα $S_{1,1}$. Για να υπολογίσουμε τα αντικείμενα του skyline συνόλου που βρίσκονται στα τμήματα $S_{1,2}$ και $S_{2,1}$ θα πρέπει να εκτελέσουμε ξανά τον αλγόριθμο για κάθε ένα από αυτά. Αντιθέτως είναι προφανές ότι τα skyline αντικείμενα του τμήματος $S_{1,1}$ θα βρίσκονται σίγουρα και στο τελικό skyline σύνολο. Αντίστοιχα στην δεύτερη εικόνα έχουμε την υλοποίηση του αλγορίθμου για τρείς διαστάσεις. Με την ίδια λογική που ακολουθήσαμε πριν τα $S_{2,3}$, $S_{2,2}$ δεν θα συνυπολογιστούν στον υπολογισμό του τελικού skyline συνόλου καθώς σίγουρα κυριαρχούνται από το τμήμα $S_{1,1}$. Για τον υπολογισμό των skyline σημείων στα τμήματα $S_{1,3}$, $S_{1,2}$ και $S_{2,1}$ θα εκτελεστεί εκ νέου ο αλγόριθμος αναδρομικά. Τέλος τα skyline σημεία του τμήματος $S_{1,1}$ θα είναι σίγουρα στο τελικό skyline σύνολο.

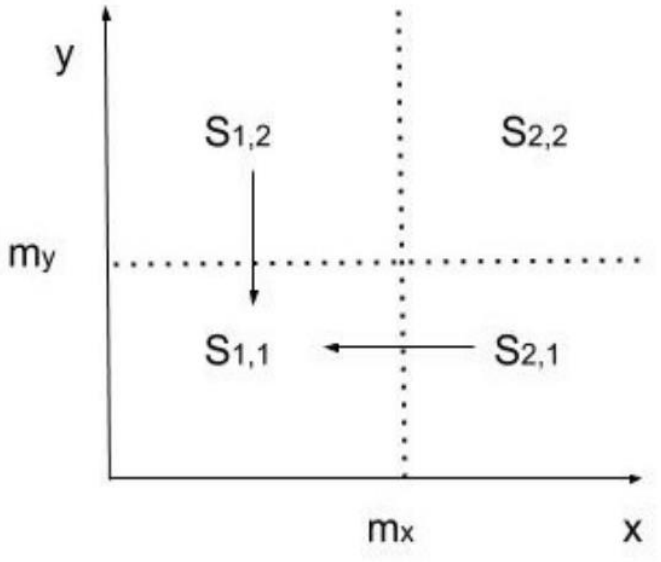


Figure 1 Divide and Conquer 2 διαστάσεις

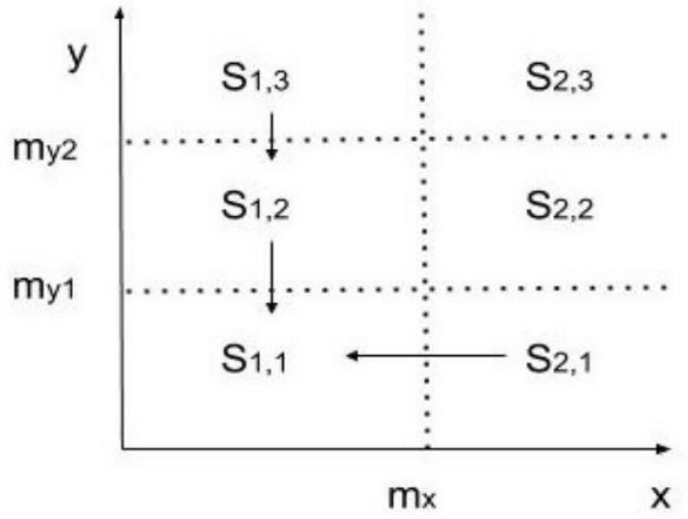


Figure 2 Divide and Conquer 3 διαστάσεις

Η πολυπλοκότητα του αλγορίθμου τόσο στη καλύτερη όσο και στη χειρότερη περίπτωση είναι $O(N(\log N)^{d-2})+O(N \log N)$, όπου d είναι ο αριθμός των διαστάσεων και N ο συνολικός αριθμός αντικειμένων. Συνεπώς σε σχέση με τον Block Nested Loop αλγόριθμο περιμένουμε ο Διαίρει και Βασίλευε να είναι καλύτερος στη χειρότερη περίπτωση και χειρότερος στη καλύτερη περίπτωση.

Bitmap

Όπως γνωρίζουμε για τον υπολογισμό του skyline συνόλου χρειάζεται να εξετάσουμε όλα τα αντικείμενα της βάσης. Στον Bitmap αλγόριθμο [1] εκμεταλλευόμαστε την ταχύτητα των δυαδικών πράξεων προκειμένου να υπολογίσουμε το skyline σύνολο κωδικοποιώντας τα αντικείμενα σε δυαδική μορφή. Έστω ένα αντικείμενο x = (x_1,…,x_d), όπου d ο αριθμός των διαστάσεων που μας απασχολούν. Η κωδικοποίηση γίνεται αναπαριστώντας κάθε σημείο ως ένα διάνυσμα μήκους m. Κάθε x_i αναπαρίσταται από k_i ψηφία. Συνεπώς ισχύει $m = \sum_{i=1}^{d} k_i$. Θεωρούμε επίσης ότι κάθε διάσταση i έχει ένα εύρος τιμών. Η j-οστή μικρότερη τιμή της διάστασης i αντιπροσωπεύεται από k_i ψηφία εκ των οποίων τα k_i-j+1 είναι άσσοι και τα υπόλοιπα μηδενικά. Παρακάτω ακολουθεί ένα παράδειγμα μιας μετατροπής.

Έστω τα αντικείμενα a(1,5), b(2,3), και c(3,4). Με τη μέθοδο που περιεγράφηκε παραπάνω τα αντικείμενα γίνονται a(111,100), b(110,111), c(100,110).

Για να υπολογίσουμε εάν ένα αντικείμενο είναι μέρος του skyline συνόλου πρέπει να υπολογίσουμε τις ακολουθίες ψηφίων V_x και V_y για αυτό και να εφαρμόσουμε σε αυτές τον λογικό τελεστή ΚΑΙ. Εάν το αποτέλεσμα αυτής της λογικής πράξης έχει μόνο έναν άσσο τότε το αντικείμενο ανήκει στο skyline σύνολο διαφορετικά δεν ανήκει. Έστω ότι η τιμή που έχει ένα αντικείμενο στη διάσταση x αντιστοιχεί στη i-οστή μικρότερη τιμή για αυτή τη διάσταση. Τότε η μεταβλητή V_x θα προκύψει εάν πάρουμε από κάθε m-bit τιμή το i-οστό ψηφίο από τα δεξιά. Αντίστοιχα υπολογίζεται και η τιμή της V_y. Στο παράδειγμά μας για το σημείο b(2,3) ισχύει ότι V_x=110 και V_y=010. V_x AND V_y= 010 το οποίο περιέχει μόνο έναν άσσο και συνεπώς το συγκεκριμένο αντικείμενο ανήκει στο skyline σύνολο. Αντίστοιχα για το αντικείμενο c(3,4) ισχύει ότι V_x=111 και V_y=011. V_x AND V_y= 011 το οποίο έχει δύο άσσους άρα το αντικείμενο c δεν ανήκει στο skyline σύνολο.

Nearest Neighbour

Στον αλγόριθμο του κοντινότερου γείτονα [6] γίνεται χρήση χωρικών δεικτών. Οι χωρικοί δείκτες είναι πολύ χρήσιμοι για τον υπολογισμό του skyline συνόλου καθώς μας δίνουν τη δυνατότητα να αποφεύγουμε περιττούς ελέγχους αντικειμένων. Ο αλγόριθμος κοντινότερου γείτονα κάνει χρήση R-δέντρων. Πιο συγκεκριμένα εκτελώντας αναδρομικά τον αλγόριθμο αναζήτησης κοντινότερου γείτονα εντοπίζει από μια αφετηρία το αντικείμενο με την κοντινότερη απόσταση και το προσθέτει στο skyline σύνολο. Η μετρική που χρησιμοποιείται για τον αλγόριθμο αναζήτησης κοντινότερου γείτονα είναι κάποια μονότονη συνάρτηση

όπως για παράδειγμα η Ευκλείδεια απόσταση. Μετά την επιλογή του skyline αντικειμένου τα υπόλοιπα δεδομένα κατατάσσονται σε περιοχές που προκύπτουν με βάση το skyline αντικείμενο. Κάποιες από αυτές γνωρίζουμε εξ αρχής ότι δεν χρειάζεται να τις ελέγξουμε για skyline αντικείμενα και απορρίπτονται καθώς όλα τα αντικείμενα μέσα σε αυτές κυριαρχούνται από κάποιο που ήδη έχει προστεθεί στο skyline σύνολο, ενώ οι υπόλοιπες προστίθενται σε μια λίστα που περιέχει όλες τις περιοχές που ακόμα δεν έχουν ελεγχθεί.

Στο παράδειγμα που ακολουθεί περιγράφεται η παραπάνω διαδικασία. Παρατηρούμε ότι έχουμε μια σειρά από αντικείμενα κάθε ένα από τα οποία αποτελείται από δύο διαστάσεις. Εφαρμόζοντας τον αλγόριθμο αναζήτησης κοντινότερου γείτονα με σημείο αφετηρίας την αρχή των αξόνων βλέπουμε ότι το αντικείμενο d είναι πλησιέστερα και άρα αυτό προστίθεται στο skyline σύνολο. Το επίπεδο πλέον χωρίζεται σε τρείς περιοχές. Γνωρίζουμε ότι η περιοχή 3 θα απορριφθεί καθώς κάθε σημείο μέσα σε αυτή κυριαρχείται από το σημείο d. Αντίστοιχα οι περιοχές ένα και δύο προστίθενται στη λίστα. Παρομοίως για την περιοχή 1 το αντικείμενο με την μικρότερη απόσταση από το σημείο αφετηρίας της είναι το b το οποίο προστίθεται στο skyline σύνολο, χωρίζοντας το επίπεδο στις περιοχές 4,5,6. Όπως και πριν απορρίπτουμε την περιοχή 5 καθώς κάθε αντικείμενο μέσα σε αυτή κυριαρχείται από το b. Οι περιοχές 4 και 6 προστίθενται στη λίστα και ο αλγόριθμος συνεχίζει με τον ίδιο τρόπο για όλες μέχρι η λίστα να μείνει κενή.

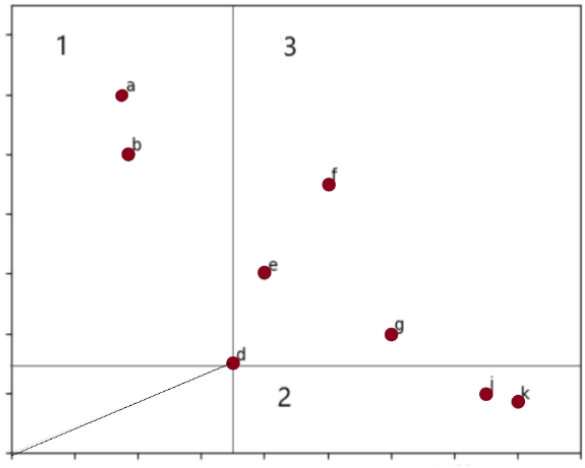


Figure 3 Nearest Neighbour Αλγόριθμος επιλογή αντικειμένου d

Ο αλγόριθμος κοντινότερου γείτονα παρουσιάζει δύο βασικά μειονεκτήματα. Αρχικά κάποια από τα skyline σημεία που εντοπίζονται σε κάποιο σημείο του αλγορίθμου έχουν ήδη εντοπιστεί πιο νωρίς στον αλγόριθμο. Κάθε αναδρομική κλήση του αλγορίθμου τερματίζει

αφού διασχίσει ένα ολόκληρο το μονοπάτι του R-δέντρου με αποτέλεσμα το I/O κόστος του αλγορίθμου να είναι σημαντικά μεγάλο. Το δεύτερο μειονέκτημα του συγκεκριμένου αλγορίθμου είναι ότι η λίστα στην οποία προστίθενται οι περιοχές που πρέπει να μελετηθούν μπορεί να γίνει σημαντικά μεγαλύτερη από το ίδιο το πλήθος των αντικειμένων με αρνητικές επιπτώσεις στη πολυπλοκότητα του.

Branch and Bound

Όπως και ο αλγόριθμος κοντινότερου γείτονα έτσι και ο Branch and Bound [7] αλγόριθμος βασίζεται στον αλγόριθμο αναζήτησης κοντινότερου γείτονα και κάνει χρήση R-δέντρων. Πρόκειται για έναν προοδευτικό (progressive) αλγόριθμο ο οποίος έχει πολύ χαμηλότερο I/O κόστος από τον αλγόριθμο κοντινότερου γείτονα καθώς επισκέπτεται τους κόμβους του R-δέντρου που περιέχουν skyline αντικείμενα μόνο μια φορά. Η κύρια μετρική που χρησιμοποιείται είναι η L_1 απόσταση. Κάθε ενδιάμεση είσοδος του R-δέντρου αντιπροσωπεύει ένα ελάχιστα οριοθετημένο ορθογώνιο (Minimum Bounding Rectangle) ενός κόμβου, ενώ τα φύλλα του R-δέντρου αντιπροσωπεύουν ένα αντικείμενο. Για την εκτέλεση του αλγορίθμου χρησιμοποιείται μια δομή σωρού Η στην οποία αποθηκεύονται κόμβοι και αντικείμενα (φύλλα του δέντρου), μαζί με τις αντίστοιχες ελάχιστες αποστάσεις τους από την εκάστοτε αφετηρία. Επιπλέον χρησιμοποιείται ένα set S στο οποίο αποθηκεύονται τα skyline αντικείμενα. Η ελάχιστη απόσταση ενός κόμβου-φύλλου υπολογίζεται αθροίζοντας τις συντεταγμένες του, ενώ η ελάχιστη απόσταση ενός ενδιάμεσου κόμβου (δηλαδή ενός ελάχιστα οριοθετημένου ορθογωνίου) αντιστοιχεί στο άθροισμα των συντεταγμένων της κάτω αριστερά γωνίας του.

Αρχικά ο σωρός Η περιέχει τα παιδιά της ρίζας του R-δέντρου και το S είναι κενό. Τα στοιχεία στον σωρό είναι διατεταγμένα σε αύξουσα σειρά με βάση την μετρική L_1. Όσο ο σωρός δεν είναι κενός αφαιρούμε το πρώτο στοιχείο του t. Εξετάζουμε τη περίπτωση το t να είναι κόμβος-φύλλο ή ενδιάμεσος κόμβος. Στη πρώτη περίπτωση εάν το αντικείμενο δεν κυριαρχείται από κάποιο στοιχείο του S, πρόκειται για αντικείμενο που ανήκει στο skyline σύνολο και έτσι το προσθέτουμε στο S, διαφορετικά το αντικείμενο απορρίπτεται. Εναλλακτικά μιλάμε για ελάχιστα οριοθετημένο ορθογώνιο όπου χρειάζεται να ελέγξουμε όλα τα παιδιά του t_i. Εάν κάποιο παιδί δεν κυριαρχείται από κάποιο αντικείμενο στο S το εισάγουμε στον σωρό Η διαφορετικά το απορρίπτουμε. Ο αλγόριθμος τερματίζει μόλις ο σωρός αδειάσει και το S είναι το skyline σύνολο.

Skyline σε Υπο-χώρους

Με την αύξηση των διαστάσεων που έχει κάθε αντικείμενο γίνεται όλο και πιο δύσκολο ένα αντικείμενο να κυριαρχείται από κάποιο άλλο. Αυτό έχει ως αποτέλεσμα όταν μελετάμε το skyline σύνολο ολόκληρου του χώρου να μας επιστρέφεται ένα πολύ μεγάλο σύνολο από skyline αντικείμενα κάνοντας τη λήψη αποφάσεων αρκετά δύσκολη. Επιπλέον μερικές φορές δεν μας ενδιαφέρει μόνο το σύνολο των skyline αντικειμένων σε ολόκληρο τον χώρο αλλά θέλουμε να γνωρίζουμε ποια είναι τα skyline αντικείμενα σε ένα συγκεκριμένο υπο-χώρο (subspace) του συνόλου είτε γιατί κάποιος χρήστης μπορεί να ενδιαφέρεται μόνο για αυτόν είτε για να μπορέσουμε να βρούμε ποια αντικείμενα είναι κυρίαρχα στους περισσότερους υπο-χώρους και άρα πιο σημαντικά. Για τον συγκεκριμένο σκοπό υπάρχουν μια σειρά από προτάσεις με χαρακτηριστική την έννοια του SKYCUBE [8] σύμφωνα με την οποία υπολογίζουμε τα skyline σύνολα όλων των πιθανών υπο-χώρων που δεν είναι άδειοι για ένα συγκεκριμένο σύνολο διαστάσεων. Οι σημαντικότερες προσεγγίσεις για τον υπολογισμό του skycube είναι η top-down και η bottom-up. Στην πρώτη υπολογίζεται με χρήση αναδρομής το skyline σύνολο του κόμβου-πατέρα μέσα από την απαρίθμηση των skyline συνόλων για τους υπο-χώρους που αντιστοιχούν στους κόμβους παιδιά. Στην bottom-up προσέγγιση το skyline σύνολο ενός κόμβου προκύπτει από τη συγχώνευση των skyline συνόλων των παιδιών του κόμβου.

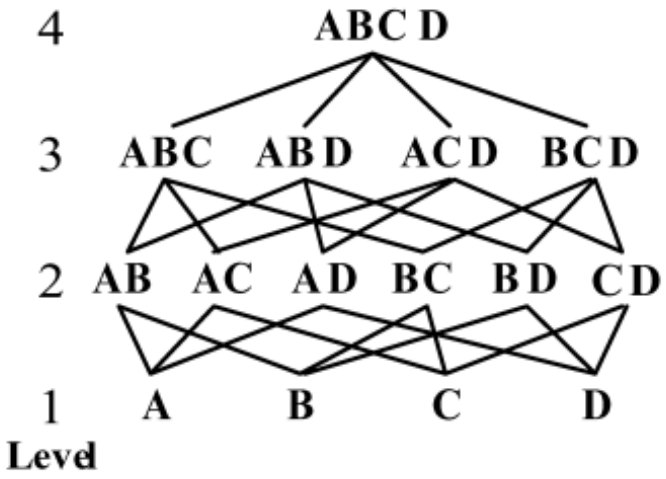


Figure 4 Skycube δομή

Skyline σύνολο και περιορισμοί

Ο μεγάλος αριθμός αντικειμένων στο skyline σύνολο εξαιτίας της αύξησης των διαστάσεων των αντικειμένων έχει οδηγήσει στην εφαρμογή περιορισμών [9] προκειμένου το skyline σύνολο να περιλαμβάνει πιο χρήσιμη και διαχειρίσιμη πληροφορία. Υπάρχουν δυο κατηγορίες περιορισμών. Η πρώτη, ονομάζεται skyline ερωτήματα με περιορισμούς (skyline queries with constraints) και αφορά ολόκληρο το σύνολο των δεδομένων. Αρχικά υπολογίζεται το skyline σύνολο και στη συνέχεια εφαρμόζονται περιορισμοί πάνω σε αυτό σε κάποιες από τις διαστάσεις των αντικειμένων. Η δεύτερη κατηγορία ονομάζεται περιορισμένα skyline ερωτήματα (constrained skyline queries) και σε αντίθεση με την πρώτη πρώτα εφαρμόζουμε τους περιορισμούς στις διαστάσεις των αντικειμένων και στη συνέχεια υπολογίζουμε το skyline σύνολο. Τα αποτελέσματα των δύο κατηγοριών ενδέχεται να διαφέρουν μεταξύ τους καθώς τα αντικείμενα που προκύπτουν στη δεύτερη κατηγορία δεν είναι απαραίτητα αντικείμενα του skyline συνόλου για όλα τα δεδομένα.

