ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Βελτίωση αλγορίθμων παραγωγής προτάσεων πολλαπλών κριτηρίων που βασίζονται στο συνεργατικό φιλτράρισμα

Γκάτση Ένρι
Mema Gerald

Επιβλέπων: **Ιωάννης Χαμόδρακας**, μέλος του Εργαστηριακού Διδακτικού Προσωπικού υπό την εποπτεία του Καθηγητή **Ιωάννη Εμίρη**

ΑΘΗΝΑ

ΜΑΙΟΣ 2016

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Βελτίωση αλγόριθμων παραγωγής προτάσεων πολλαπλών κριτηρίων που βασίζονται στο συνεργατικό φιλτράρισμα

Γκάτση Ένρι
ΑΜ: 1115200900048

Mema Gerald
ΑΜ: 1115200800108

Επιβλέπων: **Ιωάννης Χαμόδρακας**, μέλος του Εργαστηριακού Διδακτικού Προσωπικού υπό την εποπτεία του Καθηγητή **Ιωάννη Εμίρη**

ΠΕΡΙΛΗΨΗ

Υπάρχει μια ευρεία κατηγορία εφαρμογών Παγκόσμιου Ιστού η οποία αφορά την παραγωγή προτάσεων προς τους χρήστες για πόρους, αντικείμενα και υπηρεσίες που διατίθενται στον ιστό. Οι εφαρμογές αυτές ονομάζονται συστήματα προτάσεων. Τα συστήματα προτάσεων χρησιμοποιούν μια πληθώρα από διαφορετικές προσεγγίσεις. Πιο συχνά συναντώνται συστήματα προτάσεων που βασίζονται στο φιλτράρισμα περιεχομένου (content-based filtering) και τα λεγόμενα συστήματα προτάσεων συνεργατικού φιλτραρίσματος (collaborative filtering). Η συγκεκριμένη πτυχιακή είναι μια επέκταση της πτυχιακής της Γ. Μπουρτσουκλή (2014), στην οποία υλοποιήθηκε ένας αλγόριθμος πρόβλεψης βαθμολογιών για την αξιολόγηση αντικειμένων βάσει πολλαπλών κριτηρίων, με σκοπό την πρότασή τους στους χρήστες, ανάλογα με τις προτιμήσεις τους. Σκοπός της υλοποίησης ήταν η αξιολόγηση και η σύγκριση του αλγορίθμου με δύο άλλους γνωστούς αλγόριθμους που χρησιμοποιούνται ευρέως. Η πτυχιακή αυτή επικεντρώνεται σε δύο σημεία για να βελτιώσει τα αποτελέσματα του αρχικού αλγορίθμου. Πρώτον, γίνεται τροποποίηση της μεθόδου της γραμμικής παλινδρόμησης που χρησιμοποιείται για τον υπολογισμό της βαρύτητας που δίνουν οι χρήστες στα διάφορα κριτήρια αξιολόγησης με σκοπό την βελτίωση των αποτελεσμάτων της και την αντιμετώπιση των περιορισμών της. Επιπλέον, αντιμετωπίζεται το πρόβλημα της διαχείρισης μεγαλύτερου πλήθους χρηστών μέσω της χρήσης του αλγορίθμου προσεγγιστικής εύρεσης κοντινότερων γειτόνων LSH (locality sensitive hashing), ο οποίος επιτυγχάνει τη μείωση των διαστάσεων του προβλήματος και τη βελτίωση της πολυπλοκότητας. Τέλος υλοποιούνται ορισμένες μετρικές αξιολόγησης με σκοπό τη σύγκριση του αλγορίθμου ως προς την αποτελεσματικότητα και τον χρόνο σε σχέση με δύο γνωστούς αλγόριθμους, τον Decomposing Multi Criteria και τον Weighted Slope One, αλλά και με τον αρχικό αλγόριθμο.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Συστήματα Προτάσεων
ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Εξατομίκευση, Συνεργατικό Φιλτράρισμα, Πρόβλεψη Βαθμολογιών, Ομοιότητες Χρηστών, Μετρικές Αξιολόγησης

ABSTRACT

There is an extensive class of Web applications concerning the production of recommendations to users for resources, items and services which are available in the web. Such applications are called recommendation systems. Recommendation systems use a number of different approaches, the most popular being content-based filtering and collaborative filtering. This thesis is an extension of G. Bourtsoukli's thesis (2014), which implemented a rating prediction algorithm to evaluate items according to multiple criteria, in order to recommend these items to the users according to their preferences. This thesis focuses on two aspects to improve the results of the algorithm. First, a modification of the method of linear regression which is used to calculate the weights assigned by the users to each criterion is proposed in order to improve its results and overcome its limitations. Furthermore, the problem of handling a larger number of users is faced by using an algorithm which finds approximate nearest neighbors by LSH (locality sensitive hashing). LSH achieves the reduction of the dimensions of the problem and improves time and space complexity. Last but not least, the modified algorithm is evaluated and compared with regard to its effectiveness and performance with two other popular algorithms, the Decomposing Multi Criteria and Weighted Slope One, as well as the initial algorithm.

SUBJECT AREA: Recommender Systems
KEYWORDS: Personalization, Collaborative Filtering, Rating Estimation Similarities, Evaluation Metrics

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

ΕΥΧΑΡΙΣΤΙΕΣ

Θα θέλαμε να ευχαριστήσουμε θερμά τον Δρ. Ιωάννη Χαμόδρακα για την καθοδήγηση, την υποστήριξη, τη συνεργασία και τη πολύτιμη συμβολή του κατά την εκπόνηση της πτυχιακής εργασίας. Επιπρόσθετα θα θέλαμε να ευχαριστήσουμε θερμά τον Καθηγητή κ. Ιωάννη Εμίρη για την εμπιστοσύνη που μας έδειξε στην ανάθεση της παρούσας πτυχιακής εργασίας και το χρόνο που αφιέρωσε για τη συνολική εποπτεία της.

1. ΕΙΣΑΓΩΓΗ

Τα συστήματα προτάσεων έχουν αλλάξει τον τρόπο με τον οποίο οι εφαρμογές, και ιδιαίτερα οι διαδικτυακές εφαρμογές, επικοινωνούν με τους χρήστες τους. Αντί να παρέχουν μια στατική εμπειρία όπου οι χρήστες βλέπουν μόνο αντικείμενα ή επισκέπτονται υπηρεσίες που ήδη γνωρίζουν, τα συστήματα προτάσεων αυξάνουν την ελκυστικότητα των υπηρεσιών με σκοπό να παρέχουν μια πιο πλούσια εμπειρία. Τα συστήματα προτάσεων παράγουν αυτόνομα προτάσεις για αντικείμενα ή υπηρεσίες προσαρμοσμένες σε κάθε χρήστη στηριζόμενα σε παλαιότερες αγορές, επισκέψεις, αναζητήσεις ή βαθμολογήσεις και ανάλογα με τη συμπεριφορά άλλων χρηστών.

Τέτοια συστήματα μπορούν χρησιμοποιούνται κατά κόρον από επιχειρήσεις παροχής υπηρεσιών ιστού όπως το Netflix προκειμένου να δώσουν μια καλύτερη εμπειρία των υπηρεσιών που παρέχουν στους χρήστες τους με σκοπό τελικά να προσελκύσουν περισσότερους πελάτες. Γενικότερα, η ανάπτυξη συστημάτων προτάσεων αποτελεί δημοφιλές ερευνητικό πεδίο λόγω της μεγάλης χρησιμότητάς τους στην επίλυση του προβλήματος του πληροφοριακού υπερκορεσμού (information overload) το οποίο αντιμετωπίζουν οι χρήστες του παγκόσμιου ιστού.

Ένα καλό σύστημα προτάσεων οφείλει να προτείνει στο χρήστη αντικείμενα που τον ενδιαφέρουν. Εμείς θεωρούμε ότι ενδιαφέρουν τους χρήστες αντικείμενα στα οποία δίνουν καλές βαθμολογήσεις. Εύκολα καταλαβαίνει κανείς ότι αυτό είναι ένα δύσκολο πρόβλημα καθώς πολύ χρήστες τείνουν να μην βαθμολογούν τα αντικείμενα που έχουν δει ή δοκιμάσει. Τα περισσότερα συστήματα προτάσεων προσεγγίζουν το πρόβλημα της παραγωγής προτάσεων κυρίως μέσω δύο βασικών προσεγγίσεων: α) μεθόδους που εντάσσονται στην κατηγορία του συνεργατικού φιλτραρίσματος (collaborative filtering) και β) μεθόδους που βασίζονται στο περιεχόμενο των προηγούμενων επιλογών των χρηστών (content-based filtering).