Skyline σε δυναμικό περιβάλλον

Μέχρι τώρα είδαμε υπολογισμό skyline συνόλου στις αρχικές διαστάσεις των δεδομένων. Υπάρχουν περιπτώσεις όμως στις οποίες θέλουμε να προτείνουμε στον χρήστη μια σειρά από αντικείμενα με βάση ένα ιδανικό αντικείμενο για αυτόν. Για να πραγματοποιηθεί αυτό έχουν προταθεί τα δυναμικά skyline σύνολα [10]. Έστω ένα αντικείμενο q. Το δυναμικό skyline σύνολο αποτελείται από όλα τα στοιχεία που δεν κυριαρχούνται δυναμικά σε σχέση με το q. Προκειμένου να υπολογίσουμε το δυναμικό skyline σύνολο χρησιμοποιούμε μια σειρά από συναρτήσεις αποστάσεων (distance functions) οι οποίες έχουν ως είσοδο τις αρχικές συντεταγμένες ενός αντικειμένου και το αντικείμενο αναφοράς q και επιστρέφουν τις δυναμικές συντεταγμένες του αντικειμένου. Έτσι λέμε ότι ένα αντικείμενο κυριαρχεί δυναμικά ένα άλλο εάν είναι καλύτερο σε τουλάχιστον μια από τις δυναμικές διαστάσεις του ως προς το άλλο και καλύτερο ή ίσο σε όλα τις υπόλοιπες.

2.2 Συστήματα Προτιμήσεων

Καθώς ο όγκος της πληροφορίας αυξανόταν με ραγδαίους ρυθμούς η εύρεση του επιθυμητού αποτελέσματος γινόταν όλο και πιο δύσκολη διαδικασία. Τα αποτελέσματα που επιστρέφονταν στον χρήστη μετά από κάποια αναζήτηση δεν λάμβαναν υπ’ όψη τους τις προσωπικές προτιμήσεις του και καθώς ο όγκος των δεδομένων δεν καθιστούσε δυνατό τον έλεγχο κάθε αποτελέσματος ο χρήστης μπορεί να μην εύρισκε ποτέ την ιδανική επιλογή για

αυτόν. Το παραπάνω πρόβλημα οδήγησε στην ανάπτυξη μεθόδων που συμπεριλαμβάνουν στην αναζήτηση και επιστροφή δεδομένων τις εξατομικευμένες προτιμήσεις κάθε χρήστη.

2.2.1 Ορισμός

Τα συστήματα προτιμήσεων αποτελούν συστήματα συλλογής πληροφοριών σχετικά με τις προτιμήσεις των χρηστών τους σε ένα σύνολο από αντικείμενα. Στόχος τους είναι μέσα από την επεξεργασία αυτών των πληροφοριών να δημιουργεί ουσιαστικές προτάσεις σε μια ομάδα χρηστών για αντικείμενα ή προϊόντα που μπορεί να τους ενδιαφέρουν.

Τα συστήματα προτιμήσεων χωρίζονται σε τρείς βασικές κατηγορίες. Η πρώτη ονομάζεται Συνεργατικό Φιλτράρισμα (Collaborative Filtering) και βασίζεται στην ιδέα πως μια ομάδα χρηστών με παρόμοια ενδιαφέροντα θα προτιμάει και τα ίδια αντικείμενα. Η δεύτερη κατηγορία είναι το φιλτράρισμα βάση περιεχομένου (Content-Based Filtering) και προτείνει αντικείμενα σε ένα χρήστη με βάση αντικείμενα που είχε προτιμήσει στο παρελθόν. Η τρίτη κατηγορία είναι τα υβριδικά συστήματα προτιμήσεων (Hybrid Filtering) που αποτελεί συνδυασμό μεθόδων. Παρακάτω ακολουθεί η ανάλυση των τριών αυτών κατηγοριών.

2.2.2 Συνεργατικό Φιλτράρισμα

Το Συνεργατικό Φιλτράρισμα λειτουργεί συλλέγοντας πληροφορίες από τους χρήστες με τη μορφή κριτικών πάνω στα δεδομένα. Έπειτα ανακαλύπτει ομοιότητες στον τρόπο που βαθμολογούν οι χρήστες προκειμένου να τους προτείνει δεδομένα της αρεσκείας τους. Η συγκεκριμένη κατηγορία συστημάτων προτιμήσεων χωρίζεται σε δυο υποκατηγορίες το συνεργατικό φιλτράρισμα με βάση τους γείτονες (neighbourhood-based) και το συνεργατικό φιλτράρισμα με βάση το μοντέλο (model-based).

2.2.2.1 Συνεργατικό Φιλτράρισμα με βάση τους γείτονες

Υπάρχουν δυο βασικές κατηγορίες συνεργατικού φιλτραρίσματος με βάση τους γείτονες. Το πρώτο, βασίζεται στον χρήστη (user-based) και σε όμοιους χρήστες με αυτόν ενώ το δεύτερο, προτείνει αντικείμενα σε ένα χρήστη με βάση γειτονιές αντικειμένων (item-based)

User-based

Σε αυτή την υποκατηγορία, επιλέγεται με βάση την ομοιότητα με τον χρήστη-στόχο (target user) ένα υποσύνολο χρηστών και με έναν σταθμισμένο συνδυασμό των κριτικών του υποσυνόλου προκύπτουν οι προβλέψεις για τον χρήστη-στόχο. Πιο συγκεκριμένα ορίζουμε βάρη για τους χρήστες του υποσυνόλου με βάση την ομοιότητα τους ως προς τον χρήστη-στόχο, τους διατάσσουμε σε φθίνουσα σειρά και επιλέγουμε τους πρώτους k καθώς αυτοί είναι οι k πλησιέστεροι χρήστες. Η προβλεπόμενη βαθμολογία του χρήστη-στόχου προκύπτει από τις κριτικές των k επιλεγμένων χρηστών.

Ο πιο κλασικός τρόπος να υπολογιστεί η ομοιότητα ανάμεσα σε δύο χρήστες είναι η Pearson συσχέτιση [11](Pearson correlation). Έστω $w_{t,u}$ η ομοιότητα ανάμεσα στους δυο χρήστες η οποία ισούται με:

$$w_{t,u} = \frac{\sum_{i \in I}(r_{t,i} - \overline{r_t})(r_{u,i} - \overline{r_u})}{\sqrt{\sum_{i \in I}(r_{t,i} - \overline{r_t})^2 (r_{u,i} - \overline{r_u})^2}}$$

Όπου I είναι το σύνολο των αντικειμένων που έχουν ψηφίσει και οι δυο χρήστες, $r_{t,i}$ και $r_{u,i}$ οι κριτικές του χρήστη-στόχου και του άλλου χρήστη για το αντικείμενο i αντίστοιχα και $\overline{r_t}$, $\overline{r_u}$ η μέση τιμή των κριτικών του χρήστη-στόχου και του άλλου χρήστη αντίστοιχα .
Επιλέγω τις k μεγαλύτερες τιμές που θα προκύψουν (έστω ότι ονομάζουμε το σύνολο K), και με βάση αυτά τα βάρη υπολογίζω τη πρόβλεψη για τη βαθμολογία του χρήστη-στόχου. Οι προβλέψεις προκύπτουν από τον σταθμισμένο μέσο όρο των αποκλίσεων από τον μέσο όρο του χρήστη-γείτονα όπως δίνεται από τον τύπο:

$$p_{t,i} = \overline{r_t} + \frac{\sum_{u \in K}\left(r_{u,i} - \overline{r_u}\right) * w_{t,u}}{\sum_{u \in K} w_{t,u}}$$

Η ποσότητα $r_{u,i} - \overline{r_u}$ ονομάζεται mean centered rating και χρησιμοποιείται προκειμένου να αποφύγουμε τυχόν σφάλματα λόγω διαφορετικής συμπεριφοράς χρηστών (για παράδειγμα κάποιος χρήστης μπορεί να βάζει γενικά υψηλότερες βαθμολογίες). Εναλλακτικά μπορούμε να χρησιμοποιήσουμε το z-score στο οποίο η παραπάνω διαφορά διαιρείται με τη διασπορά των βαθμολογιών του χρήστη u.

Στάθμιση

Η αξιοπιστία της συνάρτησης ομοιότητας πολλές φορές ενδέχεται να επηρεαστεί από τον αριθμό των αντικειμένων που έχουν αξιολογήσει και οι δύο χρήστες. Ένα ζευγάρι χρηστών που έχει μικρό αριθμό κοινών κριτικών πρέπει να έχει μικρότερη βαρύτητα σε σχέση με ένα ζευγάρι χρηστών για το οποίο το σύνολο των κοινών αντικειμένων που έχουν βαθμολογήσει είναι μεγάλο. Για αυτό το λόγο γίνεται χρήστη ενός συντελεστή έκπτωσης (discount factor) ο οποίος προκύπτει από τον τύπο:

$$\frac{\min\{I_t \cap I_u, \beta\}}{\beta}$$

Όπου $I_t \cap I_u$ το σύνολο των κοινών κριτικών των δύο χρηστών και β η τιμή που λειτουργεί ως κατώφλι. Η τιμή του συντελεστή έκπτωσης κυμαίνεται πάντα στο [0,1] και η τελική τιμή ομοιότητας για δύο χρήστες προκύπτει πολλαπλασιάζοντας την αρχική τιμή της ομοιότητας με αυτόν τον παράγοντα. Η μέθοδος αυτή είναι γνωστή ως στάθμιση σημαντικότητας (significance weighting) [12].

Αντίστροφη Συχνότητα Χρήστη

Κατά τον υπολογισμό του συνόλου όμοιων χρηστών καθώς και κατά τον υπολογισμό πρόβλεψης της βαθμολογίας ενός χρήστη, αντικείμενα που έχουν βαθμολογηθεί από πολλούς χρήστες δεν μας είναι ιδιαίτερα χρήσιμα. Αυτό συμβαίνει καθώς συνήθως τέτοια αντικείμενα είναι η πολύ αγαπητά ή καθόλου από όλους τους χρήστες και συνεπώς δεν μας δίνουν κάποια ενδιαφέρουσα πληροφορία σχετικά με τις εξατομικευμένες προτιμήσεις ενός χρήστη. Προκειμένου να αντιμετωπιστεί αυτό το φαινόμενο προτάθηκε η έννοια της αντίστροφης συχνότητας χρήστη (inverse user frequency) [13].

Έστω m_i το πλήθος των ατόμων που έχουν αξιολογήσει το αντικείμενο i και m ο συνολικός αριθμός χρηστών. Τότε το βάρος w_i του αντικειμένου i υπολογίζεται από τον τύπο:

$$w_i = \log(m/m_i)$$

Κάθε αντικείμενο i πολλαπλασιάζεται με τον αντίστοιχο παράγοντα w_i και στη φάση του υπολογισμού της ομοιότητας των χρηστών αλλά και στον υπολογισμό της πρόβλεψης βαθμολογιών.

Item-Based

Για μεγάλο πλήθος χρηστών η μεθοδολογία που περιεγράφηκε παραπάνω δεν συμφέρει καθώς αυξάνεται κατά πολύ η υπολογιστική πολυπλοκότητα της αναζήτησης χρηστών με μεγάλη ομοιότητα στον χρήστη-στόχο. Για την αντιμετώπιση του παραπάνω προτάθηκε το συνεργατικό φιλτράρισμα που βασίζεται στο αντικείμενο [14]. Σε αυτή τη περίπτωση αντί να μας ενδιαφέρει η εύρεση παρόμοιων χρηστών ψάχνουμε παρόμοια αντικείμενα με τα αντικείμενα που έχει βαθμολογήσει ο χρήστης που μας ενδιαφέρει.

Όπως και στην user-based κατηγορία χρησιμοποιούμε Pearson συσχέτιση για να υπολογίσουμε την ομοιότητα των αντικειμένων η οποία για δύο αντικείμενα i και j προκύπτει σύμφωνα με τον τύπο:

$$w_{i,j} = \frac{\sum_{u \,\epsilon\, U}(r_{u,i} - \overline{r_i})(r_{u,j} - \overline{r_j})}{\sqrt{\sum_{u \,\epsilon\, U}(r_{u,i} - \overline{r_i})^2 (r_{u,i} - \overline{r_j})^2}}$$

Όπου U είναι το σύνολο των χρηστών που έχουν ψηφίσει και τα δύο αντικείμενα, $r_{u,i}$ και $r_{u,j}$ οι βαθμολογίες του χρήστη για τα δύο αντικείμενα και $\overline{r_i}, \overline{r_j}$ η μέση τιμή των βαθμολογιών των αντικειμένων i και j αντίστοιχα. Επιλέγουμε τις k μεγαλύτερες τιμές οι οποίες αποτελούν το σύνολο των αντικειμένων που έχουν βαθμολογηθεί από τον χρήστη που μας ενδιαφέρει και είναι τα πλησιέστερα στο αντικείμενο i (σύνολο K) και υπολογίζουμε τη προβλεπόμενη βαθμολογία του χρήστη μας (έστω t) για το αντικείμενο i με τον τύπο:

$$p_{t,i} = \frac{\sum_{j \,\epsilon\, K}(r_{t,j}) w_{i,j}}{\sum_{j \,\epsilon\, K} w_{i,j}}$$

Ουσιαστικά στη user-based υποκατηγορία χρησιμοποιούμε τις βαθμολογίες από τους γειτονικούς χρήστες για να προβλέψουμε τη βαθμολογία που επιθυμούμε, ενώ στην item-based υποκατηγορία χρησιμοποιούμε τις βαθμολογίες που έχει δώσει ο χρήστης-στόχος σε μια γειτονιά από αντικείμενα.

Ομαδοποίηση

Όπως αναφέρεται και στο [15] η επίδοση του συνεργατικό φιλτράρισμα με βάση τους γείτονες καθώς αυξάνεται ο αριθμός χρηστών και αντικειμένων μειώνεται σημαντικά. Αρκεί να σκεφτούμε πως στις user-based τεχνικές για m χρήστες και n ο μέγιστος αριθμός βαθμολογιών που έχει δώσει ένας χρήστης, η πολυπλοκότητα για τον υπολογισμό του

συνόλου των χρηστών με τη μεγαλύτερη ομοιότητα είναι O(nm) και συνεπώς για τον υπολογισμό του συνόλου των γειτονικών χρηστών για κάθε χρήστη χρειάζομαι $O(m^2n)$. Αντίστοιχα στις item-based προσεγγίσεις η χρονική πολυπλοκότητα είναι της τάξης του $O(n^2m)$.

Προκειμένου να αντιμετωπιστεί το παραπάνω πρόβλημα έχει προταθεί η δημιουργία ομάδων χρηστών (clusters). Στη user-based προσέγγιση η διαδικασία που περιεγράφηκε σε αυτή την ενότητα παραμένει και εδώ ίδια με τη διαφορά ότι πλέον ο υπολογισμός του συνόλου των όμοιων χρηστών με τον χρήστη-στόχο μας προκύπτει μέσα από την ομάδα και όχι από ολόκληρο το σύνολο των χρηστών. Εάν δηλαδή αναπαραστήσουμε τα δεδομένα μας ως ένα πίνακα mxn (όπου m οι χρήστες και n τα αντικείμενα) κάνουμε λόγο για ομαδοποίηση των γραμμών. Αντίστοιχα στις item-based τεχνικές ομαδοποιούμε τις στήλες. Αυτό έχει ως αποτέλεσμα όπως είναι λογικό να έχουμε μικρότερη ακρίβεια στις προβλέψεις μας αλλά μας παρέχει ταυτόχρονα πολύ καλύτερη χρονική πολυπλοκότητα.

Μείωση Διαστάσεων

Τις περισσότερες φορές ο πίνακας που περιεγράφηκε παραπάνω έχει πολλά κενά κελιά. Αυτό είναι αποτέλεσμα των αραιών κριτικών των χρηστών (σε μια γραμμή του πίνακα όλα τα αντικείμενα που δεν έχουν βαθμολογηθεί από τον αντίστοιχο χρήστη της γραμμής αντιστοιχούν σε κενό). Μειώνοντας τις διαστάσεις του πίνακα μπορούμε πιο εύκολα να υπολογίσουμε αποστάσεις ανάμεσα σε χρήστες που έχουν πολύν μικρό σύνολο κοινά βαθμολογημένων αντικειμένων. Η μείωση των διαστάσεων μπορεί να επιτευχθεί μέσα από δυο κύριες μεθόδους την Αυτό μπορούμε να το πραγματοποιήσουμε με 2 τρόπους την ανάλυση πίνακα σε ιδιάζουσες τιμές (singular value decomposition) και την ανάλυση κυρίων συνιστωσών (Principal component analysis).

Στην πρώτη περίπτωση έστω ότι έχουμε ένα πίνακα R διαστάσεων mxn. Στα κενά κελιά του συμπληρώνουμε τη μέση τιμή κατά γραμμή (ή κατά στήλη) και δημιουργούμε τον πίνακα R_f. Ο πίνακας ομοιότητας S υπολογίζεται ως το γινόμενο $R_f^T * R_f$. Θέλουμε ο καινούργιος πίνακας να περιέχει τις d μεγαλύτερες τιμές των ιδιοδιανυσμάτων. Για να γίνει αυτό παίρνουμε τα κυρίαρχα διανύσματα βάσης του πίνακα R_f κάνοντας διαγωνοποίηση του $S = P\Delta P^T$, όπου P είναι ένας nxn πίνακας οι στήλες του οποίου περιέχουν τα ορθοκανονικά ιδιοδιανύσματα του S και Δ έναν διαγώνιος πίνακας που περιέχει τις ιδιοτιμές του S.

Στην ανάλυση κυρίων συνιστωσών μας ενδιαφέρει ο υπολογισμός των κύριων συνιστωσών (principal components) τις οποίες χρησιμοποιούμε στη συνέχεια προκειμένου να πραγματοποιήσουμε αλλαγή βάσης στον αρχικό μας πίνακα. Πρόκειται για την ακολουθία των μοναδιαίων διανυσμάτων στα οποία το i-οστο διάνυσμα είναι η κατεύθυνση μιας γραμμής που ταιριάζει καλύτερα στα δεδομένα ενώ είναι ορθοκανονικά στα πρώτα i-1 διανύσματα.

Συνεργατικό Φιλτράρισμα με βάση τους γείτονες και Γράφοι

Ένας εναλλακτικός τρόπος αναπαράστασης των δεδομένων μας είναι οι γράφοι. Μέσα από αυτή την αναπαράσταση μας δίνεται η ευκαιρία να εξάγουμε πληροφορίες και να προτείνουμε αντικείμενα σε έναν χρήστη κάνοντας χρήση των αλγορίθμων του τομέα των δικτύων. Γράφοι μπορούν να κατασκευαστούν τόσο ως προς τους χρήστες (user-user) [16] όσο και ως προς τα αντικείμενα(item-item) ενώ υπάρχει και η κατηγορία γράφων που βασίζεται και στα δύο(user-item) [17].

Γράφοι user-item

Στους γράφους της κατηγορίας χρήστης-αντικείμενο κάνουμε λόγο για έναν διμερή γράφο ο οποίος έχεις ως κόμβους όλους τους χρήστες και τα αντικείμενα. Σε αντίθεση με τις μεθόδους που περιγράψαμε έως τώρα για να θεωρηθούν δυο χρήστες γείτονες δεν είναι απαραίτητο να έχουν πολλά κοινά αντικείμενα που έχουν βαθμολογήσει και οι δύο. Κάθε ακμή του γράφου είναι μη κατευθυνόμενη (undirected). Ακμές υπάρχουν μόνο ανάμεσα σε αντικείμενα χρήστες και προκύπτουν μόνο όταν κάποιος χρήστης έχει βαθμολογήσει ένα αντικείμενο (ο συνολικός αριθμός των ακμών είναι ίσος με τα μη κενά στοιχεία του πίνακα). Σύμφωνα με το [15], υπάρχουν δυο κυρίαρχες μέθοδοι για τον υπολογισμό των χρηστών-γειτόνων και οι δύο βασίζονται στον υπολογισμό τον συντομότερων μονοπατιών.

Η πρώτη μέθοδος χρησιμοποιεί τυχαία μονοπάτια (random walks). Κάνοντας χρήση του PageRank αλγορίθμου ή της SimRank μεθόδου, η γειτονιά ενός χρήστη προκύπτει από το σύνολο των χρηστών που συναντιούνται πιο συχνά σε ένα τυχαίο μονοπάτι το οποίο ξεκινάει από τον χρήστη-στόχο μας. Η μέθοδος αυτή μπορεί να χρησιμοποιηθεί αντίστοιχα και για τον υπολογισμό της γειτονιάς ενός αντικειμένου.

Η δεύτερη μέθοδος βασίζεται στη μετρική Katz η οποία υπολογίζεται από τον τύπο:

$$\sum_{t=1}^{\infty} \beta^t * n_{ij}^{(t)}$$

Οπου $n_{i,j}$ ο αριθμός των μονοπατιών μήκους t ανάμεσα στους κόμβους i και j και β μια σταθερά μικρότερη του 1, η οποία δρα ως συντελεστής έκπτωσης (discount factor) για τα μονοπάτια μεγαλύτερου μήκους. Δημιουργούμε τον πίνακα mxm στα κελιά του οποίου υπάρχει η τιμή της μετρικής Katz για το εκάστοτε ζεύγος χρήστη. Η γειτονιά του χρήστη που μας ενδιαφέρει προκύπτει από το σύνολο των ζευγαριών με τις μεγαλύτερες τιμές.

Γράφοι item-item

Κάνουμε λόγο για γράφους που έχουν ως κόμβους μόνο αντικείμενα και είναι κατευθυνόμενοι, με βάρη στις ακμές τους. Εάν υπάρχει χρήστης που έχει βαθμολογήσει δύο αντικείμενα, έστω i και j, δημιουργούνται δύο ακμές η μία με κατεύθυνση από τον i στον j και η δεύτερη με αντίθετη κατεύθυνση.

Έστω w_{ij} το βάρος που αντιστοιχεί στην ακμή με κατεύθυνση από το αντικείμενο i στο j. Για να το υπολογίσουμε βρίσκουμε τον αριθμό των χρηστών που έχουν ψηφίσει και τα δύο αντικείμενα και στη συνέχεια τον διαιρούμε με το σύνολο των ακμών που έχουν αφετηρία τον κόμβο i. Όπως είναι λογικό σε γενική περίπτωση ισχύει ότι $w_{ij} \neq w_{ji}$.Εφαρμόζουμε και σε αυτή τη κατηγορία γράφων μεθόδους τυχαίων μονοπατιών για να καθορίσουμε τη γειτονιά ενός αντικειμένου.

Γράφοι user-user

Οι κόμβοι των γράφων αυτής της κατηγορίας αντιστοιχούν στους χρήστες του πίνακα. Χρησιμοποιούμε τις έννοιες του horting και της προβλεψιμότητας (predictability) τις οποίες είναι σημαντικό να εξηγήσουμε για τη κατανόηση της δημιουργίας των ακμών. Η έννοια του horting περιγράφει μια ασύμμετρη σχέση ανάμεσα σε δύο χρήστες που καθορίζεται με βάση τα κοινά αντικείμενα που έχουν βαθμολογήσει. Για να ικανοποιείται η έννοια του horting από ένα χρήστη u προς ένα χρήστη v σε επίπεδο F,G πρέπει να ισχύουν τα παρακάτω:

$$I_u \cap I_v \geq F \ , \frac{I_u \cap I_v}{I_u} \geq G$$

Όπου I_u και I_v τα αντικείμενα που έχουν ψηφίσει οι χρήστες u και v αντίστοιχα. Η έννοια της προβλεψιμότητας κάνει χρήση του horting που περιεγράφηκε παραπάνω. Ένας χρήστης v λέμε ότι προβλέπει έναν άλλο u όταν ο u horts τον v και υπάρχει μια συνάρτηση γραμμικού μετασχηματισμού f() για την οποία ισχύει:

$$\frac{\sum_{k \in I_u \cap I_v} |r_{uk} - f(r_{vk})|}{I_u \cap I_v} \leq U$$

Έχουμε ακμή από τον κόμβο του χρήστη u στον κόμβο του χρήστη v εάν ο u προβλέπει τον v. Κάθε ακμή λοιπόν αντιστοιχεί σε έναν γραμμικό μετασχηματισμό ο οποίος καθορίζει μια πρόβλεψη. Η βαθμολογία λοιπόν στην αρχή της ακμής καθορίζει τη βαθμολογία στο τέλος της. Η βαθμολογία του χρήστη u για ένα αντικείμενο k υπολογίζεται καθορίζοντας όλα τα συντομότερα μονοπάτια από τον u προς όσους έχουν ψηφίσει το k. Η προβλεπόμενη βαθμολογία του χρήστη v σύμφωνα με τον χρήστη u είναι η εφαρμογή όλων των γραμμικών απεικονίσεων από τον κόμβο του u στον κόμβο του v.

2.2.2.2 Συνεργατικό φιλτράρισμα με βάση το μοντέλο

Το συνεργατικό φιλτράρισμα μπορεί να θεωρηθεί μια διαδικασία ταξινόμησης [18]. Με βάση ένα σύνολο από βαθμολογίες αντικειμένων από χρήστες προσπαθούμε να δημιουργήσουμε ένα μοντέλο για κάθε χρήστη έτσι ώστε να κατηγοριοποιήσουμε αντικείμενα που δεν έχει αξιολογήσει σε ομάδες (για παράδειγμα του αρέσει/ δεν του αρέσει). Εναλλακτικά στην περίπτωση πρόβλεψης μιας βαθμολογίας κάνουμε λόγο για πρόβλημα παλινδρόμησης (regression). Σε αυτή τη κατηγορία εξετάζουμε πώς μπορούν οι αλγόριθμοι ταξινόμησης και κατηγοριοποίησης δεδομένων (classification algorithms) να συμβάλλουν στα συστήματα προτιμήσεων. Παρακάτω περιγράφονται μια σειρά από μεθόδους.

Δέντρα αποφάσεων

Τα δέντρα αποφάσεων έχουν τα πλεονεκτήματα της υψηλής ακρίβειας καθώς και ταχύτητας ταξινόμησης και για αυτό έχουν προταθεί στο πλαίσιο του συνεργατικού φιλτραρίσματος [19]. Γενικότερα στα δέντρα αποφάσεων έχουμε ένα κριτήριο διαχωρισμού (split criteria) σύμφωνα με το οποίο πραγματοποιείται ο διαχωρισμός των δεδομένων στα κλαδιά του δέντρου. Όσο πιο ορθό το κριτήριο διαχωρισμού τόσο καλύτερα γίνεται και η κατηγοριοποίηση των δεδομένων. Η ποιότητα του διαχωρισμού μπορεί να υπολογιστεί

χρησιμοποιώντας τον σταθμισμένο μέσο δείκτη Gini των κόμβων-παιδιών που δημιουργούνται από ένα διαχωρισμό. Αν p_1, ...,p_r είναι τα κλάσματα των δεδομένων που ανήκουν σε r διαφορετικές κατηγορίες για ένα κόμβο S ο δείκτης Gini για τον συγκεκριμένο κόμβο υπολογίζεται από τον τύπο:

$$G(S) = 1 - \sum_{i=1}^{r} p_i^2$$

Ο δείκτης αυτός κυμαίνεται στο [0,1] και όσο μικρότερη είναι η τιμή του τόσο μεγαλύτερη είναι η διαχωριστική δύναμη του. Συνήθως χρησιμοποιούμε τον δείκτη αυτό για να καταλήξουμε σε κάθε στάδιο του δέντρο στο καλύτερο από τα χαρακτηριστικά των δεδομένων μας που πρέπει να χρησιμοποιηθεί ως κριτήριο διαχωρισμού.

Στην περίπτωση των συστημάτων προτιμήσεων η διαφοροποίηση έγκειται στο γεγονός ότι δεν υπάρχει ξεκάθαρος διαχωρισμός ανάμεσα στις προβλεπόμενες βαθμολογίες και στις βαθμολογίες που έχουν υποβληθεί από τους χρήστες. Έστω ότι χρησιμοποιούμε τη συνολική βαθμολογία ενός διαμερίσματος ως χαρακτηριστικό διαχωρισμού. Οι χρήστες που βαθμολόγησαν το διαμέρισμα χαμηλότερα από κατώφλι κατηγοριοποιούνται στη μία πλευρά του δέντρου και όσοι έδωσαν βαθμολογία μεγαλύτερη από το κατώφλι κατηγοριοποιούνται στην άλλη. Ερχόμαστε όμως αντιμέτωποι και με τους χρήστες που δεν έχουν βαθμολογήσει το συγκεκριμένο διαμέρισμα, οι οποίοι δεν γνωρίζουμε σε ποια κατηγορία ανήκουν. Προκειμένου να αντιμετωπιστεί αυτό το πρόβλημα χρησιμοποιούμε τεχνικές μετατροπής του αρχικού πίνακα σε πίνακα μειωμένων διαστάσεων. Μέσα από αυτή την αναπαράσταση μπορούμε να δημιουργήσουμε για κάθε αντικείμενο ένα δέντρο αποφάσεων. Έτσι συνολικά για n αντικείμενα καταλήγουμε με n δέντρα αποφάσεων. Για να υπολογίσουμε τη προβλεπόμενη βαθμολογία του i-οστού χρήστη για το j-οστο αντικείμενο επιλέγουμε την γραμμή i του πίνακα μειωμένων διαστάσεων ως παράδειγμα δοκιμής (test instance) και το j-οστό δέντρο αποφάσεων ως μοντέλο.

Κανόνες Συσχέτισης

Στο συνεργατικό φιλτράρισμα έχουν χρησιμοποιηθεί και οι κανόνες συσχέτισης για την δημιουργία προτάσεων στους χρήστες [20]. Μιλώντας γενικά για κανόνες συσχέτισης μας ενδιαφέρει να βρούμε σύνολα από αντικείμενα που είναι στενά συσχετισμένα μέσα στη βάση

δεδομένων μας. Για να πραγματοποιηθεί το παραπάνω εκμεταλλευόμαστε το γεγονός ότι όταν οι χρήστες επιλέγουν μια σειρά από αντικείμενα της αρεσκείας τους παρατηρείται μερικές φορές η τάση να επιλέγονται κάποια αντικείμενα σε ζεύγη. Κάνουμε χρήση των εννοιών της υποστήριξης (support) και της εμπιστοσύνης (confidence) οι οποίες αναλύονται παρακάτω.

Μας ενδιαφέρει να υπολογίσουμε κατά πόσο ένα σύνολο αντικειμένων εμφανίζεται συχνά ή όχι. Η υποστήριξη ενός συνόλου αντικειμένων καθορίζεται από το κλάσμα εμφάνισης του συγκεκριμένου συνόλου ως προς τα σύνολα των επιλεγμένων αντικειμένων των χρηστών. Συνεπώς όσο μεγαλύτερη είναι η υποστήριξη ενός συνόλου αντικειμένων τόσο πιο συχνά εμφανίζεται αυτό το σύνολο στις επιλογές των χρηστών.

Έστω τώρα ότι έχουμε έναν κανόνα σύμφωνα με τον οποίο εάν ένας χρήστης έχει επιλέξει τα αντικείμενα α,β (έστω σύνολο Α), θα τον ενδιαφέρει και το αντικείμενο γ (έστω σύνολο Β). Το παραπάνω μαθηματικά μεταφράζεται ως Α => Β. Η έννοια της εμπιστοσύνης αφορά το κατά πόσο αυτός ο κανόνας είναι έγκυρος. Συνεπώς η εμπιστοσύνη ενός κανόνα είναι η υπό όρους πιθανότητα στην επιλογή ενός χρήστη να υπάρχει το σύνολο αντικειμένων Β δεδομένου ότι υπάρχει το σύνολο αντικειμένων Α. Η συγκεκριμένη πιθανότητα υπολογίζεται διαιρώντας την υποστήριξη της ένωσης των συνόλων Α,Β με την υποστήριξη του συνόλου αντικειμένων Α. Όπως είναι λογικό όσο μεγαλύτερη είναι η τιμή της εμπιστοσύνης τόσο πιο έγκυρος είναι και ο κανόνας.

Σύμφωνα με όσα αναφέρθηκαν παραπάνω λέμε ότι έχουμε έναν κανόνα συσχέτισης Α => Β με ελάχιστη υποστήριξη s και εμπιστοσύνη c όταν η υποστήριξη του συνόλου AUB είναι τουλάχιστον s και η εμπιστοσύνη του κανόνα Α=>Β τουλάχιστον c. Προκειμένου να εντοπίσουμε τέτοιους κανόνες συσχέτισης πρέπει πρώτα να υπολογίσουμε όλα τα σύνολα αντικειμένων με ελάχιστη υποστήριξη s. Στη συνέχεια για κάθε σύνολο εξετάζουμε όλες τις πιθανές διασπάσεις του συνόλου σε δύο μέρη καθώς κάθε τέτοια διάσπαση είναι εν δυνάμει ένας κανόνας συσχέτισης.

Συγκεκριμένα στη περίπτωση των συστημάτων προτιμήσεων έστω ότι έχουμε ένα χρήστη u και θέλουμε να του προτείνουμε μια σειρά από αντικείμενα. Υπολογίζονται αρχικά οι κανόνες συσχέτισης που προκαλούνται από τον Α δηλαδή οι κανόνες στους οποίους το αριστερό σκέλος του κανόνα είναι υποσύνολο των αντικειμένων που αρέσουν στον Α. Στη

συνέχεια πραγματοποιείται κατάταξη των κανόνων αυτών με βάση την εμπιστοσύνη του καθένα και τα πρώτα k αντικείμενα που βρίσκονται στα σύνολα δεξιά των κανόνων είναι και τα προτεινόμενα αντικείμενα.

Ταξινομητής Naïve Bayes

Στα συστήματα προτιμήσεων γίνεται συχνά χρήστη του Μπευζιανού ταξινομητή[21]. Έστω ότι έχουμε τον γνωστό πλέον πίνακα mxn και ότι ο κάθε χρήστης μπορεί να αξιολογήσει ένα αντικείμενο με τις διακριτές τιμές $v_1,\ldots v_\kappa$. Έστω τώρα ότι ένας χρήστης u έχει βαθμολογήσει κάποια από τα αντικείμενα τα οποία δημιουργούν το σύνολο I_u. Προκειμένου να βρεθεί η προβλεπόμενη βαθμολογία που θα δώσει ο χρήστης u στο αντικείμενο j πρέπει να υπολογιστεί η πιθανότητα ο χρήστης να βαθμολογήσει το αντικείμενο j με κάθε μια από τις πιθανές τιμές χρησιμοποιώντας τον τύπο:

$$P(\kappa\rho\iota\tau\iota\kappa\acute{\eta}\ \gamma\iota\alpha\ \tau o\ j = v_i \mid \tau o\ \sigma\acute{\upsilon}\nu o\lambda o\ I_u) = \\ = \frac{P(\kappa\rho\iota\tau\iota\kappa\acute{\eta}\ \gamma\iota\alpha\ \tau o\ j = v_i) * P(\ \tau o\ \sigma\acute{\upsilon}\nu o\lambda o\ I_u \mid \kappa\rho\iota\tau\iota\kappa\acute{\eta}\ \gamma\iota\alpha\ \tau o\ j = v_i)}{P(\tau o\ \sigma\acute{\upsilon}\nu o\lambda o\ I_u)}$$

Όπου η πιθανότητα $P(\ \tau o\ \sigma\acute{\upsilon}\nu o\lambda o\ I_u \mid \kappa\rho\iota\tau\iota\kappa\acute{\eta} = v_i)$ υπολογίζεται από το γινόμενο των πιθανοτήτων ο χρήστης να έχει δώσει τη κριτική που έδωσε για ένα αντικείμενο του I_u με δεδομένο ότι βαθμολόγησε το αντικείμενο j με v_i για όλα τα αντικείμενα στο I_u.
Καθώς ο παρονομαστής είναι ίδιος για όλες τις πιθανότητες μπορεί να αγνοηθεί. Η προβλεπόμενη βαθμολογία μπορεί να προκύψει επιλέγοντας τη βαθμολογία που μεγιστοποιεί τη συγκεκριμένη πιθανότητα.

Ένα σημαντικό πρόβλημα που καλούμαστε να αντιμετωπίσουμε είναι αυτό της υπερπροσαρμογής (overfitting) το οποίο προκύπτει όταν ο πίνακας έχει πολλά κενά κελιά. Στη περίπτωση που ο αριθμός των αξιολογήσεων για ένα αντικείμενο είναι μικρός ο υπολογισμός της πιθανότητας η βαθμολογία ενός χρήστη για ένα αντικείμενο j να είναι v_i δεν είναι αρκετά έγκυρος. Προκειμένου να αντιμετωπίσουμε αυτό το ζήτημα χρησιμοποιούμε Λαπλασιανή εξομάλυνση (Laplacian smoothing) και υπολογίζουμε τη πιθανότητα ως εξής:

$$P(\kappa\rho\iota\tau\iota\kappa\acute{\eta}\ \gamma\iota\alpha\ \tau o\ j = v_i\ ) = \frac{q_i\ + a}{\sum_{t=1}^{\kappa} q_t + \kappa * a}$$

Όπου q_i ο αριθμός των χρηστών που έχουν βαθμολογήσει το αντικείμενο j με v_i και a μια παράμετρος λαπλασιανής εξομάλυνσης.

2.2.3 Συστήματα Προτιμήσεων με βάση το περιεχόμενο

Μέχρι τώρα έγινε αναφορά σε μεθόδους που προτείνουν αντικείμενα βασιζόμενες στις κριτικές που έδωσαν οι χρήστες σε άλλα αντικείμενα. Τα συστήματα προιμήσεων με βάση το περιεχόμενο διαφοροποιούνται καθώς στόχος τους είναι ο υπολογισμός της ομοιότητας δύο αντικειμένων σύμφωνα με το περιεχόμενό τους, το οποίο κυρίως είναι πλούσιο σε κείμενο. Από την επεξεργασία αυτή δημιουργείται ένα μοντέλο για τον εκάστοτε χρήστη δηλαδή το προφίλ του. Το προφίλ αυτό αποτελεί μια δομημένη αναπαράσταση των ενδιαφερόντων του και χρησιμοποιείται για να προταθούν καινούργια αντικείμενα.

Τα πλεονεκτήματα αυτής της προσέγγισης είναι πολλά. Αρχικά, στις προηγούμενες μεθόδους που περιγράψαμε ένα αντικείμενο με λίγες βαθμολογίες θα ήταν δύσκολο να προταθεί σε χρήστες. Το πρόβλημα αυτό είναι γνωστό ως πρόβλημα ψυχρής εκκίνησης (cold start problem). Στα συστήματα προτιμήσεων με βάση το περιεχόμενο προκειμένου να προταθεί ένα αντικείμενο σε ένα χρήστη δεν χρειάζεται να το έχουν βαθμολογήσει πολλοί χρήστες. Επιπλέον κάθε χρήστης δεν εξαρτάται από κανέναν άλλο καθώς οι προτάσεις που του γίνονται δεν βασίζονται σε ένα σύνολο γειτονικών χρηστών αλλά αφορούν αποκλειστικά τον ίδιο. Τέλος σημαντικό είναι το γεγονός ότι μέσα από τα συστήματα προτιμήσεων με βάση το περιεχόμενο υπάρχει διαφάνεια σχετικά με τις προτάσεις που γίνονται σε ένα χρήστη. Για παράδειγμα σε ένα σύστημα προτιμήσεων με ταινίες μπορεί να αναφερθεί ότι επειδή βαθμολογήθηκε υψηλά η ταινία x το σύστημα προτείνει την y. Έτσι και ο ίδιος ο χρήστης γνωρίζει περισσότερες πληροφορίες σχετικά με το λόγο που δέχτηκε αυτή τη πρόταση [22].

Τα δεδομένα που έχουμε συνήθως είναι πλούσια σε κείμενο και μη δομημένα συνεπώς αρχικά είναι απαραίτητα η μετατροπή τους προκειμένου να είναι πιο εύκολη η χρήση τους. Συνήθως αυτό συμβαίνει μετατρέποντας περιγραφές σε keywords. Η φάση αυτή ονομάζεται προ-επεξεργασία και εξαγωγή χαρακτηριστικών (feature extraction). Στη συνέχεια, χρησιμοποιώντας τα προηγούμενα χαρακτηριστικά δημιουργούμε το μοντέλο που αφορά τον κάθε χρήστη αποκλειστικά μέσα από το οποίο προκύπτει το προφίλ του χρήστη υπεύθυνο για τον συσχετισμό χαρακτηριστικών των αντικειμένων με ενδιαφέροντα του χρήστη. Οι προτάσεις που γίνονται στο χρήστη προκύπτουν με βάση το προφίλ του ολοκληρώνοντας τη διαδικασία του συστήματος. Παρακάτω αναλύεται κάθε μια από αυτές τις φάσεις.

2.2.3.1 Εξαγωγή Χαρακτηριστικών

Τα δεδομένα που υπάρχουν για ένα αντικείμενο σε μορφή κειμένου είναι συνήθως αδόμητα. Στη φάση της εξαγωγής χαρακτηριστικών είναι σημαντικό λοιπόν να μπορέσουμε να διαχωρίσουμε τις πληροφορίες που χαρακτηρίζουν ένα αντικείμενο από τις υπόλοιπες καθώς και να υπολογίσουμε τον βαθμό στον οποίο το καθορίζουν. Αρχικά πραγματοποιείται η αφαίρεση όλων των λέξεων που δεν προσδίδουν ιδιαίτερη σημασία για το αντικείμενο στο σύστημα προτιμήσεων (για παράδειγμα λέξεις όπως τα 'και', 'με'). Έπειτα πρέπει να ομαδοποιηθούν λέξεις που είναι σημασιολογικά ίδιες δηλαδή προέρχονται από την ίδια ρίζα ή βρίσκονται σε διαφορετικούς χρόνους. Με αυτό τον τρόπο δημιουργούνται οι λέξεις κλειδιά που θέλουμε να αξιολογήσουμε.