Στο συνεργατικό φιλτράρισμα η παραγωγή των προτάσεων βασίζεται σε ένα μοντέλο που έχει διαμορφωθεί σύμφωνα με τη παλαιότερη συμπεριφορά ενός χρήστη. Το μοντέλο αυτό μπορεί να κατασκευαστεί συνδυάζοντας τη συμπεριφορά ενός χρήστη με τη συμπεριφορά χρηστών οι οποίοι είναι όμοιοι με τον εκάστοτε χρήστη. Λαμβάνοντας υπόψη τη συμπεριφορά ενός χρήστη, το συνεργατικό φιλτράρισμα χρησιμοποιεί ένα σύνολο γνώσης το οποίο διαμορφώνεται από τις προτιμήσεις άλλων χρηστών. Στην ουσία αυτά τα συστήματα βασίζονται σε μια αυτόματη συνεργασία πολλαπλών χρηστών και φιλτράρουν αυτούς που εμφανίζουν παρόμοιες προτιμήσεις ή συμπεριφορά.[1]

Σε αυτήν την εργασία αναπτύσσουμε ένα σύστημα παραγωγής προτάσεων το οποίο στηρίζεται μόνο στην προσέγγιση του συνεργατικού φιλτραρίσματος. Συγκεκριμένα έχουμε στη διάθεσή μας βαθμολογήσεις σε αντικείμενα τα οποία έχουν ένα ολικό κριτήριο και μερικά επιμέρους κριτήρια. Μπορεί κανείς να σκεφτεί ότι ένα τέτοιο αντικείμενο μπορεί να είναι ένα ξενοδοχείο. Επιμέρους κριτήρια για το ξενοδοχείο μπορεί να αποτελούν η καθαριότητα, οι τιμές, η τοποθεσία και άλλα. Τελικά ο χρήστης μπορεί να δώσει και μια ολική βαθμολογία που πιστεύει ότι είναι αντιπροσωπευτική για το ξενοδοχείο. Σε κάθε ένα από τα κριτήρια, είτε είναι ολικό είτε επιμέρους, υπάρχουν βαθμολογίες που κυμαίνονται μεταξύ των τιμών ένα μέχρι πέντε. Προσπαθούμε να προβλέψουμε την ολική βαθμολογία που θα έδινε ένας χρήστης σε αντικείμενα τα οποία δεν έχει βαθμολογήσει ακόμα, θεωρώντας ότι δεν τα έχει δει ακόμα. Για να το πετύχουμε αυτό καθοριστικός παράγοντας είναι η εύρεση ομοιοτήτων μεταξύ χρηστών με σκοπό να

εκτιμήσουμε τις προτιμήσεις ενός χρήστη. Για να πραγματοποιηθεί αυτό, πρώτον γίνεται αναζήτηση ομοιοτήτων με βάση την βαθμολογία στα επιμέρους κριτήρια, δηλαδή στο πως αξιολογούν οι χρήστες τα κριτήρια των αντικειμένων και δεύτερων στην εύρεση ομοιοτήτων με βάση τα βάρη που δίνουν στα κριτήρια αυτά, με σκοπό να φτιάξουμε ένα μοντέλο της συμπεριφοράς του χρήστη βάσει της δομής των προτιμήσεων του. Για να υπολογιστούν τα βάρη χρησιμοποιείται η μέθοδος της γραμμικής παλινδρόμησης.

Η υλοποίηση του αλγορίθμου της γραμμικής παλινδρόμησης η οποία έγινε από την Μπουρτσουκλή [2] εμφανίζει μια αδυναμία στην περίπτωση που το πλήθος των αντικειμένων που έχει ήδη βαθμολογήσει ένας χρήστης είναι λιγότερα από το πλήθος των επιμέρους κριτηρίων. Προκειμένου να αντιμετωπίσουμε αυτό το πρόβλημα σκεφτήκαμε να βρούμε όμοιους, με το χρήστη που εξετάζουμε, χρήστες και να δανειστούμε βαθμολογίες που έχει δώσει ο όμοιος σε κάποια αντικείμενα με την λογική ότι χρήστες που είναι όμοιοι βαθμολογούν με τον όμοιο τρόπο, οπότε και ο χρήστης που εξετάζουμε εάν είχε βαθμολογήσει τα αντικείμενα που έχει βαθμολογήσει κάποιος όμοιος του πιθανόν θα τα βαθμολογούσε με τον ίδιο τρόπο.

Επίσης τα βάρη, δηλαδή το αποτέλεσμα του αλγορίθμου γραμμικής παλινδρόμησης θα πρέπει να είναι ένα διάνυσμα με θετικές τιμές μεταξύ μηδέν και ένα και το συνολικό άθροισμα των συνιστωσών του διανύσματος θα πρέπει να είναι ίσο με τη μονάδα. Η παραπάνω αναφορά γίνεται για να επισημάνουμε μία αδυναμία της προηγούμενης πτυχιακής στην οποία σε ορισμένες περιπτώσεις δεν τηρείται ο παραπάνω περιορισμός με αποτέλεσμα να παράγεται ένα διάνυσμα με μηδενικές τιμές. Ο λόγος που συμβαίνει αυτό είναι διότι οι βαθμολογίες ενός χρήστη δεν είναι γραμμικά ανεξάρτητες μεταξύ τους. Η βελτίωση σε αυτό το πρόβλημα επιτεύχθηκε στο να κρατήσουμε για τους χρήστες μόνο τις γραμμικά ανεξάρτητες βαθμολογίες και να αγνοήσουμε τις υπόλοιπες. Το πρόβλημα που έχουμε να αντιμετωπίσουμε τώρα είναι αν οι βαθμολογίες που απέμειναν για τον χρήστη είναι λιγότερες από το πλήθος των επιμέρους κριτηρίων, το όποιο όμως επιλύεται με τον τρόπο που περιγράψαμε στην προηγούμενη παράγραφο. Σκοπός αυτής της βελτίωσης είναι να έχουμε πιο σαφή εικόνα για τα βάρη που δίνει ο χρήστης στα επιμέρους κριτήρια ενός αντικειμένου αλλά και στον υπολογισμό των ομοιοτήτων μεταξύ των χρηστών με βάση τα αυτά, που αποσκοπούν στην πρόβλεψη βαθμολογιών.

Προκειμένου όμως να το καταφέρουμε αυτό αντιμετωπίζουμε δύο προβλήματα. Το πρώτο είναι πως θα βρούμε όμοιους χρήστες σε μια βάση δεδομένων που δυνητικά μπορεί να περιέχει εκατομμύρια χρήστες και το δεύτερο είναι πως θα ορίσουμε ότι δύο χρήστες είναι όμοιοι έχοντας μόνο τις βαθμολογίες που έχουν δώσει σε αντικείμενα. Ύστερα από τη καθοδήγηση των επιβλεπόντων μας λύσαμε το πρώτο πρόβλημα με μια υλοποίηση του αλγορίθμου LSH ο οποίος δοθεισών των βαθμολογιών που έχει δώσει κάποιος χρήστης σε αντικείμενα μπορεί να τοποθετήσει χρήστες που βαθμολογούν παρόμοια σε ίδιους κάδους εφαρμόζοντας μια μέθοδο κατακερματισμού που αναλύεται στο τρίτο κεφάλαιο. Για να βρούμε ομοιότητες μεταξύ των χρηστών θεωρήσαμε ότι ο τρόπος που βαθμολογεί ένας χρήστης μπορεί να εκφραστεί από ένα n-διάστατο σημείο, όπου n είναι το πλήθος των επιμέρους κριτηρίων. Ο τρόπος υπολογισμού αυτού του σημείου αναλύεται με λεπτομέρεια στο τρίτο κεφάλαιο. Παρακάτω ακολουθεί ένα παράδειγμα για τον τρόπο με τον οποίο μπορεί ο εξεταζόμενος χρήστης να “δανειστεί” αντικείμενα από κάποιον όμοιο του. Το -1 δηλώνει ότι ο χρήστης δεν έχει βαθμολογήσει αυτό το κριτήριο στο αντικείμενο. Στο παράδειγμα που ακολουθεί οι τιμές δεν αντιπροσωπεύουν πραγματικά δεδομένα αλλά δίνονται για την ευκολότερη κατανόηση

από τον αναγνώστη.

Πίνακας 1: Παράδειγμα βαθμολογιών πριν τον δανεισμό αντικειμένων

	Χρήστης(1)	Χρήστης(2)	Χρήστης(3)
Αντικείμενο(1)	[**2**,3,1,2]	[**2**,3,1,3]	[**3**,1,4,3]
Αντικείμενο(2)	[**4**,4,4,3]	[**4**,2,4,3]	[**1**,1,1,1]
Αντικείμενο(3)	[**3**,1,4,3]	[**3**,1,4,3]	[**5**,5,5,5]
Αντικείμενο(4)	[**-1**,-1,-1,-1]	[**5**,5,5,5]	[**3**,2,2,4]
Αντικείμενο(5)	[**-1**,-1,-1,-1]	[**3**,3,3,3]	[**1**,2,1,1]

Βλέπουμε στον πίνακα ότι το πλήθος των επιμέρους κριτηρίων είναι τρία. Προκειμένου να μπορέσουμε να εφαρμόσουμε γραμμική παλινδρόμηση στις βαθμολογίες του Χρήστη(1) χρειάζεται να έχει βαθμολογήσει πέντε αντικείμενα. Ο LSH θα πρέπει να τοποθετήσει τους Χρήστης(1) και Χρήστης(2) στον ίδιο κάδο επειδή έχουν παρόμοιες βαθμολογίες στα αντικείμενα ένα, δύο και τρία. Αντίστοιχα εφόσον ο Χρήστης(3) δεν έχει όμοιες με το Χρήστη(1) βαθμολογίες θα τοποθετηθεί σε διαφορετικό κάδο. Ο Χρήστης(1) τώρα θα “δανειστεί” τις βαθμολογίες του Χρήστη(2) για τα αντικείμενα τέσσερα και πέντε με αποτέλεσμα οι βαθμολογίες να διαμορφωθούν όπως φαίνεται στον ακόλουθο πίνακα.

Πίνακας 2: Παράδειγμα βαθμολογιών μετά από τον δανεισμό αντικειμένων

	Χρήστης(1)	Χρήστης(2)	Χρήστης(3)
Αντικείμενο(1)	[**2**,3,1,2]	[**2**,3,1,3]	[**3**,1,4,3]
Αντικείμενο(2)	[**4**,4,4,3]	[**4**,2,4,3]	[**1**,1,1,1]
Αντικείμενο(3)	[**3**,1,4,3]	[**3**,1,4,3]	[**5**,5,5,5]
Αντικείμενο(4)	“[**5**,5,5,5]”	[**5**,5,5,5]	[**3**,2,2,4]
Αντικείμενο(5)	“[**3**,3,3,3]”	[**3**,3,3,3]	[**1**,2,1,1]

Τώρα μπορούμε να εφαρμόσουμε γραμμική παλινδρόμηση και να υπολογίσουμε βάρη για το Χρήστη(1) στα επιμέρους κριτήρια του.