Η σημαντικότητα της εκάστοτε λέξης-κλειδί υπολογίζεται με την ανάθεση βαρών. Μια από τις πιο γνωστές μετρικές για τον υπολογισμό αυτών των βαρών είναι η συχνότητα όρου/αντίστροφη συχνότητα εγγράφου (term frequency/inverse document frequency) [23] η οποία ορίζεται ως εξής: Έστω N είναι ο συνολικός αριθμός αντικειμένων που μπορούμε να προτείνουμε σε ένα χρήστη και η λέξη-κλειδι k_j εμφανίζεται σε n_i από αυτά. Έστω επιπλέον ότι η μεταβλητή $f_{i,j}$ δηλώνει τις φορές που εμφανίστηκε η λέξη-κλειδί k_i στο αντικείμενο d_j Τότε η συχνότητα εμφάνισης της λέξης-κλειδί k_i στο αντικείμενο d_j ,$TF_{i,j}$ δίνεται από τον τύπο:

$$TF_{i,j} = \frac{f_{i,j}}{max_z f_{z,j}}$$

Όπου η μέγιστη τιμή υπολογίζεται για όλες τις λέξεις-κλειδιά που εμφανίζονται στο αντικείμενο d_j. Παρ' όλα αυτά είναι σημαντικό να λάβουμε υπ' όψη το γεγονός ότι μια λέξη με συχνή εμφάνιση στο σύνολο των αντικειμένων δεν μας δίνει σημαντικές πληροφορίες για το αν ένα αντικείμενο είναι σχετικό με κάποιο άλλο ή όχι. Για αυτό το λόγο χρησιμοποιούμε τη μετρική της αντίστροφης συχνότητας εγγράφου. Για μια λέξη κλειδί k_i ορίζεται ως:

$$IDF_i = \log\frac{N}{n_i}$$

Το τελικό βάρος μιας λέξης κλειδί υπολογίζεται από το γινόμενο $TF * IDF$.

Στη παραπάνω μέθοδο παρατηρούμε ότι δεν εντάσσεται πουθενά η βαθμολογία του χρήστη για το κάθε αντικείμενο. Παρακάτω περιγράφουμε μια σειρά από μετρικές που προκειμένου να υπολογίσουν το βαθμό σημαντικότητας μιας λέξης-κλειδί εκμεταλλεύονται και τις κριτικές των χρηστών όπως παρουσιάζονται στο [15].

Ο Gini δείκτης είναι από τους πιο συχνά χρησιμοποιήσιμους για τη διαδικασία επιλογής χαρακτηριστικών. Gini index: Έστω t ο αριθμός των πιθανών βαθμολογιών που μπορεί να δώσει ένας χρήστης για ένα αντικείμενο. Προκειμένου να υπολογίσουμε το βάρος που αντιστοιχεί στη λέξη-κλειδί w συλλέγουμε τα αντικείμενα που την περιέχουν. Έστω $\rho_1(w),\ldots,$ $\rho_t(w)$ ο αριθμός των φορών που η συγκεκριμένη λέξη-κλειδί έχει βαθμολογηθεί με την αντίστοιχη τιμή τότε:

$$Gini(w) = 1 - \sum_{i=1}^{t} p_i^2\,(w)$$

Μικρότερες τιμές του Gini δείκτη αντιστοιχούν σε λέξεις-κλειδιά μεγαλύτερης σημασίας.

Παρόμοια με τον Gini δείκτη μπορούμε να θέσουμε βάρη στις λέξεις-κλειδιά υπολογίζοντας την εντροπία τους. Και σε αυτή τη περίπτωση μικρότερες τιμές δηλώνουν μεγαλύτερη διαχωριστική δύναμη. Ορίζοντας τις ίδιες μεταβλητές με τη προηγούμενη παράγραφο ισχύει ότι:

$$Entropy(w) = -\sum_{i=1}^{t} p_i(w)\log(p_i(w))$$

Άλλος ένας τρόπος υπολογισμού βαρών για τις λέξεις-κλειδιά είναι η δοκιμασία x^2 (x^2-statistic). Προκειμένου να καταλάβουμε κατά πόσο η ύπαρξη μιας λέξης σε ένα αντικείμενο επηρεάζει την προτίμηση του από κάποιο χρήστη, υπολογίζουμε ως ανεξάρτητες πιθανότητες το να προτιμήσει κάποιος ένα αντικείμενο ή όχι σε σχέση με τον αν περιέχει την εξεταζόμενη λέξη. Η τιμή που θα προκύψει ονομάζεται προσδοκώμενη τιμή (expected value) και τη συμβολίζουμε με E_i. Έστω O_i η πραγματική τιμή των ατόμων που επέλεξαν το συγκεκριμένο αντικείμενο. Η μετρική x^2 υπολογίζεται από τον τύπο:

$$x^2 = \sum_{i=1}^{p} \frac{(O_i - E_i)^2}{E_i}$$

Όσο μεγαλύτερη η τιμή της μετρικής, τόσο μεγαλύτερη η απόκλιση ανάμεσα στη πραγματική τιμή και την αναμενόμενη και συνεπώς τόσο μεγαλύτερη η εξάρτηση ανάμεσα στο αντικείμενο και τη λέξη.

Στην περίπτωση που δεν μας ενδιαφέρει μόνο να εντάξουμε τη βαθμολογία στο βάρος των λέξεων αλλά μας ενδιαφέρει και η σχετική σειρά των βαθμολογιών μπορούμε να χρησιμοποιήσουμε τη μετρική της κανονικοποιημένης απόκλισης (normalized deviation). Μεγαλύτερες τιμές δηλώνουν μεγαλύτερη διακριτική ικανότητα. Έστω σ^2 η διασπορά των βαθμολογιών σε όλα τα αντικείμενα και $\mu^+(w)$, $\mu^-(w)$ η μέση τιμή των βαθμολογιών στα αντικείμενα που περιέχουν τη λέξη w και σε αυτά που δεν τη περιέχουν αντίστοιχα τότε η κανονικοποιημένη απόκλιση υπολογίζεται ως:

$$Dev(w) = \frac{|\mu^+(w) - \mu^-(w)|}{\sigma}$$

2.2.3.2 Δημιουργία προφίλ χρήστη και Φιλτράρισμα

Η δημιουργία ενός προφίλ για τον χρήστη είναι απαραίτητη καθώς βασιζόμαστε πάνω σε αυτό για να του προτείνουμε στη συνέχεια αντικείμενα. Το προφίλ ενός χρήστη περιέχει τις προτιμήσεις του σε σχέση με τα αντικείμενα του συνόλου. Η δημιουργία του πραγματοποιείται από την συλλογή και ανάλυση του περιεχομένου που έχει παρακολουθήσει και έχει βαθμολογήσει, δηλαδή, τις λέξεις κλειδιά που έχουν προκύψει από την προηγούμενη φάση για κάθε αντικείμενο. Ένας τρόπος αναπαράστασης του προφίλ είναι ως ένα διάνυσμα αποτελούμενο από βάρη που ορίζουν τις πιο σημαντικές λέξεις-κλειδιά για τον εκάστοτε χρήστη. Ο υπολογισμός αυτού του διανύσματος μπορεί να γίνει με χρήστη του Rocchio αλγόριθμου [24] καθώς και με χρήση Μπευζιανού ταξινομητή όπως συμβαίνει στο [25].

Το προφίλ που δημιουργείται με τις τεχνικές που αναφέρθηκαν στη προηγούμενη παράγραφο χρησιμοποιείται στη συνέχεια για να προκύψουν τα αντικείμενα που ενδεχομένως ενδιαφέρουν τον χρήστη αλλά δεν τα έχει επιλέξει ακόμα. Ορίζουμε τη συνάρτηση χρησιμότητας (utility function) uf(u,o) όπου για μεγαλύτερες τιμές τις έχουμε και μεγαλύτερη συσχέτιση ανάμεσα στον χρήστη (u) και το αντικείμενο (o) [23]. Αναπαριστώντας το προφίλ του χρήστη και το αντικείμενο ως διανύσματα με βάρη για κάθε λέξη-κλειδί , έστω w_u και w_o αντίστοιχα, η συνάρτηση χρησιμότητας μπορεί να αναπαρασταθεί με μια μετρική όπως η ομοιότητα συνημίτονου σύμφωνα με την οποία ισχύει:

$$uf(u,o) = \frac{\sum_{i=1}^{K} w_{i,u} w_{i,o}}{\sqrt{\sum_{i=1}^{K} w_{i,u}^2} \sqrt{\sum_{i=1}^{K} w_{i,o}^2}}$$

Όπου K είναι το σύνολο των λέξεων-κλειδιών.

Ένας εναλλακτικός τρόπος δημιουργίας προτάσεων για ένα χρήστη είναι η χρήση του Μπευζιανού ταξινομητή [26]. Έστω ότι έχουμε ένα σύνολο αντικειμένων από το οποίο προέκυψε το προφίλ του χρήστη και ένα σύνολο που αποτελείται από τα υπόλοιπα αντικείμενα. Χρησιμοποιώντας το μοντέλο Μπερνούλι όπως υποδεικνύεται στο [15] , αγνοούμε τη συχνότητα εμφάνισης των λέξεων-κλειδιών σε ένα αντικείμενο και εστιάζουμε στην παρουσία (+1) ή απουσία τους (0). Έστω λοιπόν ότι έχουμε ένα αντικείμενο X το οποίο αποτελείται από ένα δυαδικό διάνυσμα που μας πληροφορεί για τις λέξεις κλειδιά που περιέχει, έστω [k_1,..,k_d]. Μας ενδιαφέρει να υπολογίσουμε τη πιθανότητα να αρέσει στον χρήστη ένα αντικείμενο με δεδομένο ότι περιέχει τις συγκεκριμένες λέξεις-κλειδιά. Θεωρώντας ότι οι λέξεις κλειδιά είναι ανεξάρτητες μεταξύ τους υπολογίζουμε τις παραπάνω πιθανότητες ως εξής:

$$P(user\ likes\ X|[k_1,..,k_d]) = \frac{P(user\ likes\ X) * P([k_1,..,k_d]|user\ likes\ X)}{P([k_1,..,k_d])}$$

Το οποίο αγνοώντας τον παρονομαστή καθώς είναι ίδιο σε όλες τις πιθανότητες γίνεται:

$$P(user\ likes\ X|[k_1,..,k_d]) = P(user\ likes\ X) * \prod_{i=1}^{d} P(k_i|user\ likes\ X)$$

Με αντίστοιχο τρόπο υπολογίζουμε την πιθανότητα να μην αρέσει στον χρήστη το συγκεκριμένο αντικείμενο και ανάλογα με την μεγαλύτερη πιθανότητα του το προτείνουμε ή όχι.

Παρ' όλα τα θετικά χαρακτηριστικά των συστημάτων προτιμήσεων που βασίζονται στο περιεχόμενο παρουσιάζουν και μια σειρά από μειονεκτήματα. Όταν εισέρχεται ένας νέος χρήστης στο σύστημα καθώς δεν υπάρχει κάποια πληροφορία σχετικά με τα αντικείμενα ή

τις θεματικές που τον ενδιαφέρουν είναι δύσκολη η παροχή ουσιαστικών προτάσεων. Επιπλέον καθώς η προτάσεις που γίνονται στον χρήστη βασίζονται αποκλειστικά σε αντικείμενα που του αρέσουν, υπάρχει ο κίνδυνος να προτείνονται συνέχεια αντικείμενα των ίδιων κατηγοριών. Προκαλούνται έτσι δύο σημαντικά ζητήματα. Αρχικά υπάρχουν θεματικές με τις οποίες ο χρήστης δεν έχει έρθει ποτέ σε επαφή οι οποίες θα μπορούσαν να είναι ενδιαφέρουσες για αυτόν. Από την άλλη σε μερικές περιπτώσεις αντικείμενα με πολύ μεγάλη ομοιότητα σε άλλα δεν θα έπρεπε να προτείνονται καθώς ο χρήστης δεν ενδιαφέρεται για κάτι σχεδόν ίδιο με ένα αντικείμενο που έχει ήδη εντοπίσει. Για αυτό τον λόγο έχουν προταθεί συστήματα προτιμήσεων που συνδυάζουν τις δύο μεθόδους που περιγράψαμε προκειμένου να εκμεταλλευτούν τα πλεονεκτήματα και των δύο.

2.2.4 Υβριδικά Συστήματα Προτιμήσεων

Τα υβριδικά συστήματα προτιμήσεων συνδυάζουν δύο ή περισσότερες τεχνικές συστάσεων για να αποκτήσουν καλύτερη απόδοση με λιγότερα μειονεκτήματα από κάθε μέθοδο ξεχωριστά. Συνήθως το συνεργατικό φιλτράρισμα συνδυάζεται με κάποια άλλη τεχνική [27]. Παρακάτω θα περιγράψουμε τις πιο δημοφιλείς κατηγορίες υβριδικών συστημάτων προτιμήσεων.

2.2.4.1 Σταθμισμένα

Στα σταθμισμένα υβριδικά συστήματα προτιμήσεων η τελική προβλεπόμενη βαθμολογία ενός χρήστη για ένα αντικείμενο προκύπτει από τον συνδυασμό των αποτελεσμάτων πολλών συστημάτων προτιμήσεων εντάσσοντας μια σειρά από βάρη σε κάθε αποτέλεσμα. Πιο συγκεκριμένα έστω R ο πίνακα nxm που περιέχει τις τελικές προβλεπόμενες και πραγματικές βαθμολογίες για κάθε χρήστη του συστήματος. Έστω επίσης $R_1,\ldots R_d$ οι πίνακες που έχουν προκύψει μετά από κάθε έναν από τους d αλγόριθμους. Ο τελικός πίνακας αρά προκύπτει από τον τύπο:

$$R = \sum_{i=1}^{d} a_i R_i$$

Όπου a_i το βάρος που αντιστοιχεί σε κάθε αλγόριθμο. Η μεθοδολογία για τον υπολογισμό των βαρών μπορεί να είναι είτε κάποια χειριστική συνάρτηση είτε να χρησιμοποιεί κάποιο στατιστικό μοντέλο. Στόχος είναι να δίνεται μεγαλύτερο βάρος στο πιο ακριβές σύστημα. Για παράδειγμα στο [28] γίνεται χρήστη ενός υβριδικού σταθμισμένου συστήματος προτιμήσεων

στο οποίο τα βάρη αρχικά είναι ίδια για όλα τα συστήματα αλλά στη πορεία με την επιβεβαίωση και την απόρριψη των προτάσεων από τον χρήστη ρυθμίζονται αναλόγως.

2.2.4.2 Συστήματα εναλλαγής

Στα συστήματα εναλλαγής ο αλγόριθμος που χρησιμοποιείται βασιζόμενος στις ανάγκες που προκύπτουν αλλάζει το σύστημα προτιμήσεων που θα προτείνει αντικείμενα στον χρήστη. Αρχικά τα συγκεκριμένα συστήματα προτάθηκαν ως ένας τρόπος να αντιμετωπιστεί το πρόβλημα της ψυχρής εκκίνησης (cold-start problem) [27] ξεκινώντας από το σύστημα που έχει καλύτερη επίδοση στα αρχικά στάδια και αλλάζοντας σύστημα προτιμήσεων στη συνέχεια.

Χαρακτηριστικό παράδειγμα χρήσης του συγκεκριμένου συστήματος αποτελεί το [29] στο οποίο γίνεται χρήση δύο συστημάτων προτιμήσεων που βασίζονται στο περιεχόμενο και ενός που βασίζεται στο συνεργατικό φιλτράρισμα. Οι αρχικές προτάσεις γίνονται με τα συστήματα που βασίζονται στο περιεχόμενο και στη συνέχεια εάν το σύστημα δεν κάνει προτάσεις αρκετά ικανοποιητικές γίνεται χρήση του τρίτου συστήματος προτιμήσεων. Η ένταξη της συνεργατικής μεθοδολογίας στον αλγόριθμο επιλύει το ζήτημα των περιορισμένων προτάσεων καθώς κάνει δυνατή τη πρόταση αντικειμένων που ανήκουν σε πεδία που ο χρήστης μπορεί να μην γνωρίζει και που δεν είναι σημασιολογικά κοντά με οτιδήποτε προτιμάει. Μια επιπλέον σημαντική παράμετρος είναι αυτή που καθορίζει πότε πρέπει να γίνεται η εναλλαγή των συστημάτων προσθέτοντας κάποια πολυπλοκότητα στο σύστημα.

2.2.4.3 Αλληλουχία (Cascade)

Η βασική ιδέα στην αλληλουχία υβριδικών συστημάτων προτιμήσεων είναι η δημιουργία ενός ιεραρχικού υβριδικού συστήματος, στο οποίο ένα πιο αδύναμο σύστημα προτιμήσεων δεν θα μπορεί να ανατρέψει τις αποφάσεις που έχουν παρθεί από ένα ισχυρότερο αλλά μπορεί να τις τελειοποιήσει [30]. Μια σειρά από συστήματα προτιμήσεων ακολουθώντας σειρά προτεραιότητας επεξεργάζονται τα αποτελέσματα που προκύπτουν από το σύστημα προτιμήσεων με τη μεγαλύτερη προτεραιότητα με στόχο να αντιμετωπίσει πιθανές ισοπαλίες που προέκυψαν ανάμεσα σε αντικείμενα βελτιώνοντας έτσι τις τελικές προτάσεις που γίνονται στον χρήστη.

Όπως περιγράφεται και στο [27] το σύστημα προτιμήσεων για εστιατόρια Entree επέστρεφε πάρα πολλά αποτελέσματα με κοινές βαθμολογίες τα οποία συνεπώς δεν μπορούσαν να συγκριθούν μεταξύ τους δυσκολεύοντας την δημιουργία προτάσεων στους χρήστες. Προκειμένου να αντιμετωπιστεί το παραπάνω πρόβλημα δημιουργήθηκε το EntreeC που βασίζεται σε υβριδικό σύστημα προτιμήσεων αλληλουχίας στο οποίο οι βαθμολογίες των αντικειμένων που προκύπτουν ίδιες από το πρώτο σύστημα αποτελούν είσοδο ενός συστήματος προτιμήσεων με συνεργατικό φιλτράρισμα το οποίο τις διαχωρίζει.

2.2.4.4 Αύξηση Χαρακτηριστικών

Τα συγκεκριμένα συστήματα χρησιμοποιούν τα αποτελέσματα ενός συστήματος προτιμήσεων προκειμένου να δημιουργήσουν χαρακτηριστικά ως είσοδο για το επόμενο σύστημα. Παρά το γεγονός ότι φαινομενικά μοιάζει η μεθοδολογία τους με αυτή της αλληλουχίας παρουσιάζει θεμελιώδεις διαφορές. Το κοινό τους στοιχείο είναι ότι και στα δύο υβριδικά συστήματα το ιεραρχικά πρώτο σύστημα προτιμήσεων επηρεάζει το δεύτερο. Στα υβριδικά συστήματα με αύξηση χαρακτηριστικών όμως, τα χαρακτηριστικά που χρησιμοποιούνται ως είσοδο για το επόμενο σύστημα περιέχουν την έξοδο του προηγούμενου συστήματος σε αντίθεση με τα υβριδικά συστήματα αλληλουχίας στα οποία συνδυάζονται τα αποτελέσματα των συστημάτων προτιμήσεων με βάση την ιεραρχία που έχει οριστεί [27].

Το σύστημα Libra [31] συνδυάζει το σύστημα προτιμήσεων της Amazon με έναν Μπευζιανό ταξινομητή. Πιο συγκεκριμένα η Amazon σε κάθε αντικείμενο της εμφάνιζε μια σειρά από σχετικούς συγγραφείς και σχετικούς τίτλους που προκύπτουν μέσω ενός συστήματος προτιμήσεων που βασίζεται στο συνεργατικό φιλτράρισμα. Με τη χρήση αυτών των χαρακτηριστικών σε σύστημα προτιμήσεων με βάση το περιεχόμενο, το Libra έκανε προτάσεις βιβλίων που ήταν πολύ ακριβείς.