Ας δούμε τώρα ένα παράδειγμα της βάσης για να καταλάβουμε καλύτερα τη δομή του προβλήματος που προσπαθούμε να επιλύσουμε.

Μπορούμε να σκεφτούμε ότι η βάση δεδομένων που χρησιμοποιούμε έχει χρήστες. Κάθε χρήστης (X) έχει βαθμολογήσει ένα σύνολο από αντικείμενα. Κάθε αντικείμενο (A) έχει ένα ολικό και κάποια επιμέρους κριτήρια, ας τα ονομάσουμε κριτήρια (K). Επομένως αν η βάση μας αντιπροσωπευόταν από έναν τρισδιάστατο πίνακα η θέση πίνακας[X][A][K] θα περιείχε τη βαθμολογία του χρήστη X στο αντικείμενο A στο κριτήριο K.

Θέλουμε το σύστημά μας να κάνει προβλέψεις για αντικείμενα τα οποία δεν έχει βαθμολογήσει ο χρήστης. Επομένως χρειαζόμαστε τα αντικείμενα που δεν έχουν βαθμολογηθεί από το χρήστη X ώστε μέσω των μεθόδων που αναφέρθηκαν νωρίτερα

να προβλέψουμε την τιμή με την οποία θα βαθμολογούσε ο χρήστης αυτό το αντικείμενο όσο το δυνατόν καλύτερα. Αυτό που επιθυμούμε είναι να διερευνήσουμε τι αποτελέσματα προσφέρουν διάφορες τεχνικές βάση συγκεκριμένων μετρικών απόδοσης.

Ας δούμε δύο πίνακες με βαθμολογίες χρηστών. Στο πρώτο πίνακα παρουσιάζονται μόνο οι ολικές βαθμολογίες των χρηστών σε διάφορα αντικείμενα. Στο δεύτερο πίνακα εμφανίζονται και οι βαθμολογίες σε επιμέρους κριτήρια. Βλέπουμε ότι ο Χρήστης(3) δεν έχει βαθμολογήσει το Αντικείμενο(4). Δεδομένου ότι ο Χρήστης(1) είναι παρόμοιος με τον Χρήστης(3) θα μπορούσαμε να συμπεράνουμε ότι η βαθμολογία που θα δώσει ο Χρήστης(3) στο Αντικείμενο(4) είναι 5.

Πίνακας 3: Παράδειγμα με ολική βαθμολογία αντικειμένων

	Αντικείμενο(1)	Αντικείμενο(2)	Αντικείμενο(3)	Αντικείμενο(4)
Χρήστης(1)	4	2	3	5
Χρήστης(2)	4	4	1	2
Χρήστης(3)	4	2	3	?

Στη συνεχεία χρησιμοποιώντας μία βάση με βαθμολογίες σε πολλαπλά κριτήρια έχουμε τον παρακάτω πίνακα.

Πίνακας 4: Παράδειγμα με βαθμολογίες αντικειμένων σε πολλαπλά κριτήρια

	Αντικείμενο(1)	Αντικείμενο(2)	Αντικείμενο(3)	Αντικείμενο(4)
Χρήστης(1)	[**4**,5,3,4]	[**2**,2,2,3]	[**3**,5,3,1]	[**5**,5,5,5]
Χρήστης(2)	[**4**,4,4,4]	[**4**,5,1,4]	[**2**,5,1,1]	[**2**,5,1,1]
Χρήστης(3)	[**4**,4,4,4]	[**2**,5,1,0]	[**3**,5,1,2]	?

Σε αυτή τη περίπτωση ο πλησιέστερος στον Χρήστης(3) με βάση τα πολλαπλά κριτήρια είναι ο Χρήστης(2). Σε αυτή τη περίπτωση το σύστημα προτάσεων θα προέβλεπε ότι ο Χρήστης(3) για το Αντικείμενο(4) θα δώσει βαθμολογία ίση με δύο.

Ένας από τους στόχους είναι η πειραματική εξέταση της παραπάνω παραδοχής και η επιλογή της αποδοτικότερης προσέγγισης για την πρόβλεψη των προτιμήσεων των χρηστών.

Τελικά γίνεται σύγκριση των αποτελεσμάτων με κάποιες ήδη γνωστές υλοποιήσεις. Αυτές είναι οι Weighted Slope One[3], ο αλγόριθμος που αναφέρεται στο άρθρο Adomavicious και Kwon [4] ο οποίος αποσυνθέτει το αρχικό πρόβλημα σε k υποπροβλήματα, με την παράμετρο k να αντιπροσωπεύει τον αριθμό των κριτηρίων. Παράγεται μια πρόβλεψη για κάθε επιμέρους κριτήριο. Ύστερα συνθέτονται τα υποπροβλήματα που επιλύθηκαν παράγοντας το ολικό κριτήριο και οι εξάγονται οι αντίστοιχες προβλέψεις. Εξετάζουμε και τον αρχικό αλγόριθμο πολλαπλών βημάτων ο οποίος είναι ο τελευταίος που θα δούμε σε αυτή την εργασία.

2. ΑΝΑΛΥΣΗ ΓΝΩΣΤΩΝ ΑΛΓΟΡΙΘΜΩΝ

Σε αυτό το κεφάλαιο θα περιγράψουμε δύο γνωστούς αλγόριθμους τους οποίους θα χρησιμοποιήσουμε αργότερα για να συγκρίνουμε τα αποτελέσματά τους.

2.1 Αποσύνθεση πολλαπλών κριτηρίων

Ο αλγόριθμος αυτός αρχικά διασπάει το πρόβλημα σε k επιμέρους προβλήματα , όπου k είναι ο αριθμός των επιμέρους κριτηρίων. Για κάθε ένα επιμέρους κριτήριο υπολογίζονται ομοιότητες μεταξύ ενός χρήστη με όλους τους υπόλοιπους χρήστες οι οποίες θα χρησιμοποιηθούν για να παραχθεί η πρόβλεψη της αξιολόγησης που θα έδινε ο χρήστης για το συγκεκριμένο κριτήριο. Για τον υπολογισμό των ομοιοτήτων χρησιμοποιούμε την Pearson correlation-based similarity που περιγράφεται στο κεφάλαιο 3.2.

Έχοντας προβλέψει τις βαθμολογίες για τα επιμέρους κριτήρια αυτό που μας απομένει είναι να βρούμε τους συντελεστές ,δηλαδή τα βάρη που δίνει ένας χρήστης στο κάθε επιμέρους κριτήριο. Όπως και στον προηγούμενο αλγόριθμο θα χρησιμοποιήσουμε την μέθοδο της γραμμικής παλινδρόμησης για να βρούμε αυτούς τους συντελεστές.

Τελευταίο βήμα είναι να υπολογίσουμε την ολική βαθμολογία μέσω του τύπου

$$y_j = \sum x_{ji} c_{ji} \quad (2.1)$$

όπου j είναι ο δείκτης για κάθε αντικείμενο και i δείκτης για κάθε επιμέρους κριτήριο. Ο αλγόριθμος σε ψευδογλώσσα είναι ο εξής.

Αλγόριθμος Αποσύνθεσης πολλαπλών κριτηρίων:

1. Για κάθε επιμέρους κριτήριο υπολόγισε τις ομοιότητες μεταξύ των χρηστών.

2. Υπολόγισε την πρόβλεψη για κάθε επιμέρους κριτήριο x_i .

3. Υπολόγισε τους συντελεστές c_i οπου i = 0,1,...,k με χρήση της γραμμικής παλινδρόμησης.

4. Υπολόγισε την ολική βαθμολογία $y_j = \sum x_{ji} c_{ji}$ με j = 0,1,...,m αντικείμενα και i = 0,1,2,...,k επιμέρους κριτήρια.

2.2 Μέθοδος Σταθμισμένης Μοναδιαίας Κλίσης (Weighted Slope One)

Η μέθοδος Weighted Slope One είναι μια δημοφιλής μέθοδος συνεργατικού φιλτραρίσματος. Η μέθοδος αυτή κάνει προβλέψεις με τη χρήση μιας μοναδικής βαθμολογίας ανά αντικείμενο και στην περίπτωση μας θα χρησιμοποιήσουμε την ολική βαθμολογία. Η Weighted Slope One είναι μια επέκταση της κλασσικής Slope One, η οποία βασίζεται τόσο στην έννοια της διαφορικής δημοτικότητας μεταξύ των αντικειμένων, δηλαδή πόσο περισσότερο προτιμάται ένα αντικείμενο σε σχέση με ένα άλλο, όσο και στη στάθμιση των προβλέψεων, βάσει του αριθμού των χρηστών που βαθμολόγησαν ένα αντικείμενο. Το πλεονέκτημα της μεθόδου αυτής είναι ότι έχει σχετικά μικρή πολυπλοκότητα χρόνου, ενώ ταυτόχρονα η ορθότητα της είναι συχνά στο ίδιο επίπεδο με έναν πιο πολύπλοκο και δαπανηρό υπολογιστικά αλγόριθμο. Τα βήματα του αλγορίθμου είναι δύο.

2.2.1 Διαφορική απόκλιση

Υπολογίζουμε την διαφορική απόκλιση για κάθε ζεύγος αντικειμένων. Έστω S_{ij} το σύνολο των χρηστών οι οποίοι έχουν βαθμολογήσει το αντικείμενο i και το αντικείμενο j και $|S_{ij}|$ το πλήθος του συνόλου αυτού. Τότε ο τύπος για την απόκλιση μεταξύ των δύο αυτών αντικειμένων είναι

$$dev_{ij} = \frac{\sum_{u \in S_{ij}} r(u,i) - r(u,j)}{|S_{ij}|} \quad , (2.2)$$

2.2.2 Υπολογισμός ομοιοτήτων

Χρησιμοποιώντας την διαφορική απόκλιση μπορούμε να κάνουμε την πρόβλεψη από τον παρακάτω τύπο.

$$r(u,i) = \frac{\sum_{j \in R_u} dev_{ij} + r(u,j)|S_{ij}|}{\sum_{j \in Ru} |S_{ij}|} \quad , (2.3)$$

όπου R_u είναι το σύνολο των αντικειμένων που έχει βαθμολογήσει ο χρήστης u.

Μπορούμε να δούμε ένα παράδειγμα για να κατανοήσουμε καλύτερα τους παραπάνω τύπους. Ας υποθέσουμε ότι έχουμε τρεις χρήστες που έχουν βαθμολογήσει κάποια αντικείμενα όπως βλέπουμε στο παράδειγμα του πίνακα 5.

Πίνακας 5: Πίνακας βαθμολογιών

User/Item	A	B	C
John	5	3	2
Mark	3	4	-
Lucy	-	2	5

Έστω ότι εμείς θέλουμε να προβλέψουμε την βαθμολογία της Lucy για το αντικείμενο A. Εφαρμόζοντας τον τύπο υπολογισμού της διαφορικής απόκλισης προκύπτει πως $dev_{AB} = \frac{(5-3)+(3-4)}{2} = 0.5$ και $dev_{AC} = \frac{5-2}{1} = 3$. Ύστερα μπορούμε να προβλέψουμε τη βαθμολογία της Lucy για το αντικείμενο A και το αποτέλεσμα είναι $r_{LucyA} = \frac{(0.5+2)*2+(3+5)*1}{2+1} = 4.333$.

Ο πίνακας με τα δεδομένα μας θα αποτελείται από έναν τρισδιάστατο πίνακα που σαν στοιχεία θα έχει τις βαθμολογίες που έχουν δώσει οι χρήστες. Θα έχουμε δηλαδή έναν X-άξονα που θα αντιστοιχεί στους χρήστες, έναν άξονα Y που θα αντιστοιχεί στα αντικείμενα και ένα άξονα Z που θα αντιστοιχεί στα επιμέρους κριτήρια συν την ολική βαθμολογία η οποία θα βρίσκεται ως πρώτο στοιχείο στον άξονα Z. Οι βαθμολογίες των αντικειμένων που δεν έχει βαθμολογήσει ο χρήστης θα αρχικοποιούνται με -1 στην ολική

βαθμολογία και στα επιμέρους κριτήρια.

3. Αλγόριθμος Πολλαπλών Βημάτων Πολλαπλών Κριτηρίων με χρήση του αλγορίθμου LSH και αναβαθμισμένης γραμμικής παλινδρόμησης

Ο αλγόριθμος πολλαπλών βημάτων πολλαπλών κριτηρίων υλοποιήθηκε αρχικά στην πτυχιακή εργασία με τίτλο “Ανάπτυξη συστήματος προτάσεων πολλαπλών κριτηρίων”[5] και στην συνέχεια επεκτάθηκε από μια δεύτερη πτυχιακή εργασία με τίτλο “Αξιολόγηση μεθόδων προτάσεων πολλαπλών κριτηρίων”. Η παρούσα εργασία επεκτείνει τον αλγόριθμο με την προσθήκη νέων τεχνικών σε κάθε βήμα του. Ο εν λόγω αλγόριθμος αποτελείται από μια σειρά βημάτων με μια σειρά επιλογών που συνοδεύει το κάθε βήμα. Παρακάτω παρατίθενται αναλυτικά τα βήματα του αλγορίθμου MSMC-LSH.

3.1 Συνάρτηση συνάθροισης

Η συνάρτηση συνάθροισης μας δίνει την συσχέτιση που έχει η ολική βαθμολογία με τις επιμέρους βαθμολογίες για τα αντικείμενα που έχει βαθμολογήσει ο χρήστης. Για παράδειγμα αν ο χρήστης έχει βαθμολογήσει ορισμένα αντικείμενα μαζί με τα επιμέρους κριτήρια, τότε η κατάλληλη συνάρτηση συνάθροισης θα ήταν μία συνάρτηση η οποία όταν της δίνουμε ως όρισμα τις επιμέρους βαθμολογίες και την ολική βαθμολογία να μας επιστρέφει τους συντελεστές-βάρη, στα οποία το άθροισμα των γινομένων των επιμέρους βαθμολογιών επί τον αντίστοιχο συντελεστή θα έχουν σαν αποτέλεσμα την ολική βαθμολογία.

Το ζητούμενο δηλαδή είναι να βρούμε μέσω κάποιας τεχνικής τι βάρος δίνει ο κάθε χρήστης στα κριτήρια αξιολόγησης . Υπάρχουν πολλοί τρόποι για την εύρεση της συνάρτησης αυτής από πλευράς μηχανικής μάθησης και στατιστικών τεχνικών, όπως τα νευρωτικά δίκτυα (neural networks), μηχανές διανυσμάτων υποστήριξης (support vector machines), η γραμμική και η μη γραμμική παλινδρόμηση. Εμείς θα χρησιμοποιήσουμε την τεχνική της γραμμικής παλινδρόμησης.

Με την μέθοδο της απλής γραμμικής παλινδρόμησης ψάχνουμε να βρούμε μια ευθεία $f(x)=y=\alpha+\beta x$ η οποία θα “ταιριάζει” καλύτερα στα δείγματα τιμών { x_i, y_i } που έχουμε. Ουσιαστικά ψάχνουμε να βρούμε τις κατάλληλες τιμές α και β. Για να βρεθεί αυτή η ευθεία, δηλαδή οι παράμετροι α και β, μπορεί να χρησιμοποιηθεί η μέθοδος των ελαχίστων τετραγώνων. Προσπαθούμε να βρούμε μια ευθεία όπου η απόσταση κάθε σημείου x_i, y_i να είναι ελάχιστη. Βρες $min_{a,\beta} Q(\alpha,\beta)$ όπου

$$Q(\alpha,\beta)=\sum_{i=1}^{n}\left(\hat{e}_i\right)^2=\sum_{i=1}^{n}\left(y_i-\alpha-\beta x_i\right)^2 \quad , (3.1)$$

Αρχικά θα αναφέρουμε κάποιες λειτουργίες που πρέπει να εφαρμόσουμε πριν την υλοποίηση αυτής της μεθόδου. Για να εφαρμόσουμε γραμμική παλινδρόμηση θα πρέπει να ισχύει M > N, όπου N είναι ο αριθμός των επιμέρους κριτηρίων και M ο αριθμός των αξιολογημένων αντικειμένων, δηλαδή ο αριθμός των αντικειμένων που έχει αξιολογήσει ο χρήστης πρέπει είναι μεγαλύτερος από των αριθμό των επιμέρους κριτηρίων. Αυτό είναι λογικό αν σκεφτούμε πως για να επιλύσουμε ένα σύστημα γραμμικών εξισώσεων θα πρέπει ο αριθμός των εξισώσεων να είναι μεγαλύτερος από τον αριθμό των αγνώστων αλλιώς θα έχουμε άπειρο σύνολο λύσεων. Επιπροσθέτως, ένας άλλος περιορισμός για την ομαλή λειτουργία της γραμμικής παλινδρόμησης που πρέπει να λάβουμε υπόψη είναι η σχέση των βαθμολογιών μεταξύ τους. Θα πρέπει να έχουμε

τουλάχιστον N+1 από τις M βαθμολογίες, γραμμικά ανεξάρτητες μεταξύ τους. Στην εργασία αυτή αντιμετωπίζουμε το πρόβλημα που αναδύεται όταν οι βαθμολογίες του χρήστη δεν πληρούν αυτούς τους περιορισμούς. Για να το κάνουμε αυτό η γενική ιδέα είναι να "δανειστούμε" βαθμολογίες από άλλους όμοιους χρήστες. Εδώ εξετάζεται ένα σημαντικό ζήτημα το οποίο είναι με ποιον τρόπο θα επιλέξουμε τους χρήστες αυτούς έτσι ώστε ο τρόπος με τον οποίο βαθμολογεί ένας χρήστης να μην αλλοιωθεί. Όταν αναφερόμαστε στον τρόπο εννοούμε την βαρύτητα που δίνει ένας χρήστης στα επιμέρους κριτήρια και όχι στην ίδια την βαθμολογία. Για παράδειγμα αν έχουμε 3 χρήστες οι οποίοι έχουν βαθμολογήσει 3 αντικείμενα ο καθένας , έστω δηλαδή έχουμε τα αντικείμενα Α, Β, Γ, και οι βαθμολογίες τους είναι οι εξής.