2.2.4.5 Meta-Level

Τα Meta-Level συστήματα προτιμήσεων αντί να χρησιμοποιούν τα χαρακτηριστικά που προκύπτουν από ένα σύστημα προτιμήσεων ως είσοδο σε ένα άλλο χρησιμοποιούν ολόκληρο το μοντέλο που δημιουργείται στο αρχικό σύστημα. Ο πιο συνήθης συνδυασμός είναι αυτός ανάμεσα σε ένα σύστημα βασισμένο στο περιεχόμενο και ένα σύστημα συνεργατικού φιλτραρίσματος. Χαρακτηριστικό παράδειγμα αποτελεί το [32], στο οποίο γίνεται αρχικά χρήση του συστήματος προτιμήσεων βασισμένο στο περιεχόμενο δημιουργώντας έτσι για

κάθε χρήστη ένα διάνυσμα που περιέχει βάρη για κάθε μια από τις λέξεις-κλειδιά. Στη συνέχεια ο πίνακας που προκύπτει ανάμεσα σε χρήστες και λέξεις-κλειδιά χρησιμοποιείται από το σύστημα προτιμήσεων με συνεργατικό φιλτράρισμα με για να δημιουργήσει την ομάδα των γειτονικών χρηστών του χρήστη που μας ενδιαφέρει. Στη συνέχεια οι προβλεπόμενες βαθμολογίες προκύπτουν από τον σταθμισμένο μέσο όρο των βαθμολογιών όλων τον χρηστών της ομάδας.

Η συγκεκριμένη προσέγγιση είναι επίσης γνωστή και ως συνεργασία μέσω περιεχομένου (collaboration via content) και εμφάνισε σημαντικά καλύτερα αποτελέσματα σε σχέση με την εφαρμογή κάθε ενός από τα συστήματα προτιμήσεων ξεχωριστά. Το βασικότερο πλεονέκτημα των meta-level μεθόδων είναι ότι το μοντέλο που προκύπτει από το πρώτο σύστημα προτιμήσεων είναι μια συμπυκνωμένη αναπαράσταση των ενδιαφερόντων ενός χρήστη καθώς ο πίνακας έχει πολύ λιγότερα κενά κελιά σε σχέση με τον πίνακα των βαθμολογιών. Συνεπώς το δεύτερο σύστημα προτιμήσεων μπορεί να κάνει προτάσεις με πολύ μεγαλύτερη ακρίβεια.

2.2.4.6 Συνδυασμός χαρακτηριστικών

Τα συγκεκριμένα συστήματα συλλέγουν χαρακτηριστικά από διαφορετικές πηγές δεδομένων και τα συνδυάζουν προκειμένου να δημιουργήσουν ένα πίνακα με περισσότερες πληροφορίες από ότι ο κλασσικός πίνακας mxn ανάμεσα σε χρήστες και αντικείμενα. Στη συνέχεια αυτά τα χαρακτηριστικά χρησιμοποιούνται από ένα σύστημα προτιμήσεων το οποίο πραγματοποιεί και τις τελικές προτάσεις στον χρήστη. Συνήθως η πρόβλεψη πραγματοποιείται από ένα σύστημα προτιμήσεων που βασίζεται στο περιεχόμενο και το οποίο κάνει χρήστη επιπλέον χαρακτηριστικών που προέκυψαν από μεθόδους συνεργατικού φιλτραρίσματος [15]. Τα υβριδικά συστήματα προτιμήσεων με συνδυασμό χαρακτηριστικών έχουν το πλεονέκτημα ότι επιτρέπουν την χρήση βαθμολογιών των χρηστών που χρησιμοποιούνται και στις συνεργατικές μεθόδους χωρίς όμως να βασίζονται αποκλειστικά σε αυτές. Έτσι δεν έρχονται αντιμέτωπα με τα προβλήματα που προκύπτουν για αντικείμενα που έχουν μικρό αριθμό βαθμολογιών.

Αυτή η προσέγγιση παρουσιάζεται και στο [33] όπου γίνεται χρήση του Ripper προκειμένου να προτείνει ταινίες στους χρήστες χρησιμοποιώντας τόσο τις κριτικές τους όσο και τα χαρακτηριστικά των ταινιών. Παρατηρήθηκε σημαντική βελτίωση στην ακρίβεια των προτάσεων συγκριτικά με τη χρήση του συστήματος προτιμήσεων που βασίζεται στο

συνεργατικό φιλτράρισμα, είναι σημαντικό όμως να αναφερθεί ότι αυτή η βελτίωση προήλθε επιλέγοντας συγκεκριμένα χαρακτηριστικά περιεχομένου.

2.2.4.7 Μεικτά Συστήματα

Η τελευταία κατηγορία υβριδικών συστημάτων προτιμήσεων διαφέρει σημαντικά από τις υπόλοιπες. Μέχρι τώρα αναφερθήκαμε σε μεθόδους που συνδυάζουν τα αποτελέσματα συστημάτων προτιμήσεων για να προκύψει ένα κοινό αποτέλεσμα μέσα από κατάλληλη επεξεργασία. Στα μεικτά συστήματα παρουσιάζονται στον χρήστη προτάσεις από διαφορετικά συστήματα, οι οποίες είναι ανεξάρτητες μεταξύ τους, ταυτόχρονα. Το βασικό χαρακτηριστικό αυτής της προσέγγισης είναι ότι οι βαθμολογίες που προκύπτουν για κάθε πρόταση χρησιμοποιούνται προκειμένου να δημιουργηθεί μια κατάταξη κατά την παρουσίαση τους στον χρήστη. Όπως είναι λογικό η χρήση μεικτών συστημάτων προτιμήσεων επιλύει σε μεγάλο βαθμό το πρόβλημα που αντιμετωπίζουμε με την εισαγωγή καινούργιου αντικειμένου στο σύνολο.

Συνήθως τα μεικτά συστήματα χρησιμοποιούνται όταν θέλουμε να προτείνουμε σε ένα χρήστη μεγάλο αριθμό αντικειμένων. Χαρακτηριστικό παράδειγμα χρήσης της συγκεκριμένης μεθόδου είναι το PTV [34] το οποίο δημιουργεί προσωποποιημένα τηλεοπτικά προγράμματα. Χρησιμοποιεί τις προτάσεις ενός συστήματος προτιμήσεων που βασίζεται στο περιεχόμενο και ενός συστήματος που βασίζεται στο συνεργατικό φιλτράρισμα. Επιπλέον μεικτό σύστημα προτιμήσεων χρησιμοποιήθηκε και στο [35] προκειμένου να προταθούν ονόματα με βάση τις προτιμήσεις ενός χρήστη. Παρατηρήθηκε ότι ο συνδυασμός των προτάσεων τριών διαφορετικών συστημάτων προτιμήσεων έχει καλύτερα αποτελέσματα από τη χρήση καθενός ξεχωριστά.

2.2 Συστήματα Προτιμήσεων και Skyline Αλγόριθμοι

Τα συστήματα προτιμήσεων έρχονται αντιμέτωπα με δύο βασικές προκλήσεις. Στην εποχή μας η αύξηση των δεδομένων σε κάθε τομέα είναι ραγδαία με αποτέλεσμα οι προτάσεις που γίνονται σε ένα χρήστη να μην είναι οι βέλτιστες δυνατές. Ο αριθμός των προτεινόμενων αντικειμένων που προκύπτουν από ένα σύστημα προτιμήσεων με ίδια βαθμολογία για έναν χρήστη ενδέχεται να είναι δυσλειτουργικά μεγάλο με αποτέλεσμα η ακρίβεια στις προτάσεις να μη είναι η επιθυμητή. Επιπλέον είναι σημαντικό να αναγνωρίσουμε το γεγονός ότι δύο χρήστες μπορεί να έχουν δώσει την ίδια βαθμολογία σε ένα αντικείμενο για πολύ

διαφορετικούς λόγους. Αυτό έχει ως αποτέλεσμα το σύστημα προτιμήσεων να μην μπορεί να αναγνωρίσει τους διαφορετικούς παράγοντες που οδήγησαν σε δυο φαινομενικά ίδιες βαθμολογίες και να προτείνει και στους δύο χρήστες πανομοιότυπα αντικείμενα.

Ένας τρόπος να αντιμετωπίσουμε τα παραπάνω είναι να δώσουμε έμφαση στα γενικότερα χαρακτηριστικά ενός αντικειμένου καθώς συνήθως η επιλογή ενός αντικειμένου ως προς ένα άλλο βασίζεται σε μια σειρά από χαρακτηριστικά του και όχι μόνο σε ένα. Για αυτό τον λόγο έχουν προταθεί μια σειρά από μεθόδους που συνδυάζουν τα συστήματα προτιμήσεων με τους skyline αλγορίθμους και οι οποίες παρουσιάζονται παρακάτω.

Στο [36] θίγεται το ζήτημα των αξιολογήσεων πολλαπλών κριτηρίων (multi-criteria ratings) στα συστήματα προτιμήσεων. Έστω ότι ένας χρήστης έχει βαθμολογήσει ένα αντικείμενο ως προς μια σειρά χαρακτηριστικών του δίνοντας διαφορετική βαθμολογία σε κάθε ένα από αυτά στην ίδια κλίμακα. Για κάθε ένα από τα χαρακτηριστικά υπολογίζεται ξεχωριστά η προβλεπόμενη τιμή που προκύπτει με χρήση τεχνικών συνεργατικού φιλτραρίσματος. Κάθε χαρακτηριστικό είναι και μια διάσταση σε έναν πολυδιάστατο χώρο. Έπειτα εφαρμόζοντας τους αλγόριθμους για υπολογισμό του skyline συνόλου σε αυτό τον χώρο, επιστρέφονται τα skyline αντικείμενα που είναι και αυτά που θα προταθούν στον χρήστη. Τα συγκεκριμένα αντικείμενα έχουν προκύψει ως βέλτιστα λαμβάνοντας υπ' όψη όλα τα χαρακτηριστικά που μπορεί να επηρεάσουν την επιλογή ενός αντικειμένου από τον εκάστοτε χρήστη. Στη περίπτωση που τα skyline αντικείμενα που επιστρέφονται είναι πάρα πολλά μια αντιμετώπιση είναι η ταξινόμηση σύμφωνα με το σύνολο των αντικειμένων κυριαρχούν.

Μια διαφορετική μέθοδος για την αντιμετώπιση των αξιολογήσεων πολλαπλών κριτηρίων παρουσιάζεται στο [37] όπου δημιουργείται ένα προσωπικό skyline σύνολο για κάθε χρήστη βασισμένο στις βαθμολογίες που έχουν δώσει για κάθε διάσταση-χαρακτηριστικό άλλοι χρήστες που παρουσιάζουν μεγάλη ομοιότητα ως προς αυτόν. Καταφέρνει έτσι να διαχωρίσει τα μοτίβα στις βαθμολογίες από τα κριτήρια επιλογής του κάθε χρήστη.

Προκειμένου να αντιμετωπιστεί το ζήτημα του μεγάλου πλήθους δεδομένων στον τομέα των υπηρεσιών του διαδικτύου που βασίζονται στη ποιότητα εξυπηρέτησης (QoS-Based Web Services) στο [38] γίνεται χρήση υβριδικού συστήματος προτιμήσεων σε συνδυασμό με skyline αλγόριθμους. Πιο συγκεκριμένα υπολογίζεται αρχικά το skyline σύνολο των υπηρεσιών με χρήση του αλγορίθμου Branch and Bound. Στη συνέχεια υπολογίζεται η

ομοιότητα ανάμεσα στις υπηρεσίες και τα χαρακτηριστικά που επιθυμεί ο χρήστης με μετρικές ομοιότητας. Έτσι προκύπτει ένα σύνολο από αντικείμενα τα οποία είναι πιο κοντά στους περιορισμούς του εκάστοτε χρήστη. Το υβριδικό σύστημα βασίζεται στη χρήση τεχνικών συνεργατικού φιλτραρίσματος βασισμένα στο αντικείμενο και βασισμένα στον χρήστη. Έτσι δημιουργούνται δύο σύνολα γειτόνων, ένα ως προς τον χρήστη και ένα ως προς τις υπηρεσίες. Οι τελικές προβλέψεις προκύπτουν με χρήση και των δύο συνόλων.

3 Μεθοδολογία

Στόχος της παρούσας διπλωματικής είναι η μελέτη της επίδρασης του skyline συνόλου στην ικανοποίηση του χρήστη ως προς τα προτεινόμενα αποτελέσματα ενός συστήματος προτιμήσεων. Το κεφάλαιο αυτό είναι δομημένο ως εξής: Αρχικά περιγράφεται η αρχιτεκτονική που επιλέχθηκε, στη συνέχεια αναλύονται τα δεδομένα που χρησιμοποιήθηκαν για την εκτέλεση του πειράματος και τέλος η πειραματική διαδικασία.

3.1 Αρχιτεκτονική

Στόχος μας είναι να κατασκευάσουμε ένα σύστημα προτιμήσεων που βασίζεται στο περιεχόμενο και κάνει χρήση της ομοιότητας συνημίτονου, το οποίο χρησιμοποιεί το skyline σύνολο μιας βάσης δεδομένων για τις προτάσεις του. Η αρχιτεκτονική του συστήματος αποτελείται από δυο φάσεις. Η πρώτη αφορά τον υπολογισμό του skyline συνόλου, κατά την οποία εφαρμόζεται ο κατάλληλος αλγόριθμος στα δεδομένα της βάσης. Το σύνολο που προκύπτει χρησιμοποιείται στη δεύτερη φάση, στην οποία συμμετέχει και ο χρήστης ορίζοντας τις προτιμήσεις του με βάση τις οποίες προκύπτουν οι αντίστοιχες προτάσεις από το σύστημα προτιμήσεων που κατασκευάστηκε. Η εικόνα που ακολουθεί περιγράφει την αρχιτεκτονική που μόλις περιγράψαμε. Για την πραγματοποίηση της παρακάτω αρχιτεκτονικής θα γίνει χρήση μιας βάσης που περιέχει πληροφορίες για ενοικιαζόμενα διαμερίσματα η οποία περιγράφεται αναλυτικότερα στην ενότητα 3.2. Ο αλγόριθμος που επιλέχθηκε για τον υπολογισμό του skyline συνόλου είναι ο Block Nested Loop. Στη συνέχεια ο χρήστης μέσα από μια σειρά φίλτρων θα κληθεί να επιλέξει το διαμέρισμα της αρεσκείας του. Μέσα από αυτή την επιλογή θα μπορέσει να πραγματοποιηθεί συλλογή χαρακτηριστικών που προτιμάει ο χρήστης σε ένα διαμέρισμα. Αυτά τα δεδομένα θα χρησιμοποιηθούν στη συνέχεια από το σύστημα προτιμήσεων το οποίο θα προτείνει τελικά μια σειρά από δωμάτια του skyline συνόλου που ταιριάζουν στις προτιμήσεις του χρήστη. Η

διαδικασία επιλογής ιδανικού διαμερίσματος και η πρόταση εναλλακτικών από το σύστημα προτιμήσεων μπορεί να επαναληφθεί από τον χρήστη όσες φορές επιθυμεί.

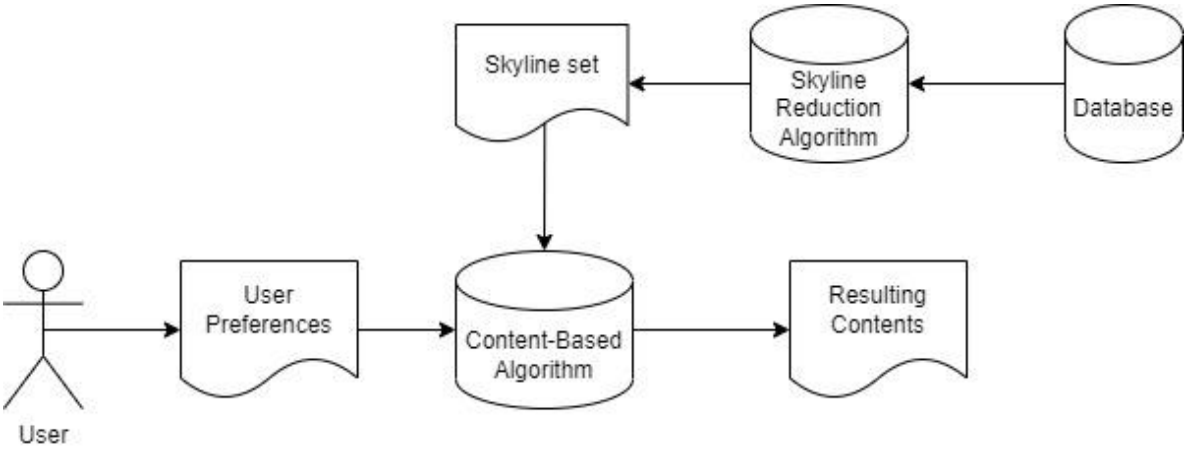


Figure 5 Αρχιτεκτονική

3.2 Δεδομένα

Για την πραγματοποίηση του πειράματος έγινε χρήση της βάσης δεδομένων που βρίσκεται στο [39]. Η συγκεκριμένη βάση περιέχει στοιχεία για ενοικιαζόμενα διαμερίσματα στο Μιλάνο. Στον χώρο της ενοικίασης διαμερισμάτων τα συστήματα προτιμήσεων έχουν πολύ σημαντικό ρόλο, για αυτό το λόγο επιλέχθηκε η συγκεκριμένη βάση δεδομένων στο συγκεκριμένο πείραμα. Επιπλέον η δομή των δεδομένων της συγκεκριμένης βάσης ήταν ιδιαίτερα βοηθητική για την χρήση skyline αλγορίθμων. Πιο συγκεκριμένα η βάση περιέχει συνολικά 61 στήλες με δεδομένα για κάθε ενοικιαζόμενο διαμέρισμα όπως αριθμός των ατόμων που μπορεί να φιλοξενήσει το διαμέρισμα, πληροφορίες σχετικά με τον ιδιοκτήτη καθώς και παροχές του διαμερίσματος.

3.2.1 Επεξεργασία των δεδομένων

Τα δεδομένα που επιθυμούμε να περιέχονται στη βάση μετά την επεξεργασία πρέπει να ικανοποιούν δύο διαφορετικά σκέλη της αρχιτεκτονικής που περιεγράφηκε στο 3.1. Αρχικά χρειαζόμαστε μια κατηγορία δεδομένων που θα χρησιμοποιηθεί προκειμένου να εκτελεστεί ο αλγόριθμος για τον υπολογισμό του skyline συνόλου. Για αυτή τη κατηγορία επιλέχθηκαν 3 κατηγορίες αριθμητικών δεδομένων (κόστος ανά βράδυ, βαθμολογία τοποθεσίας, βαθμολογία κριτικών). Επιλέχθηκαν τα συγκεκριμένα δεδομένα καθώς είναι γνωστό ότι για διαμερίσματα με παρόμοια χαρακτηριστικά είναι προτιμότερα αυτά που έχουν χαμηλότερο κόστος και υψηλότερες βαθμολογίες στην τοποθεσία και στις κριτικές. Η δεύτερη κατηγορία δεδομένων που χρειαζόμαστε, είναι αυτή στην οποία θα βασίζεται το σύστημα προτιμήσεων

για να παρέχει προτάσεις στον χρήστη. Πιο συγκεκριμένα με την επιλογή ενός διαμερίσματος το σύστημα θα πρέπει χρησιμοποιώντας τα συγκεκριμένα δεδομένα να προτείνει στον χρήστη άλλα παρόμοια διαμερίσματα. Για αυτό το σκοπό επιλέχθηκαν 11 κατηγορίες αριθμητικών δεδομένων που αφορούν τις παροχές κάθε διαμερίσματος (τηλεόραση, Ίντερνετ, Κλιματισμός, Κουζίνα, Πρωινό, Ανελκυστήρας, Θέρμανση, Πλυντήριο Ρούχων, Σίδερο, Luggage Drop-off, Κάπνισμα σε εσωτερικό χώρο). Επιπλέον από τα 61 διαφορετικά είδη δεδομένων διατηρήθηκε και μια σειρά από αριθμητικά δεδομένα που θεωρήθηκαν απαραίτητα για την καλύτερη περιγραφή των ενοικιαζόμενων διαμερισμάτων (συνολικός αριθμός σε μπάνια, κρεβατοκάμαρες, κρεβάτια καθώς και συνολικός αριθμός κριτικών, προσβασιμότητα αναπηρικών αμαξιδίων, αριθμός ατόμων που μπορούν να φιλοξενηθούν). Συνολικά λοιπόν από τις 61 στήλες δεδομένων η βάση μετά την επεξεργασία αποτελούταν από 21.