Πίνακας 6: Παράδειγμα ομοιότητας με βάρη

Χρήστες/Αντικείμενα	Α	Β	Γ	Ολικό
Χρήστης 1	5	4	4	5
Χρήστης 2	5	4	4	4
Χρήστης 3	2	3	3	2

Αν θέλουμε να βρούμε για τον χρήστη 1 όμοιους γείτονες έτσι ώστε να δανειστούμε βαθμολογίες και έχουμε να επιλέξουμε ανάμεσα στον χρήστη 2 και 3 τότε στην προκειμένη περίπτωση θα επιλεγεί ο χρήστης 3, διότι όπως βλέπουμε στο παράδειγμα ο χρήστης 1 και 2 έχουν δώσει ακριβώς τις ίδιες βαθμολογίες στα αντικείμενα αλλά διαφέρουν στην ολική βαθμολογία. Αυτό οφείλεται κυρίως γιατί ο χρήστης 1 έχει δώσει ιδιαίτερη βαρύτητα στο αντικείμενο Α. Αν παρατηρήσουμε και τις βαθμολογίες του χρήστη 3 θα συμπεράνουμε το ίδιο πράγμα. Άρα καταλήγουμε ότι ο χρήστης 1 και 3 έχουν πιο παρόμοια χαρακτηριστικά ως προς την έμφαση που δίνουν σε κάθε επιμέρους κριτήριο για κάποιο αντικείμενο. Για τον λόγο αυτό θα πρέπει να βρούμε έναν τρόπο με τον οποίο θα συσχετίσουμε αυτά τα βάρη μεταξύ των χρηστών. Όπως γνωρίζουμε η ολική βαθμολογία έπεται από τον τύπο 3.2.

$$y_j = c_{0i} + \sum x_{ji} * c_{ji} \quad , \quad 1 \le j \le N \quad \text{και} \quad 1 \le i \le M \quad (3.2)$$

Όπου N το πλήθος των αντικειμένων και M το πλήθος των επιμέρους κριτηρίων. Επομένως θα πρέπει να βρούμε γειτονικούς χρήστες σε έναν χώρο n διαστάσεων, όπου n είναι το πλήθος των επιμέρους κριτηρίων, με βάση τα c τα οποία όμως δεν γνωρίζουμε αλλά μπορούμε να τα συσχετίσουμε με τον μέσο όρο των λόγων $\frac{y}{xi}$ του κριτηρίου i για κάθε αντικείμενο. Ο αλγόριθμος του κοντινότερου γείτονα που χρησιμοποιούμε είναι ο LSH που θα τον περιγράψουμε παρακάτω. Ο αλγόριθμος αυτός βρίσκει γειτονικούς χρήστες στον χρήστη που εξετάζουμε με βάση ένα διάνυσμα που παράγεται από τον τύπο 3.3.

$$v_i = Avg_j \left(\Sigma \frac{y_j}{x_{ij}} \right) \quad , (3.3)$$

Όπου $i = 1,2,\ldots,n$ και $j = 1,2,..,m$, m είναι το πλήθος των αντικειμένων που έχει βαθμολογήσει ο χρήστης. Αφού λάβουμε υπόψη τους παραπάνω περιορισμούς

μπορούμε να υλοποιήσουμε την γραμμική παλινδρόμηση με πίνακες. Οι πίνακες που θα χρειαστούμε είναι 3 και σαν αποτέλεσμα ο αλγόριθμος θα μας επιστρέψει έναν πίνακα-διάνυσμα που θα περιέχει τους συντελεστές (τα βάρη) για κάθε επιμέρους κριτήριο. Ο πρώτος πίνακας Y θα αναπαριστά ένα διάνυσμα που θα περιλαμβάνει όλες τις ολικές βαθμολογίες για κάθε αξιολογημένο αντικείμενο του χρήστη. Χρειαζόμαστε επίσης έναν πίνακα X μεγέθους n επί m, όπου n είναι ο αριθμός των επιμέρους κριτηρίων συν ένα για το ολικό κριτήριο, η οποία στήλη (δηλαδή η πρώτη) του ολικού κριτηρίου αρχικοποιείται με μονάδα και m ο αριθμός των αξιολογημένων αντικειμένων (όσο δηλαδή του μέγεθος του πίνακα Y) και ένα μοναδιαίο πίνακα-διάνυσμα W ίδιου μεγέθους με το Y. Παρακάτω περιγράφεται ο αλγόριθμος της γραμμικής παλινδρόμησης [6].

Αλγόριθμος Γραμμικής Παλινδρόμησης:

1. Υπολογισμός Πίνακα V ελάχιστων τετραγώνων με βάση τον πίνακα X.
2. Υπολογισμός Πίνακα διανύσματος B το οποίο προκύπτει από τα αθροίσματα

$$\left[\sum W_i X_{0i} Y_i, \sum W_i X_{1i} Y_i, ..., \sum W_i X_{ji} Y_i\right]$$

3. Υπολογισμός του $D = V^{-1}$
4. Υπολογισμός συντελεστών από τον τύπο

$$C = \left[\sum D_{0i} B_i, \sum D_{1i} B_i, ..., \sum D_{ji} B_i\right] \quad , (3.5)$$

3.2 Υπολογισμός ομοιοτήτων με βάση τους συντελεστές των επιμέρους κριτηρίων

Σε αυτό το βήμα θα υπολογίσουμε τις ομοιότητες μεταξύ των χρηστών με βάση τα βάρη που δίνει ο καθένας στα επιμέρους κριτήρια των αντικειμένων. Κάθε χρήστης θα μπορούσε να αναπαρασταθεί ως ένα διάνυσμα μήκους n, όπου n το πλήθος των επιμέρους κριτηρίων και σαν στοιχεία του διανύσματος να έχουμε τους συντελεστές του χρήστη. Επειδή το βήμα αυτό θα το χρησιμοποιήσουμε στην πρόβλεψη που θα κάνουμε για το πως ένας χρήστης βαθμολογεί ένα αντικείμενο θα πρέπει να γνωρίζουμε την ομοιότητα που έχει ο κάθε χρήστης με άλλους χρήστες. Για να υπολογίσουμε τις ομοιότητες των χρηστών με όλους τους χρήστες αλλά για να γλιτώσουμε αυτήν την τετραγωνική πολυπλοκότητα ως προς τον πλήθος των χρηστών θα χρησιμοποιήσουμε και πάλι τον αλγόριθμο LSH για την εύρεση των κοντινότερων γειτόνων. Σαφώς και έχουμε ένα κόστος που απαιτεί ο αλγόριθμος αυτός για να τοποθετεί χρήστες σε διαφορετικούς κάδους αλλά κερδίζουμε εξετάζοντας έναν πιο μικρό πίνακα δεδομένων. Υπάρχουν πολλοί τρόποι για να υπολογίσουμε ομοιότητες που ουσιαστικά είναι οι αποστάσεις που έχουν οι χρήστες μεταξύ τους ως προς τα βάρη τους. Στο βήμα 2 δίνεται η δυνατότητα επιλογής της μεθόδου του υπολογισμού της ομοιότητας ανάμεσα σε αυτή με βάση το συνημίτονο (Cosine-base similarity)[7] και την βασισμένη στη συσχέτιση Pearson (Pearson correlation-based similarity)[8]. Ας δώσουμε ένα παράδειγμα για να καταλάβει καλύτερα ο αναγνώστης την έννοια της ομοιότητας με βάση το συνημίτονο.

Έστω ότι έχουμε το σημείο w_i που αντικατοπτρίζει τα βάρη του χρήστη i και το σημείο w_j αντικατοπτρίζει τα βάρη του χρήστη j σε χώρο n διαστάσεων, όσο δηλαδή το πλήθος των επιμέρους κριτηρίων. Αν υποθέσουμε ότι φτιάχνουμε 2 ευθείες έτσι ώστε να

ξεκινάνε από την αρχή των αξόνων και η πρώτη να περνάει από το σημείο w_i και η δεύτερη από το σημείο w_j τότε η Cosine-base similarity ουσιαστικά εξετάζει ομοιότητες με βάση το συνημίτονο της γωνίας που εμφανίζουν οι 2 ευθείες μεταξύ τους, αν δηλαδή η γωνία είναι μικρή τότε οι χρήστες αυτοί μοιάζουν πιο πολύ από 2 χρήστες που έχουν μεγάλη γωνία. Οι τιμές που μπορεί να πάρει το συνημίτονο που υπολογίζουμε είναι από -1 έως 1. Ο τύπος της παραπάνω μεθόδου είναι ο εξής.

$$S(u,u')=\frac{\left(\sum c_{u,i}c_{u',i}\right)}{\left(\sqrt{\sum c_{u,i}^2}\sqrt{\sum c_{u',i}^2}\right)} \quad , (3.4)$$

Όπου $S(u,u')$ είναι η ομοιότητα του χρήστη u με τον χρήστη u' και $c_{u,i}$ ο συντελεστής του χρήστη u για το i κριτήριο $c(u',i)$ ο συντελεστής του χρήστη u' για το i κριτήριο.

Η μέθοδος Pearson correlation-based similarity που βασίζεται στο πόσο κοντά είναι τα σημεία μας από μία διαχωριστική γραμμή, προσαρμόζει ουσιαστικά την ομοιότητα μεταξύ δύο χρηστών με βάση τον μέσο όρο των συντελεστών. Οι δυνατές τιμές που παίρνει η μέθοδος αυτή είναι από -1 έως 1. Αν πάρουμε σαν αποτέλεσμα μια τιμή κοντά στο 0 τότε σημαίνει ότι δύο χρήστες δεν έχουν συσχέτιση μεταξύ τους. Αν πάρουμε θετική συσχέτιση τότε έχουμε μία θετική συσχέτιση αλλιώς αν πάρουμε αρνητική τιμή σημαίνει αρνητική συσχέτιση, δηλαδή αν οι συντελεστές ενός χρήστη μεταβληθούν τότε θα έχουμε αντιστρόφως ανάλογη μεταβολή συντελεστών ενός άλλου χρήστη. Ο τύπος της παραπάνω μεθόδου είναι

$$S(u,u')=\frac{\sum\left(c_{u,i}-Avg\,C_u\right)\left(c_{u',i}-AvgC_{u'}\right)}{\sqrt{\sum\left(c_{(u,i)}-AvgC_u\right)^2}\sqrt{\sum\left(c_{u',i}-AvgC_{u'}\right)^2}} \quad , (3.6)$$

$S(u,u')$ είναι η ομοιότητα του χρήστη u με τον χρήστη u', $c_{u,i}$ ο συντελεστής του χρήστη u για το i_κριτήριο, $c_{u',i}$ ο συντελεστής του χρήστη u' για το i κριτήριο, $AvgC_u$ ο μέσος όρος των συντελεστών του χρήστη u και $AvgC_{u'}$ ο μέσος όρος των συντελεστών του χρήστη u'.