Skyline σύνολο

Στη παρούσα διπλωματική μας ενδιαφέρει να διαχωρίσουμε τα στοιχεία ενός συνόλου που είναι επιθυμητά για έναν χρήστη. Για παράδειγμα στη περίπτωση που το σύνολο μας αφορά ενοικίαση κατοικίας, γνωρίζουμε ότι εάν δύο σπίτια έχουν ίδια χαρακτηριστικά και διαφέρουν μόνο στο κόστος, ο δυνητικός χρήστης μας θα προτιμήσει σίγουρα την κατοικία που κοστίζει λιγότερο. Έτσι λοιπόν εφαρμόζοντας τη σχέση κυριαρχίας σε αυτές τις δύο κατοικίες, αυτή με το υψηλότερο κόστος δεν θα βρίσκεται τελικά στο skyline σύνολο. Αντιθέτως όταν πρόκειται για δύο κατοικίες εκ των οποίων η μια κοστίζει παραπάνω αλλά έχει περισσότερα τετραγωνικά μέτρα δεν μπορούμε να γνωρίζουμε εκ των προτέρων ποιο από τα δύο χαρακτηριστικά είναι πιο σημαντικό για τον χρήστη μας και συνεπώς θέλουμε και οι δύο κατοικίες να βρίσκονται στο skyline σύνολό, κάτι που θα συμβαίνει καθώς μεταξύ τους δεν ορίζεται Pareto κυριαρχία. Αυτό που επιτυγχάνουμε λοιπόν με τη δημιουργία skyline συνόλου είναι να παρουσιάζουμε στον χρήστη μας μόνο δεδομένα για τα οποία δεν μπορούμε να γνωρίζουμε εκ των προτέρων την επιλογή του φιλτράροντας δεδομένα που γνωρίζουμε ότι δεν θα προτιμούσε ανεξαρτήτως των προτιμήσεων του.

Για τον υπολογισμό του skyline συνόλου χρησιμοποιήθηκε ο Block Nested Loop αλγόριθμος που περιγράφεται στην ενότητα 2. Η αρχική βάση δεδομένων αποτελείται από 9321 ενοικιαζόμενα διαμερίσματα. Τα διαμερίσματα χωρίστηκαν σε ομάδες ανάλογα με τον αριθμό των ατόμων που μπορούν να φιλοξενήσουν και ο αλγόριθμος εκτελέστηκε σε κάθε μια από αυτές.

Ένα σημαντικό ζήτημα που έπρεπε να διευθετήσουμε ήταν η σχέση κυριαρχίας, σύμφωνα με την οποία καθορίζεται πότε ένα διαμέρισμα μπορεί να θεωρηθεί ασύγκριτο με ένα άλλο ως προς μια διάσταση και πότε να μείνει εκτός του skyline συνόλου επειδή κυριαρχείται από κάποιο άλλο. Για παράδειγμα εάν χρησιμοποιούσαμε σαν χαρακτηριστικό την τιμή ενός διαμερίσματος ανά βράδυ, ένα διαμέρισμα που κοστίζει 50 ευρώ θα αποκλειστεί από το skyline σύνολο από κάποιο διαμέρισμα που κοστίζει 49 ευρώ. Ενώ κάτι τέτοιο είναι λογικό να συμβεί μαθηματικά, στη πραγματικότητα δεν είναι κάτι που επιθυμούμε, καθώς για οποιονδήποτε χρήστη η διαφορά του ενός ευρώ δεν αποτελεί αιτία επιλογής του συγκεκριμένου διαμερίσματος σε σχέση το άλλο. Προκειμένου να αποφύγουμε το παραπάνω, πραγματοποιήθηκε κατάταξη των δεδομένων σε κουβάδες (bins) με βάση την αριθμητική τιμή τους στα τρία χαρακτηριστικά που μας ενδιαφέρουν για την εκτέλεση του skyline αλγορίθμου. Η συγκεκριμένη μέθοδος είναι γνωστή ως μέθοδος διακριτοποίησης (binning). Στον πίνακα που ακολουθεί παρακάτω παρουσιάζονται οι κατηγορίες των χαρακτηριστικών στις οποίες εφαρμόστηκε η παραπάνω μέθοδος μαζί με το εύρος τιμών που ανήκει σε κάθε έναν από τους κουβάδες που δημιουργήθηκαν.

Κατηγορίες / **Bins**	**Κόστος ανά Βράδυ**	**Βαθμολογία κριτικών**	**Βαθμολογία Τοποθεσίας**
Bin no.1	0€ - 99€	0 - 29	0 - 2
Bin no.2	100€ - 199€	30 - 49	3 - 4
Bin no.3	200€ - 299€	50 - 69	5 - 6
Bin no.4	300€ - 399€	70 - 79	7
Bin no.5	400€ - 499€	80 - 100	8
Bin no.6	700€ - 999€	-	9 - 10
Bin no.7	1000€ +	-	-

Μετά την αντιστοίχιση των δεδομένων στον κατάλληλο κάδο ή κυριαρχία ενός αντικειμένου ως προς ένα άλλο είχε πέρα από μαθηματική και ουσιαστική σημασία. Έπειτα από την εκτέλεση του skyline αλγορίθμου επιστράφηκαν 3669 δεδομένα. Το skyline σύνολο αποτελεί το 40% των αρχικών δεδομένων κάτι που είναι αναμενόμενο και λογικό σύμφωνα με την κατάρα των πολλών διαστάσεων που περιγράφηκε στο βιβλιογραφικό μέρος της παρούσας διπλωματικής.

Σύστημα Προτιμήσεων

Στο σύστημα κάθε διαμέρισμα αναπαρίσταται ως ένας μονοδιάστατος πίνακας με τόσες θέσεις όσα τα χαρακτηριστικά που μας ενδιαφέρουν προκειμένου να πραγματοποιηθούν οι προτάσεις, δηλαδή 11. Ο πίνακας είναι δυαδικός με μηδενικό να σημαίνει απουσία του συγκεκριμένου χαρακτηριστικού και άσσο παρουσία. Κάθε χρήστης έχει τον δικό του πίνακα αρχικοποιημένο στο μηδέν. Επιλέγοντας το ιδανικό διαμέρισμα για αυτόν, ο χρήστης δηλώνει τα χαρακτηριστικά που τον ενδιαφέρουν περισσότερο τα οποία αποθηκεύονται στον πίνακα 11 θέσεων που του αντιστοιχεί. Το σύστημα προτιμήσεων χρησιμοποιώντας τη συνάρτηση ομοιότητας συνημίτονου παρέχει προτάσεις για δύο διαφορετικές ομάδες. Στην μια χρησιμοποιεί το skyline σύνολο της βάσης δεδομένων που επιλέχθηκε, ενώ στη δεύτερη χρησιμοποιεί ολόκληρη τη βάση δεδομένων. Υπολογίζονται τα πέντε πλησιέστερα δωμάτια σε σχέση με το δωμάτιο που επέλεξε ως ιδανικό για κάθε ομάδα, τα οποία και εμφανίζονται στον χρήστη ως προτεινόμενα.

3.3 Πειραματική Διαδικασία

Προκειμένου να εξετάσουμε κατά πόσο επηρεάζει το skyline σύνολο την ικανοποίηση του χρήστη σε ένα σύστημα προτιμήσεων κατασκευάσαμε μια εκδοχή του συστήματος προτιμήσεων στην οποία χρησιμοποιούνται τα αρχικά δεδομένα και μια που βασίζεται στην αρχιτεκτονική μας, χρησιμοποιώντας τα δεδομένα του skyline συνόλου.

3.3.1 Πειραματικός Σχεδιασμός

Είναι απαραίτητο για το πείραμα μας, να ορίσουμε τις ανεξάρτητες και τις εξαρτημένες μεταβλητές του. Ως ανεξάρτητες μεταβλητές ορίζουμε την εμπειρία του χρήστη σε σχέση με συστήματα προτιμήσεων γενικότερα, καθώς και τα δωμάτια που θα προταθούν από τα δύο συστήματα προτιμήσεων που δημιουργήθηκαν. Μπορούμε να γνωρίζουμε το επίπεδο εξοικείωσης του χρήστη με παρόμοια συστήματα μέσα από την πρώτη ερώτηση του ερωτηματολογίου. Οι εξαρτημένες μεταβλητές του πειράματος είναι το επίπεδο ικανοποίησης του χρήστη από το κάθε σύστημα προτιμήσεων, η επιλογή συστήματος, καθώς και μια σειρά από μεταβλητές που αξιολογούν τον φόρτο εργασίας κατά την πραγματοποίηση του πειράματος. Οι μεταβλητές αυτές ορίζονται από το επίσημο NASA Task Load Index [40]. Τόσο οι εξαρτημένες όσο και οι ανεξάρτητες μεταβλητές αξιολογούνται μέσα από το ερωτηματολόγιο που καλείται να συμπληρώσει ο εκάστοτε χρήστης στο τέλος της πειραματικής διαδικασίας.

Η μελέτη που πραγματοποιούμε ανήκει στη κατηγορία Single Subject Study καθώς έχουμε μια ομάδα χρηστών στην οποία εμφανίζουμε αποτελέσματα και από τα δύο συστήματα προτιμήσεων ζητώντας από αυτούς να αξιολογήσουν ποιο από τα δύο παρέχει πιο ικανοποιητικά αποτελέσματα. Ξεκινώντας από τη μηδενική υπόθεση H_0 = 'Η χρήση του skyline συνόλου στα συστήματα προτιμήσεων δεν επιφέρει πιο ικανοποιητικά αποτελέσματα' θέλουμε να εξετάσουμε την εναλλακτική υπόθεση H_1 = 'Η χρήση του skyline συνόλου στα συστήματα προτιμήσεων επιφέρει πιο ικανοποιητικά αποτελέσματα'.

3.3.2 Μελέτη Περίπτωσης

Κάθε χρήστης καλείται να επιλέξει το ιδανικό διαμέρισμα που θα επιθυμούσε να ενοικιάσει. Για να το πραγματοποιήσει αυτό, χρησιμοποιεί μια σειρά από φίλτρα στα χαρακτηριστικά όπως 'κόστος ανά βράδυ', 'άτομα', 'βαθμολογία τοποθεσίας' και 'βαθμολογία κριτικών' περιορίζοντας έτσι τα αποτελέσματα που προκύπτουν από την αναζήτηση του. Επιλέγει επιπλέον εάν το διαμέρισμα επιθυμεί να έχει προσβασιμότητα αναπηρικών αμαξιδίων. Η επιλογή του διαμερίσματος είναι και αυτή που ορίζει τα επιθυμητά χαρακτηριστικά του χρήστη και πάνω σε αυτά βασίζονται τα δύο συστήματα προτιμήσεων για να παρουσιάσουν τις προτάσεις τους. Τα δύο συστήματα προτιμήσεων παρουσιάζουν συνολικά πέντε προτεινόμενα διαμερίσματα το κάθε ένα όπως περιεγράφηκε στην προηγούμενη ενότητα. Η διαφορά είναι ότι το πρώτο κάνει χρήση του skyline συνόλου, ενώ το δεύτερο χρησιμοποιεί το αρχικό σύνολο δεδομένων χωρίς τη χρήση skyline αλγόριθμου. Και τα δύο συστήματα προτιμήσεων βασίζονται στο φιλτράρισμα με βάση το περιεχόμενο που αναλύθηκε στην ενότητα 2. Στον χρήστη παρουσιάζονται ως 'Recommender 1' και 'Recommender 2', έτσι ώστε να μην γνωρίζει τα χαρακτηριστικά κάθε συστήματος. Η διαδικασία επιλογής ιδανικού διαμερίσματος για την εμφάνιση προτεινόμενων από τα δύο συστήματα προτιμήσεων μπορεί να επαναληφθεί από τον χρήστη όσες φορές επιθυμεί. Με την ολοκλήρωση αυτής της διαδικασίας, καλείται να συμπληρώσει μια σειρά από ερωτήσεις σχετικά με τα δυο συστήματα καθώς και με την συνολική διαδικασία. Ο κώδικας για τον σχεδιασμό της ιστοσελίδας βρίσκεται στο [41] ενώ οι ερωτήσεις του ερωτηματολογίου ακολουθούν στις παρακάτω εικόνες.

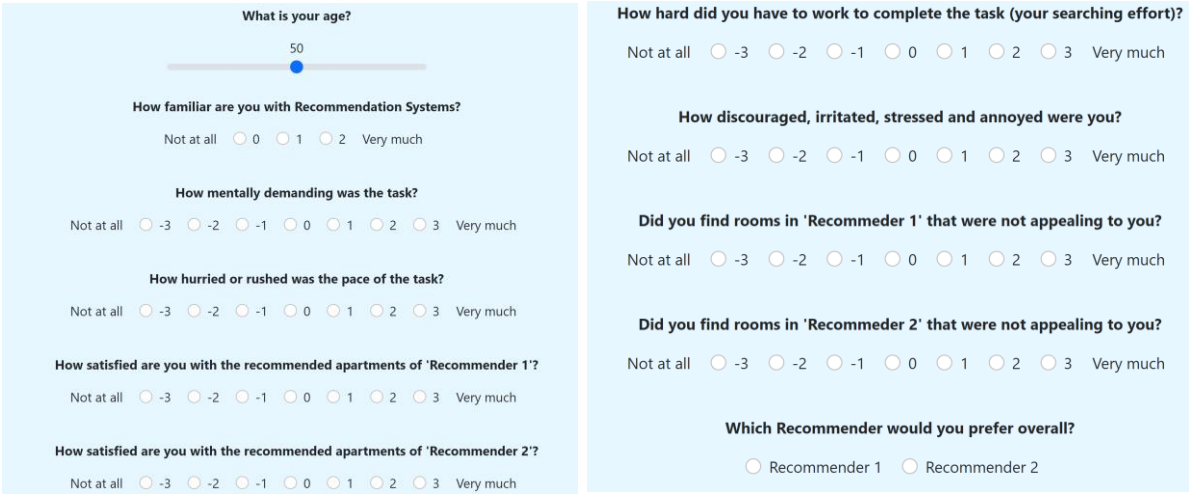


Figure 6 Ερωτηματολόγιο

3.3.3 Πρότυπα Ιστοσελίδας

Παρακάτω ακολουθούν τα πρότυπα (templates) της ιστοσελίδας που δημιουργήθηκε για τους σκοπούς της πειραματικής διαδικασίας. Η πρώτη σελίδα ορίζει το έργο που πρέπει να ολοκληρώσει ο χρήστης και στη συνέχεια περιγράφει τη διαδικασία που πρέπει να ακολουθήσει στις υπόλοιπες σελίδες προκειμένου να επιτευχθεί ο στόχος του.

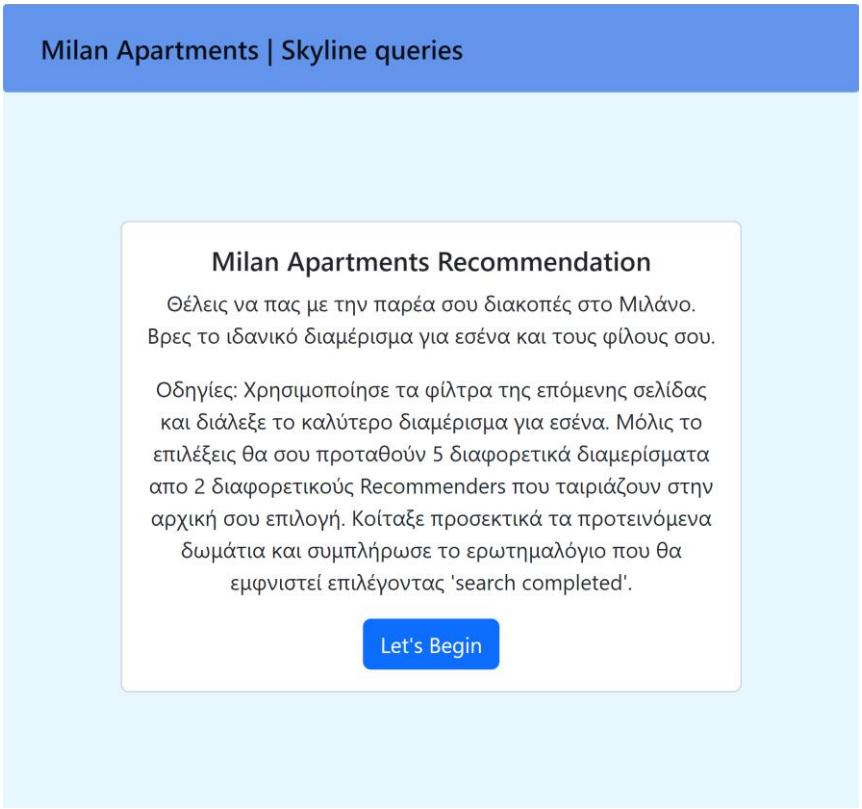
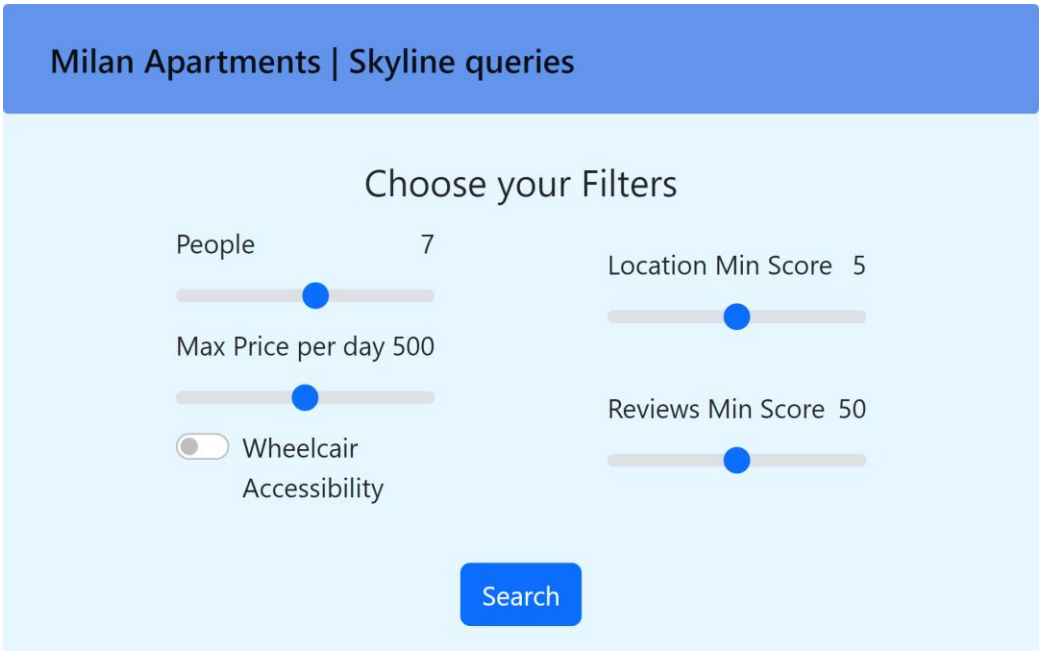


Figure 7 1ο Πρότυπο Ιστοσελίδας

Στη συνέχεια επιλέγοντας ‘Let’s Begin’ μεταφέρεται στην επόμενη σελίδα στην οποία ρυθμίζοντας κατάλληλα τα διάφορα φίλτρα και επιλέγοντας ‘Search’, εμφανίζονται τα αντίστοιχα δωμάτια.