Τέλος γίνεται η κανονικοποίηση των παραπάνω τιμών στο διάστημα [0,1] με βάση το τύπο.

$$S=\frac{S-S_{min}}{S_{max}-S_{min}} \quad , (3.7)$$

3.3 Εύρεση των ομοιοτήτων με βάση το ολικό κριτήριο

Σε αυτό το βήμα θα υπολογίσουμε τις ομοιότητες μεταξύ των χρηστών με βάση την ολική βαθμολογία που δίνει ο καθένας στα αντικείμενα που έχει αξιολογήσει. Θα κρατήσουμε και σε αυτήν την περίπτωση την ίδια λογική με την παραπάνω με την διαφορά ότι αντί για διάνυσμα θα έχουμε έναν πραγματικό αριθμό ο οποίος θα είναι ο μέσος όρος των ολικών βαθμολογιών του κάθε χρήστη. Εδώ βέβαια έχουμε μια ιδιαιτερότητα σε σχέση με πριν διότι δεν μπορούμε να εφαρμόσουμε απ' ευθείας τον αλγόριθμο LSH πάνω σε αυτά τα σημεία αν δεν εξετάσουμε πρώτα αν δύο χρήστες έχουν αξιολογήσει τουλάχιστον έναν αριθμό κοινών αντικειμένων. Για παράδειγμα αν έχουμε δύο χρήστες και έξι αντικείμενα και οι χρήστες έχουν βαθμολογήσει από τρία αντικείμενα από τα οποία κανένα δεν είναι κοινό μεταξύ τους, τότε αν η βαθμολόγηση

αυτών των αντικειμένων μοιάζει είναι πιθανόν ο LSH να θεωρήσει ότι και οι χρήστες μοιάζουν μεταξύ τους αλλά στην πραγματικότητα δεν έχουν αξιολογήσει κάποιο κοινό αντικείμενο. Οπότε ψάχνουμε για γειτονικούς χρήστες σε κάποιο υποσύνολο χρηστών, αλλά στο υποσύνολο αυτό θα πρέπει οι χρήστες να έχουν βαθμολογήσει παρόμοια αντικείμενα. Στο βήμα 3 δίνεται η δυνατότητα επιλογής ανάμεσα στη μέθοδο Cosine-base similarity και την Pearson correlation-based similarity.

3.4 Σύνθεση ομοιοτήτων

Στα βήματα 2 και 3 υπολογίσαμε τις ομοιότητες μεταξύ των χρηστών είτε με βάση τα βάρη που δίνανε σε κάθε επιμέρους κριτήριο είτε με βάση την ολική βαθμολογία τους σε κάποια κοινά βαθμολογημένα αντικείμενα . Σκοπός μας είναι να κάνουμε πρόβλεψη και να λάβουμε υπόψη και τις δύο παραπάνω ομοιότητες. Για τον λόγο αυτό θα χρησιμοποιήσουμε τη σύνθεση των ομοιοτήτων ενός χρήστη για να υπολογίσουμε την τελική ομοιότητα η οποία θα έχει υπολογιστεί με βάση k γείτονες, όπου k είναι το πλήθος γειτόνων με βάση τα βάρη συν το πλήθος γειτόνων με βάση τα ολικά κριτήρια μείον το πλήθος κοινών γειτόνων. Στους κοινούς γείτονες θα εφαρμόσουμε $SimW * SimO$, όπου SimW η ομοιότητα με βάση τα βάρη και SimO η ομοιότητα με βάση τα ολικά κριτήρια , και έτσι θα αντιπροσωπεύουμε έναν χρήστη με ένα διάνυσμα μήκους k με τις τελικές ομοιότητες.

Έχουμε επίσης άλλες τρεις επιλογές για να κάνουμε αυτήν την σύνθεση. Οι τρεις αυτές τεχνικές βασίζονται στην παράσταση

$$\mu^c = 1 - min\left[1, \left[\left(1-\mu^A\right)^p + \left(1-\mu^B\right)^p\right]^{\left(\frac{1}{p}\right)}\right], p \geq 1 \quad , (3.8)$$

όπως αναφέρεται στο άρθρο των I. Chamodrakas , N. Alexopoulou, D. Martakos [9]. Αν το p τείνει στο άπειρο τότε το μ^c προκύπτει από τον τύπο 3.9.

$$\mu^c = min\left[\mu^A, \mu^B\right] \quad , (3.9)$$

Γνωστό ως τελεστής του Zadeh. Αν το p είναι ίσο με ένα τότε το μ^c υπολογίζεται από τον τύπο 3.10.

$$\mu^c = max\left[0, \mu^A + \mu^B - 1\right] \quad , (3.10)$$

Γνωστό ως τελεστής του Lukasiewiczs. Αν το p τείνει στο δύο τότε όπου μ^c είναι η συνάθροιση της ομοιότητας από τα βάρη μ^A και της ομοιότητας από τα ολικά κριτήρια μ^B .

$$\mu^c = 1 - min\left[1, \left[\left(1-\mu^A\right)^2 + \left(1-\mu^B\right)^2\right]^{(1/2)}\right] \quad , (3.11)$$

3.5 Παραγωγή προβλέψεων βαθμολογιών

Στο τελευταίο βήμα του αλγορίθμου γίνεται η πρόβλεψη των βαθμολογιών για τα αντικείμενα τα οποία ο χρήστης δεν έχει βαθμολογήσει ακόμα. Από το βήμα 4 έχουμε την πληροφορία της ομοιότητας ενός χρήστη με κάποιο υποσύνολο χρηστών το οποίο και θα χρησιμοποιήσουμε για να παράγουμε το ζητούμενο αποτέλεσμα.

Εμείς θα προβλέψουμε τις βαθμολογίες του χρήστη για όλα τα αντικείμενα τα οποία έχουν βαθμολογήσει κάποιο υποσύνολο των γειτόνων του και στο τέλος θα τα συγκρίνουμε με τα πραγματικά δεδομένα μας για να γίνουν οι κατάλληλες μετρήσεις.

Η πρώτη μέθοδος που χρησιμοποιείται είναι αυτή της σταθμισμένης προσέγγισης (weighted sum approach) η οποία δίνεται από τον τύπο

$$R(u,i)=\frac{\left(\sum S(u,u')*R(u',i)\right)}{\left|\left(\sum S(u,u')\right)\right|} \quad , (3.12)$$

όπου $R(u,i)$ είναι η βαθμολογία του χρήστη που θέλουμε να κάνουμε την πρόβλεψη. $S(u,u')$ η ομοιότητα που έχει ο χρήστης που θέλουμε να κάνουμε την πρόβλεψη με τον γείτονα u'. $R(u',i)$ η βαθμολογία του γείτονα u'.

Η δεύτερη μέθοδος που χρησιμοποιούμε για την πρόβλεψη είναι η προσαρμοζόμενη σταθμισμένη προσέγγιση (adjusted weighted sum approach) όπου σε σχέση με πριν προσαρμόζουμε την πρόβλεψη με βάση τον μέσο των όρο βαθμολογιών του κάθε χρήστη που κάνουμε την πρόβλεψη. Ο τύπος της έχει ως εξής

$$R(u,i)=Avg(R(u))+\sum \frac{S(u,u')*\left(R(u',i)-Avg(R(u'))\right)}{\left|\left(\sum S(u,u')\right)\right|} \quad , (3.13)$$

$R(u,i)$ είναι η βαθμολογία του χρήστη που θέλουμε να κάνουμε την πρόβλεψη. $S(u,u')$ είναι η ομοιότητα που έχει ο χρήστης που θέλουμε να κάνουμε την πρόβλεψη με τον γείτονα u'. $R(u',i)$ είναι η βαθμολογία του γείτονα u', το $Avg(R(u))$ αντιπροσωπεύει το μέσο όρο βαθμολογίας του χρήστη που θέλουμε να κάνουμε την πρόβλεψη και $Avg(R(u'))$ είναι ο μέσος όρος βαθμολογίας του γείτονα u'. Ο Αλγόριθμος πολλαπλών βημάτων πολλαπλών κριτηρίων περιγράφεται από τα εξής βήματα.

Βήμα 1: Αν ισχύουν οι περιορισμοί της γραμμικής παλινδρόμησης υπολόγισε τα βάρη. Αλλιώς χρησιμοποιώντας βαθμολογίες των κοντινότερων γειτόνων εκπλήρωσε τους περιορισμούς της γραμμικής παλινδρόμησης και ύστερα υπολόγισε τα βάρη.

Βήμα 2: Βρες τους κοντινότερους γείτονες με βάση τα βάρη και υπολόγισε τις ομοιότητες.

Βήμα 3: Βρες τους κοντινότερους γείτονες με βάση την ολική βαθμολογία και υπολόγισε ομοιότητες.

Βήμα 4: Υπολόγισε την ένωση των συνόλων από το Βήμα 2 και το Βήμα 3.

Βήμα 5: Υπολόγισε την πρόβλεψη των βαθμολογιών.

3.6 Περιγραφή αλγορίθμου κατακερματισμού τοπικής ευαισθησίας (LSH)

Ο αλγόριθμος LSH μειώνει τη διάσταση πολυδιάστατων δεδομένων. Ο LSH κατακερματίζει ένα αντικείμενο με τέτοιο τρόπο ώστε παρόμοια αντικείμενα τοποθετούνται στον ίδιο κάδο με μεγάλη πιθανότητα, με τον αριθμό των κάδων να είναι πολύ μικρότερος από το σύνολο των δυνατών αντικειμένων προς είσοδο. Ο LSH διαφέρει από τις συμβατικές συναρτήσεις κατακερματισμού επειδή μεγιστοποιεί την πιθανότητα “σύγκρουσης” για παρόμοια αντικείμενα.

Στη συγκεκριμένη πτυχιακή χρειάστηκε να βρούμε παρόμοιους χρήστες οπότε χρειαζόμαστε ένα κριτήριο ομοιότητας. Για να το βρούμε έχουμε ακολουθήσει την εξής λογική. Αρχικά παράγουμε ένα διάνυσμα που αντιπροσωπεύει το χρήστη. Αυτό παράγεται από τον τύπο.

$$Διανυσμα_X = \sum_{i=0}^{πλήθοςαντικειμένωντουX} \frac{ολικήβαθμολογία}{επιμερουςβαθμολογία\ j_i} \quad , (3.14)$$

Όπου j ξεκινάει από την τιμή ένα διότι δεν θέλουμε να λάβουμε υπόψη το ολικό κριτήριο. Δηλαδή αθροίζουμε τα διανύσματα των βαθμολογιών του χρήστη αφού πρώτα έχουν διαιρεθεί με το ολικό κριτήριο και παράγεται το διάνυσμα του χρήστη X με διαστάσεις όσο το πλήθος των επιμέρους κριτηρίων. Έχουμε δηλαδή ένα διάνυσμα της μορφής $d_x = [d_1, d_2, d_{3,} d_4, d_5, d_6, d_7]$ αυτό το διάνυσμα αντιπροσωπεύει το τρόπο με τον οποίο βαθμολογεί ένας χρήστης. Τώρα χρειαζόμαστε έναν τρόπο να χαρτογραφήσουμε αυτά τα διανύσματα σε κάδους έτσι ώστε χρήστες με παρόμοια d να βρίσκονται στον ίδιο κάδο. Προκειμένου να το πετύχουμε αυτό χρησιμοποιούμαι την εξής μέθοδο κατακερματισμού η οποία είναι παρόμοια με αυτή που αναφέρεται και στο βιβλίο MultiProbe LSH: Efficient Indexing for HighDimensional Similarity Search [10] και προσαρμοσμένη ύστερα από πειράματα στο τύπο 3.15.

Αρχικοποιούμε ένα τυχαία παραγόμενο διάνυσμα επτά διαστάσεων με Γκαουσιανή (Gaussian) κατανομή και άθροισμα των στοιχείων του ίσο με ένα, ας το ονομάσουμε v. Έπειτα προβάλουμε το d πάνω σε αυτό το διάνυσμα και επιστρέφεται ένας πραγματικός αριθμός με τον εξής τρόπο:

$$αποτέλεσμα = \sum_{i=0}^{6} d_i v_i \quad , (3.15)$$

Από το αποτέλεσμα ανάλογα με το πλήθος των δεκαδικών ψηφίων που θα κρατήσουμε μπορούμε να αυξομειώσουμε την ευαισθησία του κατακερματισμού. Ύστερα από πειραματισμό κρατώντας δύο δεκαδικά προκύπτει αρκετά ικανοποιητικός κατακερματισμός. Οπότε αυτή η τιμή αντιπροσωπεύει το κλειδί για το κάδο στον οποίο ανήκει ο χρήστης. Έτσι έχουμε καταφέρει να βρούμε χρήστες οι οποίοι βαθμολογούν με παρόμοιο τρόπο, επειδή αυτοί θα ανήκουν στον ίδιο κάδο. Συμβαίνει όμως χρήστες που είναι όμοιοι να βρεθούν σε διαφορετικούς κάδους. Η λύση σε αυτό το πρόβλημα είναι να έχουμε περισσότερους πίνακες κατακερματισμού προκειμένου να είναι σχεδόν απίθανο όμοιοι χρήστες να μην βρεθούν στον ίδιο κάδο σε κάποιο πίνακα κατακερματισμού.

4. ΕΠΕΞΗΓΗΣΗ ΕΦΑΡΜΟΓΗΣ

4.1 Επεξήγηση της βάσης δεδομένων

Η βάση δεδομένων σχεδιάστηκε στο σύστημα διαχείρισης βάσεων δεδομένων MySQL.

Ο πίνακας users περιέχει τις στήλη Id όπου αποτελεί το κλειδί για την αναζήτηση χρήστη στο πίνακα και τη στήλη UserName η οποία αποθηκεύει σε τύπο VAR(45) της MySQL τα ονόματα των χρηστών.

Στο πίνακα με όνομα meta_data θα βρούμε τη στήλη Id η οποία χρησιμοποιείται για να αναγνωριστούν μοναδικά οι εγγραφές στο πίνακα, τη στήλη number_Users η οποία περιέχει το πλήθος των χρηστών στη βάση δεδομένων μας, τη στήλη number_Items η οποία περιέχει το πλήθος των αντικειμένων που υπάρχουν στη βάση, τη στήλη number_Ratings στην οποία φυλάσσεται το πλήθος των πολλαπλών κριτηρίων συν ένα για το ολικό και τη στήλη Max_Rating η οποία περιέχει τη τιμή της μέγιστης βαθμολογίας που μπορεί να δώσει ένα χρήστης σε κάποιο επιμέρους ή στο ολικό κριτήριο.

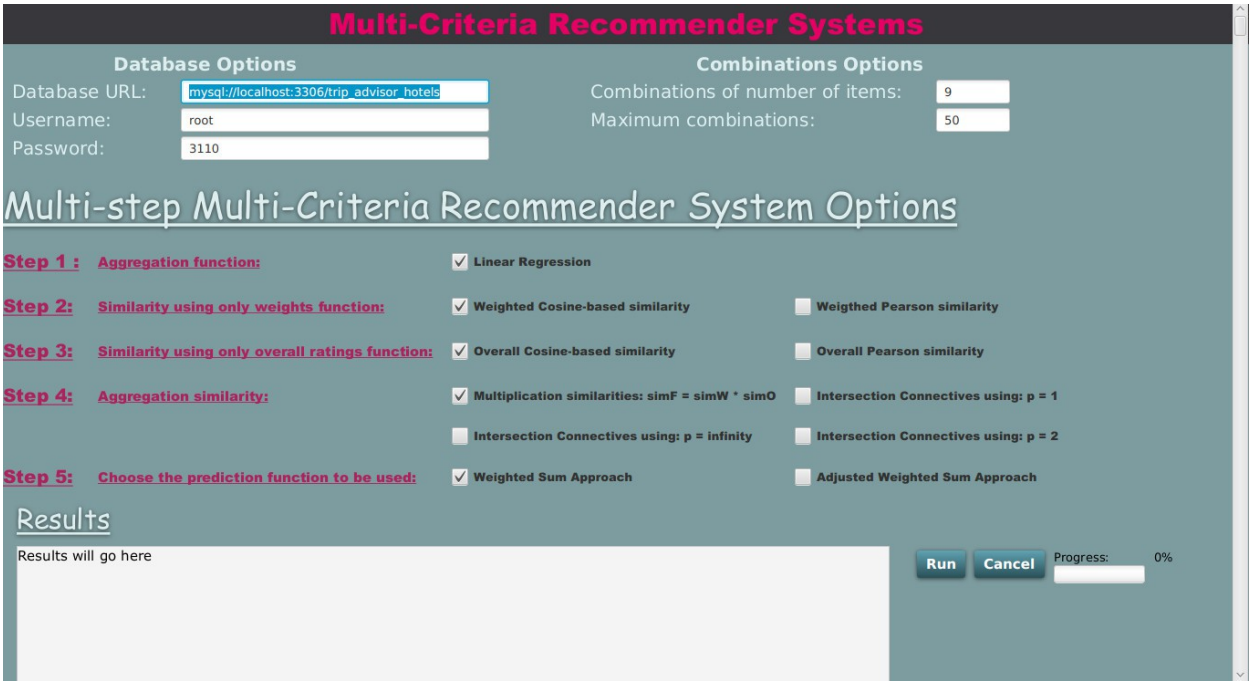
Ο πίνακας items περιέχει τη στήλη Id που παρέχει λειτουργικότητα ίδια με αυτή των ομώνυμων στηλών που περιγράφονται στις προηγούμενες παραγράφους και τη στήλη ItemName η οποία αποθηκεύει συμβολοσειρές που αντιπροσωπεύουν τα ονόματα των αντικειμένων.

Στο πίνακα users_items_ratings θα βρούμε τη στήλη Id που χρησιμοποιείται για τη μοναδική αναγνώριση της κάθε γραμμής στο πίνακα, τη στήλη UserID η οποία είναι ξένο κλειδί στο πεδίο Id του πίνακα users. Το πεδίο ItemID αποτελεί ξένο κλειδί στη στο πεδίο Id του πίνακα items. Τέλος, υπάρχουν τα πεδία Rating0 μέχρι Rating7 στα οποία αποθηκεύονται οι βαθμολογίες των χρηστών, με την ολική βαθμολογία να αποθηκεύεται στο Rating0 και τις επιμέρους στα πεδία Rating1 έως Rating7.

4.2 Περιγραφή γραφικής διεπαφής

Το πρόγραμμα παρέχει μία γραφική διεπαφή για την αλληλεπίδραση με το χρήστη. Παρέχει κάποιες επιλογές που μπορεί να χρησιμοποιήσει. Έχει αναπτυχθεί σε γλώσσα Java 8 (jdk 8u10) χρησιμοποιώντας την JavaFX 2.0 πλατφόρμα και για την ανάπτυξη χρησιμοποιήθηκε το Eclipse IDE. Παρακάτω φαίνεται η γραφική του παρουσίαση και αμέσως μετά επεξηγούνται τα διάφορα πεδία.