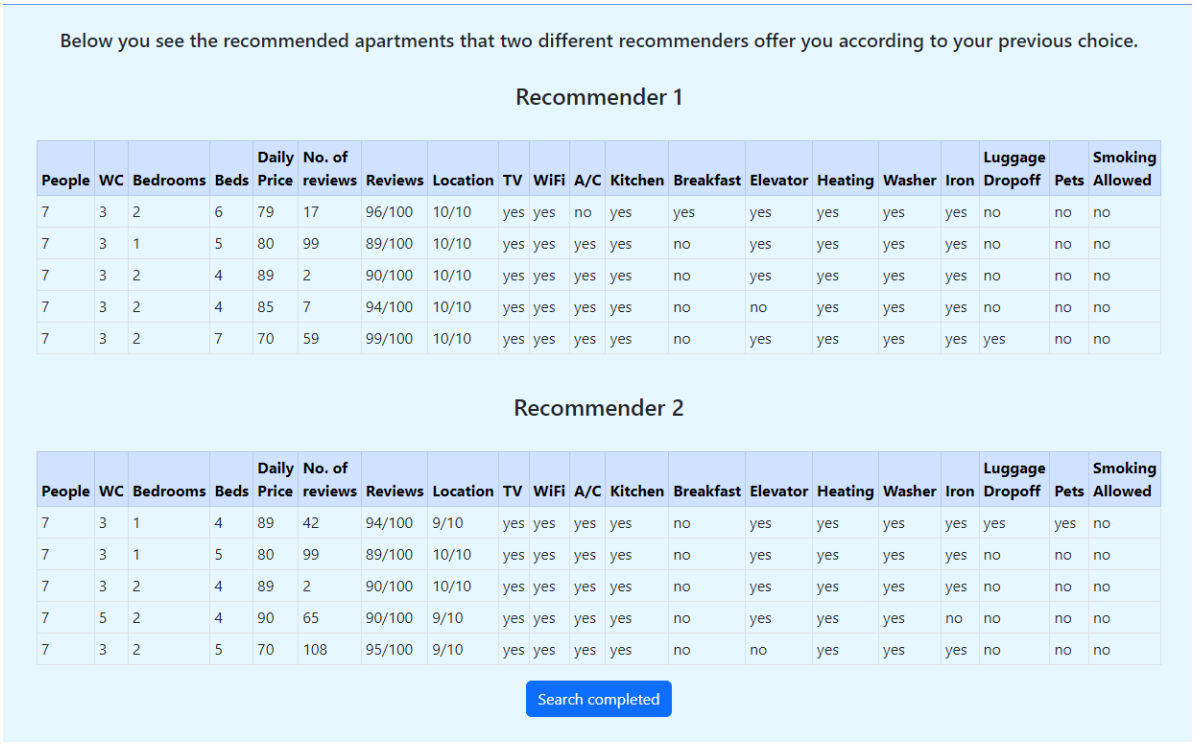
Previous Next From: 0 To: 10

People	WC	Bedrooms	Beds	Daily Price	No. of reviews	Reviews	Location	TV	WiFi	A/C	Kitchen	Breakfast	Elevator	Heating	Washer	Iron	Luggage Dropoff	Pets	Smoking Allowed	Select
7	3	1	5	80	99	89/100	10/10	yes	yes	yes	yes	no	yes	yes	yes	yes	no	no	no	Select
7	3	2	4	95	103	97/100	10/10	yes	yes	no	yes	no	yes	yes	yes	yes	no	yes	no	Select
7	9	3	7	87	121	100/100	10/10	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	no	no	Select
7	3	2	4	85	15	92/100	10/10	yes	yes	yes	yes	no	yes	yes	no	no	no	no	no	Select
7	5	2	5	100	182	98/100	10/10	yes	yes	yes	yes	no	yes	yes	yes	yes	yes	no	no	Select
7	3	2	6	79	17	96/100	10/10	yes	yes	no	yes	yes	yes	yes	yes	yes	no	no	no	Select
7	5	3	6	50	13	95/100	10/10	yes	yes	no	yes	no	yes	yes	yes	yes	no	yes	yes	Select
7	3	3	4	100	86	98/100	10/10	no	yes	yes	yes	no	no	yes	yes	yes	yes	yes	no	Select
7	3	2	6	92	145	99/100	10/10	yes	yes	no	yes	no	yes	yes	yes	yes	yes	yes	yes	Select
7	3	2	4	99	40	97/100	10/10	yes	yes	yes	yes	no	yes	yes	yes	yes	no	no	no	Select

Figure 8 2ο Πρότυπο Ιστοσελίδας

Παρατηρώντας τα διάφορα χαρακτηριστικά των διαμερισμάτων με την εφαρμογή των φίλτρων, ο χρήστης επιλέγει το ιδανικό σύμφωνα με τις προτιμήσεις του επιλέγοντας το ‘Select’ που αντιστοιχεί σε αυτό. Με βάση τα χαρακτηριστικά του επιλεγμένου διαμερίσματος τα δύο συστήματα προτιμήσεων χρησιμοποιούν τις βάσεις δεδομένων που

τους έχουν ανατεθεί και στην επόμενη σελίδα εμφανίζονται οι προτάσεις τους όπως φαίνεται στην παρακάτω εικόνα.



Below you see the recommended apartments that two different recommenders offer you according to your previous choice.

Recommender 1

People	WC	Bedrooms	Beds	Daily Price	No. of reviews	Reviews	Location	TV	WiFi	A/C	Kitchen	Breakfast	Elevator	Heating	Washer	Iron	Luggage Dropoff	Pets	Smoking Allowed
7	3	2	6	79	17	96/100	10/10	yes	yes	no	yes	yes	yes	yes	yes	yes	no	no	no
7	3	1	5	80	99	89/100	10/10	yes	yes	yes	yes	no	yes	yes	yes	yes	no	no	no
7	3	2	4	89	2	90/100	10/10	yes	yes	yes	yes	no	yes	yes	yes	yes	no	no	no
7	3	2	4	85	7	94/100	10/10	yes	yes	yes	yes	no	no	yes	yes	yes	no	no	no
7	3	2	7	70	59	99/100	10/10	yes	yes	yes	yes	no	yes	yes	yes	yes	yes	no	no

Recommender 2

People	WC	Bedrooms	Beds	Daily Price	No. of reviews	Reviews	Location	TV	WiFi	A/C	Kitchen	Breakfast	Elevator	Heating	Washer	Iron	Luggage Dropoff	Pets	Smoking Allowed
7	3	1	4	89	42	94/100	9/10	yes	yes	yes	yes	no	yes	yes	yes	yes	yes	yes	no
7	3	1	5	80	99	89/100	10/10	yes	yes	yes	yes	no	yes	yes	yes	yes	no	no	no
7	3	2	4	89	2	90/100	10/10	yes	yes	yes	yes	no	yes	yes	yes	yes	no	no	no
7	5	2	4	90	65	90/100	9/10	yes	yes	yes	yes	no	yes	yes	yes	no	no	no	no
7	3	2	5	70	108	95/100	9/10	yes	yes	yes	yes	no	no	yes	yes	yes	no	no	no

Search completed

Figure 9 3ο Πρότυπο Ιστοσελίδας

Η παραπάνω διαδικασία επαναλαμβάνεται όσες φορές επιθυμεί ο χρήστης και επιλέγοντας ‘Search Completed’ μεταφέρεται στο ερωτηματολόγιο της εικόνας Figure 6 Ερωτηματολόγιο] το οποίο ολοκληρώνεται μόλις απαντήσει σε όλες τις ερωτήσεις και επιλέξει ‘Submit’.

4 Αποτελέσματα

Συνολικά το ερωτηματολόγιο συμπληρώθηκε από 33 άτομα με εύρος ηλικίας 20-62 έτη. Οι απαντήσεις του ερωτηματολογίου έδειξαν ότι οι χρήστες βαθμολόγησαν το σύστημα προτιμήσεων που έκανε χρήση του skyline συνόλου υψηλότερα σε σχέση με το σύστημα προτιμήσεων που χρησιμοποιούσε ολόκληρο το σύνολο των δεδομένων ως προς τον βαθμό ικανοποίησης τους σχετικά με τα προτεινόμενα αποτελέσματα του καθενός. Πιο συγκεκριμένα σε κλίμακα από το 0 έως το 7 βαθμολόγησαν τον skyline recommender με 6.42 ενώ τον απλό recommender με 5.42. Επιπλέον όταν ερωτήθηκαν κατά πόσο το κάθε σύστημα είχε αποτελέσματα που δεν ήταν ικανοποιητικά δήλωσαν ότι ο αριθμός μη ικανοποιητικών αποτελεσμάτων ήταν μεγαλύτερος στο σύστημα προτιμήσεων που δεν έκανε χρήστη του

skyline συνόλου συγκεντρώνοντας μέση βαθμολογία 3.9/7 σε αντίθεση με τον skyline recommender που βαθμολογήθηκε με 2.84/7. Τα παραπάνω αποτελέσματα εμφανίζονται και στα διαγράμματα της εικόνας που ακολουθεί. Στη συνέχεια με βάση την επιλογή συστήματος προτιμήσεων οι χρήστες κατηγοριοποιούνται σε δύο ομάδες και εξετάζονται οι υπόλοιπες εξαρτημένες μεταβλητές του πειράματος.

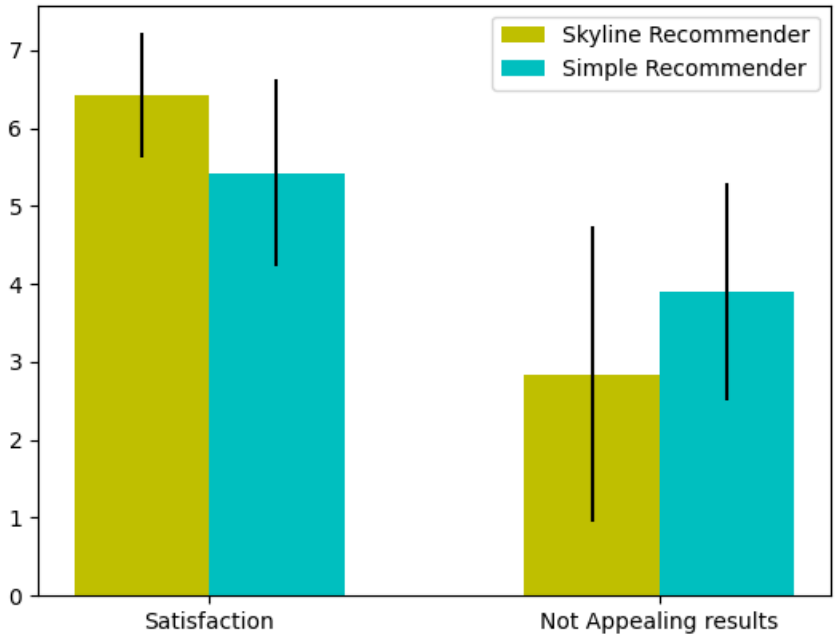


Figure 10 Ικανοποίηση χρηστών

Στην ερώτηση 'Ποιο σύστημα συστάσεων θα προτιμούσατε συνολικά', οι 10 χρήστες από τους 33 επέλεξαν το απλό σύστημα προτιμήσεων που δεν έκανε χρήση του skyline συνόλου και οι υπόλοιποι 23 επέλεξαν το σύστημα προτιμήσεων που χρησιμοποιούσε το skyline σύνολο. Θέλαμε επιπλέον να εξετάσουμε εάν υπήρχε κάποια συσχέτιση ανάμεσα στην εξοικείωση των χρηστών με τα συστήματα προτιμήσεων και την επιλογή συστήματος προτιμήσεων που έκαναν. Από το διάγραμμα που προέκυψε, διαπιστώνουμε ότι δεν προκύπτει κάτι τέτοιο καθώς ο μέσος όρος εξοικείωσης και για τις δυο ομάδες χρηστών είναι στα ίδια επίπεδα. Στις δύο εικόνες που ακολουθούν παρουσιάζονται τα αποτελέσματα που αναφέρθηκαν.

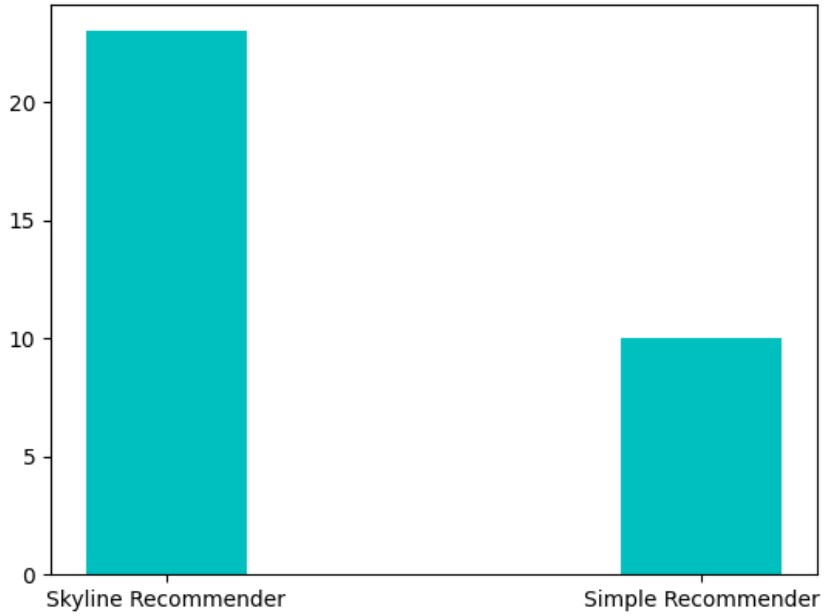


Figure 11 Προτίμηση συστήματος προτιμήσεων

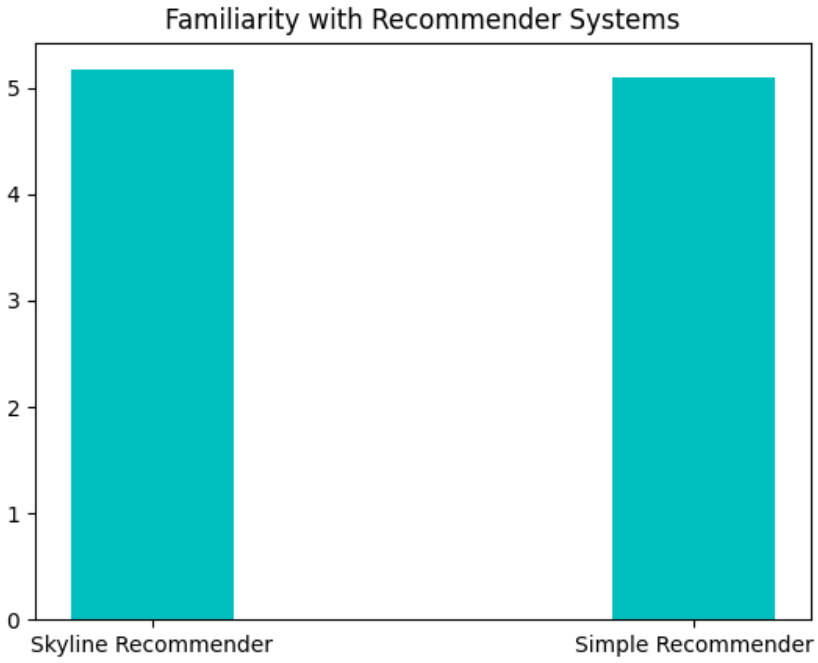


Figure 12 Εξοικείωση χρηστών

Τέλος παρουσιάζονται τα αποτελέσματα σχετικά με τις ανεξάρτητες μεταβλητές που αφορούν την αποθάρρυνση των χρηστών κατά την πειραματική διαδικασία, την δυσφορία τους, καθώς και τη προσπάθεια που κατέλαβαν στην αναζήτηση τους, ανάλογα με το σύστημα προτιμήσεων που επέλεξαν. Παρατηρούμε ότι οι χρήστες που επέλεξαν το απλό

σύστημα προτιμήσεων αντιμετώπισαν μεγαλύτερη δυσκολία ως προς την αναζήτηση διαμερίσματος σε σχέση με αυτούς που επέλεξαν το σύστημα προτιμήσεων με χρήση skyline συνόλου. Οι υπόλοιπες μεταβλητές όμως βλέπουμε ότι δεν έχουν κάποια συσχέτιση με αυτή την επιλογή καθώς οι τιμές τους κυμαίνονται στα ίδια επίπεδα και για τις δυο ομάδες.

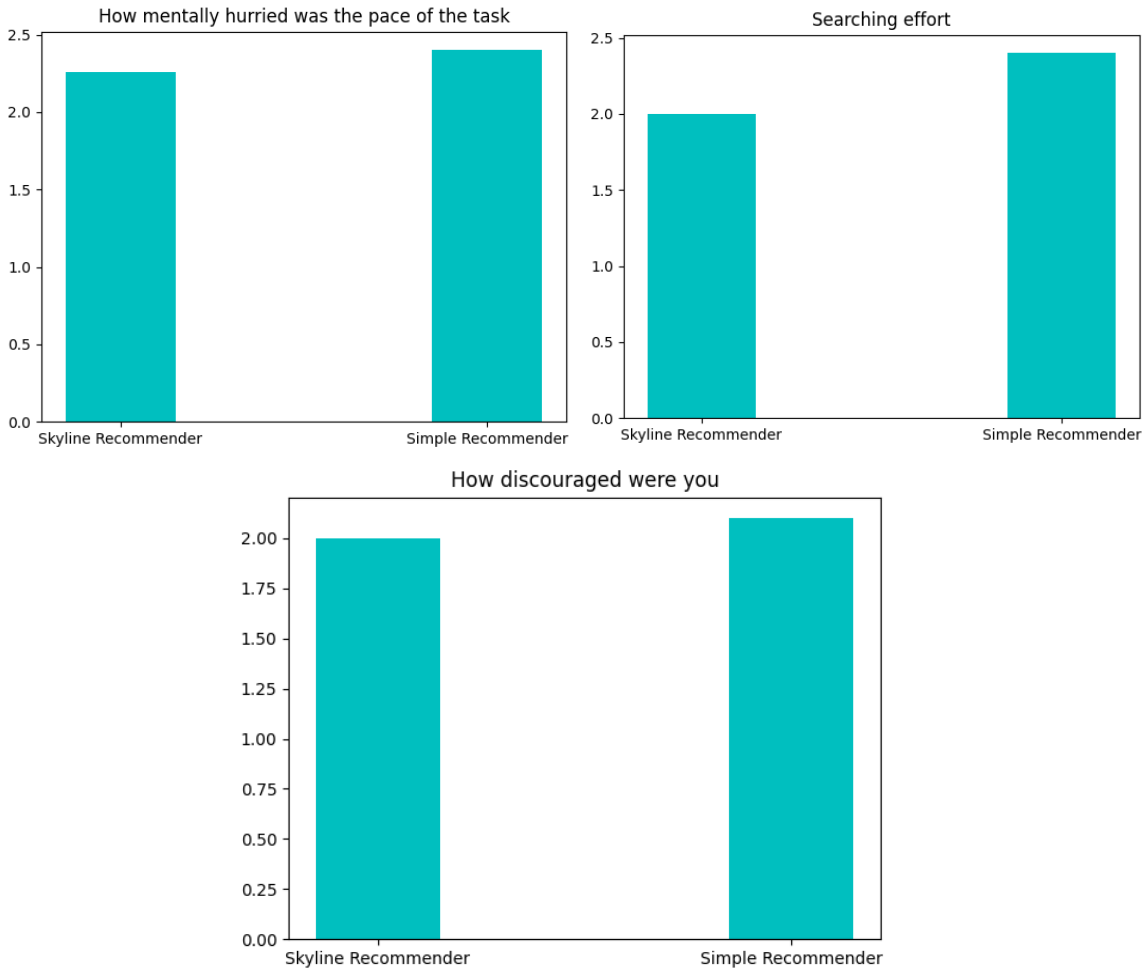


Figure 13 Αποτελέσματα ερωτηματολογίου για τις εξαρτημένες μεταβλητές

Όπως τονίσαμε και στην στον πειραματικό σχεδιασμό, στόχος μας είναι να ελέγξουμε εάν ισχύει η εναλλακτική υπόθεση της προηγούμενης ενότητας ή όχι. Χρησιμοποιούμε λοιπόν τα δεδομένα που προέκυψαν σχετικά με την ικανοποίηση των χρηστών για να πραγματοποιήσουμε τεστ σημαντικότητας (test of significance). Η τιμή που προκύπτει από το τεστ σημαντικότητας είναι ευρέως χρησιμοποιούμενη στον στατιστικό έλεγχο υποθέσεων, για αυτό τον λόγο τη χρησιμοποιούμε και στη συγκεκριμένη πειραματική διαδικασία. Το τεστ που επιλέξαμε να χρησιμοποιήσουμε προκειμένου να συγκρίνουμε τις μέσες τιμές που προέκυψαν για τον βαθμό ικανοποίησης των χρηστών από τα αποτελέσματα που προέκυψαν στις δύο περιπτώσεις που εξετάσαμε ονομάζεται t-test. Για να γίνει χρήση του συγκεκριμένου

τεστ πρέπει τα δείγματά μας να είναι ανεξάρτητα μεταξύ τους, μια συνθήκη που ικανοποιείται από τα δικά μας δεδομένα. Επιπλέον, για πολύ μικρό δείγμα δεδομένων (N<20) πρέπει να ακολουθείται κανονική κατανομή. Για μεγαλύτερο αριθμό δειγμάτων όμως, το t-test δεν επηρεάζεται από την κατανομή των δεδομένων και καθώς στη συγκεκριμένη περίπτωση έχουμε N = 33, μπορούμε να κάνουμε χρήση του συγκεκριμένου τεστ.

Έστω X_1 και X_2 οι μέσες τιμές του βαθμού ικανοποίησης των χρηστών από το σύστημα προτιμήσεων με χρήση skyline συνόλου και του απλού συστήματος προτιμήσεων αντίστοιχα. Έστω επίσης N_1 και N_2 το μέγεθος των δύο δειγμάτων και s_1, s_2 η τυπική απόκλιση των δύο δειγμάτων αντίστοιχα. Τότε η μεταβλητή t υπολογίζεται σύμφωνα με τον τύπο:

$$t = \frac{X_1 - X_2}{\sqrt{\left(\frac{(N_1 - 1) * s_1^2 + (N_2 - 1) * s_2^2}{N_1 + N_2 - 2}\right) * \left(\frac{1}{N_1} + \frac{1}{N_2}\right)}}$$

Εφαρμόζοντας το t-test με είσοδο τις δύο μέσες τιμές ικανοποίησης και επίπεδο σημαντικότητας (significance level) στο 0.05, προκύπτει ότι το αποτέλεσμα είναι σημαντικό και άρα υπάρχει πράγματι διαφοροποίηση ανάμεσα στα δύο συστήματα προτιμήσεων, επιβεβαιώνοντας την εναλλακτική μας υπόθεση ότι η χρήση του skyline συνόλου στα συστήματα προτιμήσεων επιφέρει πιο ικανοποιητικά αποτελέσματα.

5 Συζήτηση – Συμπεράσματα

Με τη συνεχή αύξηση της πληροφορίας σε όλους τους τομείς η αναζήτηση πληροφοριών και αντικειμένων γίνεται όλο και πιο απαιτητική διαδικασία. Είναι σημαντικό τα δεδομένα να φιλτράρονται και να κατηγοριοποιούνται κατάλληλα, να είναι εξατομικευμένα στον κάθε χρήστη, διευκολύνοντας την αναζήτηση του. Τα συστήματα προτιμήσεων συμβάλλουν σημαντικά στη διαχείριση και τον έλεγχο της συνεχώς αυξανόμενης πληροφορίας, εμφανίζοντας στον χρήστη αποτελέσματα που είναι σχετικά με τις προτιμήσεις του χρησιμοποιώντας μια σειρά από διαφορετικές τεχνικές και προσεγγίσεις. Παρ' όλα αυτά ο όγκος των δεδομένων οδηγεί ακόμα και τα συστήματα προτιμήσεων σε προτάσεις που ενδέχεται να μην είναι οι πιο ικανοποιητικές για ένα χρήστη. Γίνεται λοιπόν επιτακτική η ανάγκη εύρεσης μεθόδων βελτιστοποίησης αυτών των προτάσεων.

Οι skyline αλγόριθμοι συμβάλλουν στην καλύτερη διαχείριση και επεξεργασία της πληροφορίας επιτυγχάνοντας τη διάκριση των καλύτερων αντικειμένων σε ένα σύνολο, εξετάζοντας και συγκρίνοντας ορισμένα από τα χαρακτηριστικά τους. Τα αντικείμενα που διακρίνουν ως ιδανικότερα συντελούν το skyline σύνολο. Ο υπολογισμός του skyline συνόλου βασίζεται σε γενικευμένες αποδοχές προτιμήσεων καθώς δεν απαιτεί τον προσδιορισμό προτιμήσεων από τον εκάστοτε χρήστη. Το skyline σύνολο περιέχει τα κυρίαρχα αντικείμενα μιας βάσης δεδομένων, αποτελείται δηλαδή από όλα τα αντικείμενα ενός συνόλου που δεν κυριαρχούνται από κανένα άλλο αντικείμενο σύμφωνα με την Pareto έννοια κυριαρχίας.

Στόχος της παρούσας διπλωματικής εργασίας είναι να εξετάσει κατά πόσο ή χρήση του skyline συνόλου σε ένα σύστημα προτιμήσεων επιδρά στα επίπεδα ικανοποίησης των χρηστών ως προς το προτεινόμενο περιεχόμενο τους. Για τον σκοπό αυτό σχεδιάστηκε ένα σύστημα προτιμήσεων που βασίζεται στο περιεχόμενο και χρησιμοποιεί τη μετρική ομοιότητας συνημίτονου, το οποίο στη πρώτη περίπτωση πρότεινε δεδομένα από το skyline σύνολο μιας βάσης δεδομένων και στη δεύτερη χρησιμοποιούσε τη βάση δεδομένων χωρίς κάποια περαιτέρω επεξεργασία. Η βάση που χρησιμοποιήθηκε αποτελείται από δεδομένα για ενοικιαζόμενα διαμερίσματα και περιέχει μόνο αριθμητικά δεδομένα. Ο αλγόριθμος που επιλέχθηκε για τον υπολογισμό του skyline συνόλου είναι ο Block Nested Loop [4] καθώς είναι από τους απλούστερους αλγόριθμους στη βιβλιογραφία και είναι ιδιαίτερα αποτελεσματικός στη περίπτωση μας όπου το σύνολο των δεδομένων δεν είναι πολύ μεγάλο.

Καθώς ο κάθε χρήστης που συμμετείχε στην έρευνα δεν αλληλεπιδρούσε με τους υπόλοιπους τα προτεινόμενα διαμερίσματα σε αυτών θα έπρεπε να γίνουν με βάση τις δικές του προτιμήσεις αποκλειστικά. Για αυτό τον λόγο το κατασκευάστηκε ένα σύστημα προτιμήσεων που βασίζεται στο περιεχόμενο.

Όπως έδειξε το πείραμα που πραγματοποιήθηκε στη συγκεκριμένη διπλωματική εργασία, η εφαρμογή skyline αλγορίθμου και η χρήση του skyline συνόλου στη θέση ολόκληρης της βάσης δεδομένων σε ένα σύστημα προτιμήσεων, αυξάνει την ικανοποίηση του χρήστη σημαντικά, παρέχοντας του τις καλύτερες δυνατές προτάσεις σύμφωνα με τις προτιμήσεις του. Αυτό συμβαίνει καθώς με την επιλογή ενός ιδανικού διαμερίσματος από τον χρήστη, το σύστημα προτιμήσεων που κάνει χρήση του skyline συνόλου θα αναζητήσει κατάλληλα διαμερίσματα για να του προτείνει μέσα από ένα σύνολο με ασύγκριτα αντικείμενα. Συνεπώς κάθε πρόταση που θα προκύψει διαφέρει ουσιαστικά σε σχέση με τις υπόλοιπες και έτσι όλες οι προτάσεις που γίνονται από το σύστημα προτιμήσεων έχουν να προσφέρουν κάτι διαφορετικό. Αντίθετα, η χρήση ολόκληρου του συνόλου δεδομένων από το σύστημα προτιμήσεων ενώ οδηγεί στη πρόταση διαμερισμάτων που παρέχουν όλες τις παροχές που επιθυμεί ο χρήστης, δεν αποκλείει τα διαμερίσματα που για τις ίδιες παροχές κυριαρχούνται από κάποιο άλλο αντικείμενο σε βασικά χαρακτηριστικά όπως είναι η τιμή ή η συνολική βαθμολογία που συγκεντρώνει ένα διαμέρισμα, με αποτέλεσμα τελικά, στο σύνολο των προτάσεων που γίνονται να συμπεριλαμβάνονται και δωμάτια που δεν προσφέρουν κάτι διαφορετικό στον χρήστη και συνεπώς δεν έχουν λόγο να αποτελούν πρόταση του συστήματος. Αυτό επιβεβαιώθηκε και μέσα από το ερωτηματολόγιο που κλήθηκαν οι χρήστες να συμπληρώσουν στο τέλος της πειραματικής διαδικασίας, από το γεγονός ότι το μεγαλύτερο ποσοστό των χρηστών δήλωσαν πως το σύστημα προτιμήσεων που δεν έκανε χρήση του skyline συνόλου είχε περισσότερα αποτελέσματα που δεν θεώρησαν ικανοποιητικά.

Όρια και περιορισμοί

Στα συμπεράσματα που παρουσιάστηκαν παραπάνω, πρέπει να ληφθούν υπ’ όψη τα όρια και οι περιορισμοί της παρούσας έρευνας. Τα αποτελέσματα της πειραματικής διαδικασίας προήλθαν από ένα σχετικά μικρό δείγμα συμμετεχόντων (33 άτομα συνολικά). Μεγαλύτερο δείγμα θα είχε ενδεχομένως και σαφέστερα αποτελέσματα σχετικά με τα επίπεδα ικανοποίησης των χρηστών από τα δύο συστήματα προτιμήσεων. Επιπλέον, είναι σημαντικό να αναφερθούμε και στα διαμερίσματα που συντέλεσαν το skyline σύνολο, καθώς

καθοριστικό ρόλο είχε η μέθοδος της διακριτοποίησης που πραγματοποιήθηκε. Διαφορετικά εύρη τιμών στους κάδους που δημιουργήθηκαν, θα οδηγούσαν σε διαφορετική κατάταξη των δεδομένων, αλλάζοντας συνεπώς σε κάποιο βαθμό και τα στοιχεία του skyline συνόλου. Αυτή η αλλαγή ενδέχεται να επηρέαζε τα προτεινόμενα διαμερίσματα από τα συστήματα προτιμήσεων και συνεπώς την επιλογή των χρηστών.

Τέλος, σημαντικό είναι να αναφερθεί ότι τα χαρακτηριστικά που χρησιμοποιούσε το σύστημα προτιμήσεων προκειμένου να παρουσιάσει στον χρήστη το σύνολο των προτεινόμενων δωματίων, είχαν προκύψει από την επιλογή ενός μόνο ιδανικού δωματίου από αυτόν στην αρχή της πειραματικής διαδικασίας με αποτέλεσμα τα χαρακτηριστικά στα οποία βασιζόταν το σύστημα να μην αντιπροσωπεύουν στο μέγιστο τον χρήστη. Η επιλογή περισσότερων ιδανικών δωματίων μπορεί να οδηγούσε σε μεγαλύτερη ακρίβεια στη καταγραφή των προτιμήσεων του, δημιουργώντας έτσι ένα προφίλ χρήστη που τον αντικατοπτρίζει καλύτερα και συνεπώς θα οδηγήσει σε ακόμα πιο ικανοποιητικές προτάσεις.

Μελλοντικές Επεκτάσεις

Η συγκεκριμένη έρευνα θα μπορούσε να επεκταθεί μελλοντικά διαφοροποιώντας την τεχνική που χρησιμοποιήθηκε προκειμένου να δημιουργηθεί το προφίλ του χρήστη με βάση το οποίο το σύστημα προτιμήσεων συγκέντρωσε το σύνολο των αντικειμένων που πρότεινε και στις δύο περιπτώσεις. Κατά τη διαδικασία επιλογής του ιδανικού διαμερίσματος στη πειραματική διαδικασία, ο χρήστης δεν γνώριζε ότι συλλέγονται συγκεκριμένα χαρακτηριστικά του δωματίου που επέλεξε. Συνεπώς, μπορεί να αγνόησε κάποιο από τα χαρακτηριστικά που υπήρχε στο διαμέρισμα που επέλεξε ή αντίστοιχα να απουσίαζε από αυτό, με αποτέλεσμα να υπάρχουν προτάσεις στη συνέχεια που δεν τον ικανοποιούν. Εναλλακτικά, θα είχε νόημα να πριν την επιλογή ιδανικού διαμερίσματος για τον υπολογισμό του εύρους τιμής και βαθμολογίας που αναζητά, να υπήρχε ένα στάδιο σκιαγράφησης του προφίλ του χρήστη στο οποίο θα ζητείται να επιλέξει με πιο άμεσο τρόπο τα ιδανικά χαρακτηριστικά ενός διαμερίσματος παρέχοντας έτσι μεγαλύτερη ακρίβεια στις προτάσεις του συστήματος προτιμήσεων.

Μια ακόμα σημαντική επέκταση μελλοντικά θα ήταν η πραγματοποίηση του ίδιου πειράματος για μια σειρά από διαφορετικές κατηγορίες συστημάτων προτιμήσεων. Ιδιαίτερο ενδιαφέρον θα είχε για παράδειγμα η ενσωμάτωση του skyline συνόλου σε ένα σύστημα προτιμήσεων που βασίζεται στο συνεργατικό φιλτράρισμα, καθώς σε αυτή τη περίπτωση οι

προτάσεις του ενός χρήστη θα βασίζονταν στη προτίμηση skyline αντικειμένων άλλων χρηστών. Έτσι θα έχουμε μια συνολική εικόνα ως προς τη συμβολή της χρήσης του skyline συνόλου στα επίπεδα ικανοποίησης ενός χρήστη από τις προτάσεις ενός συστήματος προτιμήσεων.

Τέλος, σύμφωνα και με το βιβλιογραφικό σκέλος της παρούσας διπλωματικής εργασίας, υπάρχουν πολλές διαφορετικές προσεγγίσεις όσο αφορά τον τρόπο ένταξης των skyline αλγορίθμων και συνόλων στα συστήματα προτιμήσεων. Μελλοντικά θα ήταν πολύ σημαντική και η διερεύνηση διαφορετικών μεθόδων χρήσης του skyline συνόλου στα συγκεκριμένα συστήματα με στόχο την βέλτιστη διαχείριση της πληροφορίας και την αύξηση της ικανοποίησης των χρηστών από τα προτεινόμενα αποτελέσματα.

Βιβλιογραφία

[1] Touloumis, K. (2019). Choosing from skyline sets..

[2] Tiakas, E., Papadopoulos, A. N., & Manolopoulos, Y. (2015, July). Skyline queries: An introduction. In *2015 6th International Conference on Information, Intelligence, Systems and Applications (IISA)* (pp. 1-6). IEEE.

[3] Peng, Y. W., & Chen, W. M. (2019). Parallel k-dominant skyline queries in high-dimensional datasets. *Information Sciences*, *496*, 538-552.

[4] Borzsony, S., Kossmann, D., & Stocker, K. (2001, April). The skyline operator. In *Proceedings 17th international conference on data engineering* (pp. 421-430). IEEE.

[5] Chomicki, J., Godfrey, P., Gryz, J., & Liang, D. (2005). Skyline with presorting: Theory and optimizations. In *Intelligent Information Processing and Web Mining* (pp. 595-604). Springer, Berlin, Heidelberg.

[6] Kossmann, D., Ramsak, F., & Rost, S. (2002, January). Shooting stars in the sky: An online algorithm for skyline queries. In *VLDB'02: Proceedings of the 28th International Conference on Very Large Databases* (pp. 275-286). Morgan Kaufmann.

[7] Papadias, D., Tao, Y., Fu, G., & Seeger, B. (2003, June). An optimal and progressive algorithm for skyline queries. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data* (pp. 467-478).

[8] Pei, J., Yuan, Y., Lin, X., Jin, W., Ester, M., Liu, Q., ... & Zhang, Q. (2006). Towards multidimensional subspace skyline analysis. *ACM Transactions on Database Systems (TODS)*, *31*(4), 1335-1381.

[9] Zhang, M., & Alhajj, R. (2010). Skyline queries with constraints: Integrating skyline and traditional query operators. *Data & Knowledge Engineering*, *69*(1), 153-168.

[10] Papadias, D., Tao, Y., Fu, G., & Seeger, B. (2005). Progressive skyline computation in database systems. *ACM Transactions on Database Systems (TODS)*, *30*(1), 41-82.

[11] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994, October). Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work* (pp. 175-186).

[12] Herlocker, J. L., Konstan, J. A., Borchers, A., & Riedl, J. (1999, August). An algorithmic framework for performing collaborative filtering. In *Proceedings of the*

22nd annual international ACM SIGIR conference on Research and development in information retrieval (pp. 230-237).

[13] Breese, J. S., Heckerman, D., & Kadie, C. (2013). Empirical analysis of predictive algorithms for collaborative filtering. *arXiv preprint arXiv:1301.7363*.

[14] Linden, G., Smith, B., & York, J. (2003). Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, *7*(1), 76-80.

[15] Aggarwal, C. C. (2016). Neighborhood-based collaborative filtering. In *Recommender systems* (pp. 29-70). Springer, Cham.

[16] Aggarwal, C. C., Wolf, J. L., Wu, K. L., & Yu, P. S. (1999, August). Horting hatches an egg: A new graph-theoretic approach to collaborative filtering. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 201-212).

[17] Fouss, F., Pirotte, A., Renders, J. M., & Saerens, M. (2007). Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Transactions on knowledge and data engineering*, *19*(3), 355-369.

[18] Billsus, D., & Pazzani, M. J. (1998, July). Learning collaborative information filters. In *Icml* (Vol. 98, pp. 46-54).

[19] Jadhav, S. D., & Channe, H. P. (2016). Efficient recommendation system using decision tree classifier and collaborative filtering. *Int. Res. J. Eng. Technol*, *3*(8), 2113-2118.

[20] Shyu, M. L., Haruechaiyasak, C., Chen, S. C., & Zhao, N. (2005, April). Collaborative filtering by mining association rules from user access sequences. In *International Workshop on Challenges in Web Information Retrieval and Integration* (pp. 128-135). IEEE.

[21] Miyahara, K., & Pazzani, M. J. (2000, August). Collaborative filtering with the simple Bayesian classifier. In *Pacific Rim International conference on artificial intelligence* (pp. 679-689). Springer, Berlin, Heidelberg.

[22] Lops, P., Gemmis, M. D., & Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. *Recommender systems handbook*, 73-105.

[23] Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, *17*(6), 734-749.

[24] Rocchio, J. (1971). Relevance feedback in information retrieval. *The Smart retrieval system-experiments in automatic document processing*, 313-323.

[25] Pazzani, M., & Billsus, D. (1997). Learning and revising user profiles: The identification of interesting web sites. *Machine learning*, *27*(3), 313-331.

[26] Pazzani, M., & Billsus, D. (1997). Learning and revising user profiles: The identification of interesting web sites. *Machine learning*, *27*(3), 313-331.

[27] Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, *12*(4), 331-370.

[28] Miranda, T., Claypool, M., Gokhale, A., Mir, T., Murnikov, P., Netes, D., & Sartin, M. (1999). Combining content-based and collaborative filters in an online newspaper. In *In Proceedings of ACM SIGIR Workshop on Recommender Systems*.

[29] Billsus, D., & Pazzani, M. J. (2000). User modeling for adaptive news access. *User modeling and user-adapted interaction*, *10*(2), 147-180.

[30] Burke, R. (2007). Hybrid web recommender systems. *The adaptive web*, 377-408.

[31] Mooney, R. J., & Roy, L. (2000, June). Content-based book recommending using learning for text categorization. In *Proceedings of the fifth ACM conference on Digital libraries* (pp. 195-204).

[32] Pazzani, M. J. (1999). A framework for collaborative, content-based and demographic filtering. *Artificial intelligence review*, *13*(5), 393-408.

[33] Basu, C., Hirsh, H., & Cohen, W. (1998, July). Recommendation as classification: Using social and content-based information in recommendation. In *Aaai/iaai* (pp. 714-720).

[34] Smyth, B., & Cotter, P. (2000). A personalized television listings service. *Communications of the ACM*, *43*(8), 107-111.

[35] Glauber, R., Loula, A., & Rocha-Junior, J. B. (2013). A mixed hybrid recommender system for given names. *ECML PKDD Discovery Challenge*, 25.

[36] Lee, H. H., & Teng, W. G. (2007, August). Incorporating multi-criteria ratings in recommendation systems. In *2007 IEEE International Conference on Information Reuse and Integration* (pp. 273-278). IEEE.

[37] Bartolini, I., Zhang, Z., & Papadias, D. (2010). Collaborative filtering with personalized skylines. *IEEE transactions on knowledge and data engineering*, *23*(2), 190-203.

[38] Rhimi, F., Yahia, S. B., & Ahmed, S. B. (2015, September). Enhancing Skyline Computation with Collaborative Filtering Techniques for QoS-Based Web Services Selection. In *2015 IEEE 14th International Symposium on Network Computing and Applications* (pp. 247-250). IEEE.

[39] Mastrandrea, A. (2019, December 31). *Milan Airbnb Open Data (only entire apartments)*. Kaggle. Retrieved August 26, 2022, from https://www.kaggle.com/datasets/antoniokaggle/milan-airbnb-open-data-only-entire-apartments .

[40] NASA. (n.d.). *TLX @ NASA Ames - Home*. NASA. Retrieved August 26, 2022, from https://humansystems.arc.nasa.gov/groups/tlx/

[41] LydiaBal. (n.d.). *Lydiabal/Thesis*. GitHub. Retrieved August 26, 2022, from https://github.com/LydiaBal/Thesis

[42] Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, *12*(4), 331-370.

[43] Tan, K. L., Eng, P. K., & Ooi, B. C. (2001, September). Efficient progressive skyline computation. In *VLDB* (Vol. 1, pp. 301-310).

[44] Luo, Y., Lu, H. X., & Lin, X. (2004, July). A scalable and I/O optimal skyline processing algorithm. In *International Conference on Web-Age Information Management* (pp. 218-228). Springer, Berlin, Heidelberg.