Εικόνα 1: Γραφική παρουσίαση διεπαφής

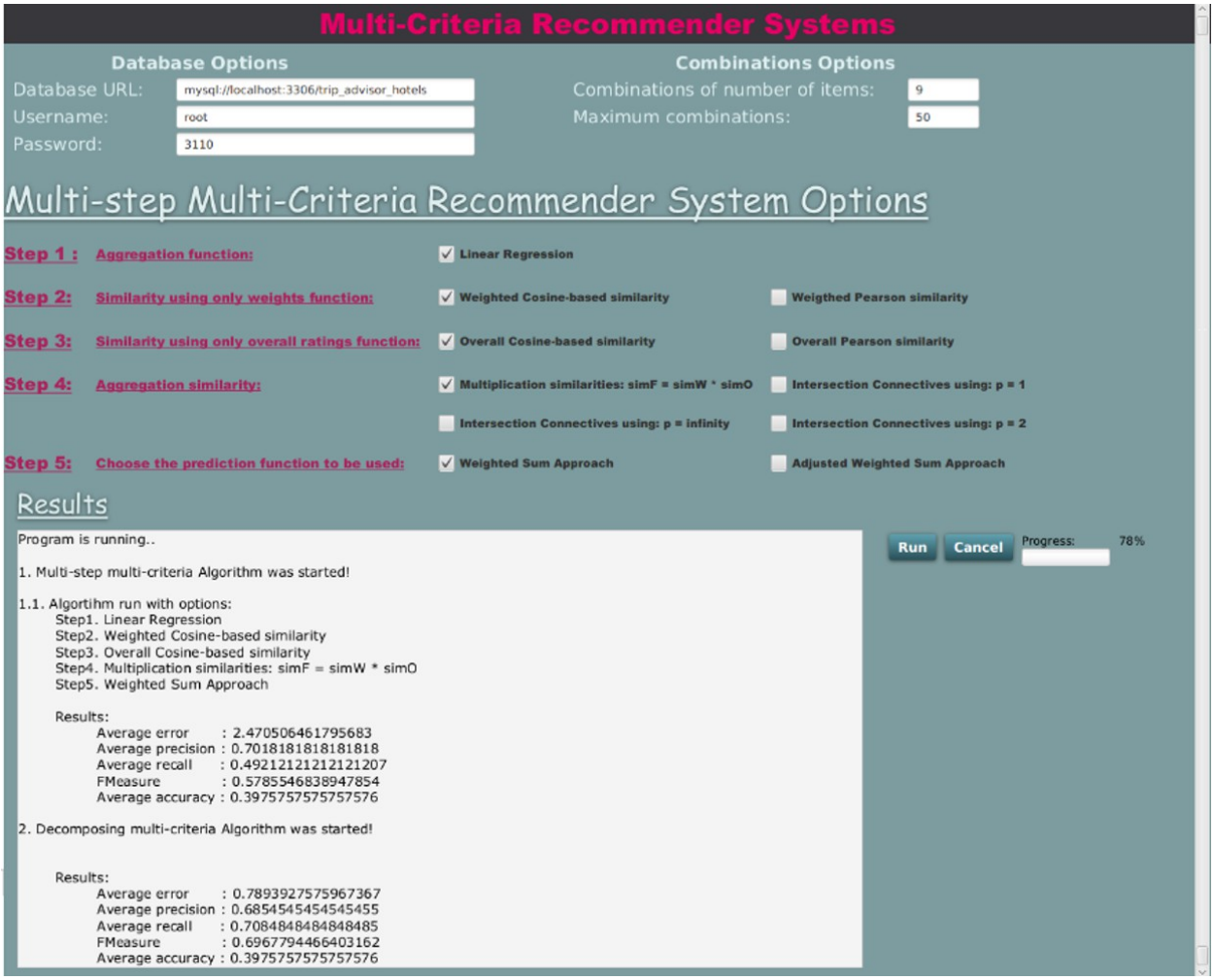


Ξεκινώντας από το Database Options μπορεί κανείς να εισάγει τον υπερσύνδεσμο όπου βρίσκεται η βάση δεδομένων στο χώρο που δίνεται δεξιά της συμβολοσειράς Database URL. Δεξιά από τη συμβολοσειρά Username εισάγεται το όνομα χρήστη για να γίνει η σύνδεση στη βάση δεδομένων και κάτω δεξιά της συμβολοσειράς Password εισάγεται ο κωδικός που απαιτείται για την σύνδεση.

Στις επιλογές κάτω από το Combination Options παρέχεται η δυνατότητα στο χρήστη να εισάγει το Combination of number of items. Αυτή η τιμή θα χρησιμοποιηθεί για να επιλέξουμε το πλήθος των αντικειμένων από τα οποία θα πάρουμε δυνατούς συνδυασμούς από το σύνολο των αντικειμένων που έχουν βαθμολογήσει οι χρήστες. Με το πεδίο Maximum Combinations μπορούμε να επιλέξουμε το μέγιστο αριθμό από το πλήθος των δυνατών συνδυασμών που μπορούν να δημιουργηθούν από τα αντικείμενα που έχει βαθμολογήσει ένας χρήστης. Δηλαδή εάν το πλήθος των δυνατών συνδυασμών με 9 αντικείμενα από τα αντικείμενα που έχει βαθμολογήσει ένας χρήστης είναι 100 στο στιγμιότυπο που φαίνεται στην εικόνα 2 εμείς θα πάρουμε τα 50 από αυτά. Για να τρέξει η εφαρμογή με όλους αυτούς τους δυνατούς συνδυασμούς μπορούμε να το κάνουμε δίνοντας μια αρνητική τιμή στο πεδίο αυτό.

Κάτω από τη συμβολοσειρά Multi-step Multi-Criteria Recommender System Options παρέχονται επιλογές για τον τρόπο με τον οποίο επιθυμεί κάποιος να υπολογιστούν οι προβλέψεις. Εάν σε κάποιο βήμα δοθούν παραπάνω από μία επιλογές η εφαρμογή θα εκτελέσει όλους τους δυνατούς συνδυασμούς αυτών. Για παράδειγμα αν στο βήμα 2 επιλεχθεί το Weighted Cosine-based similarity και το Weighted Pearson similarity τότε η εφαρμογή θα εκτελεστεί μια φορά με την επιλογή Weighted Cosine-based similarity και δεύτερη φορά με την επιλογή Weighted Pearson similarity. Εάν επιλεγούν δύο τρόποι υπολογισμού και σε κάποιο άλλο βήμα τότε οι συνολικές εκτελέσεις θα γίνουν τέσσερις, μια για κάθε δυνατό συνδυασμό κ.ο.κ.

Μπορούμε να δούμε μια εκτέλεση της εφαρμογής παρακάτω.



Εικόνα 2: Παράδειγμα εκτέλεσης

4.3 Αναλυτική περιγραφή του κώδικα της εφαρμογής

Για να εκτελεστούν οι αλγόριθμοι που περιγράψαμε στη δεύτερη ενότητα χρειαζόμαστε τις βαθμολογίες των χρηστών. Προκειμένου να τις αποκτήσουμε είναι απαραίτητο πρώτα να εκτελεστεί η συνάρτηση CreateDataset() της κλάσης Dataset που βρίσκεται στο πακέτο rs.logic.staticvariables, της οποίας η λειτουργικότητα είναι να ανακτήσει τα δεδομένα από τη βάση και να τα αποθηκεύσει στο τρισδιάστατο πίνακα που αναφέραμε στη παράγραφο 2.1. Όσοι χρήστες δεν έχουν βαθμολογήσει τα επιμέρους κριτήρια σε κάποιο αντικείμενο, δηλαδή η τιμή του επιμέρους κριτηρίου είναι μείον ένα, τότε αντικαθιστούμε τα μείον ένα με τη τιμή που έχει το ολικό κριτήριο σε αυτό το αντικείμενο για αυτό το χρήστη.

Επειδή στον τρισδιάστατο πίνακα υπάρχουν αντικείμενα τα οποία δεν έχουν βαθμολογηθεί από τον εκάστοτε χρήστη χρειαζόμαστε ένα τρόπο να αναγνωρίζουμε ποια είναι αυτά που έχει όντως βαθμολογήσει ο κάθε χρήστης. Επομένως δημιουργούνται λίστες όπου η κάθε μία αντιστοιχεί σε ένα χρήστη και ως περιεχόμενο τους έχουν μόνο τα αντικείμενα τα οποία έχει βαθμολογήσει αυτός ο χρήστης. Αυτή είναι η λίστα με όνομα “list” στο αρχείο RecommenderSystem.java στο πακέτο rs.logic.algorithms.

Λαμβάνοντας υπόψη τις επιλογές που έχει κάνει ο χειριστής της γραφικής διεπαφής, για

τον τρόπο υπολογισμού των προβλέψεων, επιλέγονται οι αντίστοιχοι συνδυασμοί τους. Αρχικά παράγεται ένα καινούργιο dataset στο οποίο αντιγράφονται οι τιμές του αρχικού dataset. Ωστόσο, για το χρήστη για τον οποίο θέλουμε να παράγουμε προτάσεις δεν κρατάμε όλες τις βαθμολογίες του στα πλαίσια του πειράματος αξιολόγησης. Θα κρατήσουμε ένα συνδυασμό k αντικειμένων αυτού του χρήστη. Αυτή η μέθοδος ακολουθείται διότι θέλουμε να βρούμε συσχέτιση του χρήστη προς εξέταση ως προς τους υπόλοιπους χρήστες στη βάση έτσι ώστε να μπορούμε να παράγουμε τις προτάσεις μας. Το k αντιπροσωπεύει τον αριθμό των αντικειμένων που πρόκειται να διατηρήσουμε στο καινούριο dataset στο οποίο περιέχονται μόνο αντικείμενα που έχουν βαθμολογηθεί από τον εκάστοτε χρήστη. Τιμή στο k δίνεται μέσω της γραφικής διεπαφής στο πεδίο “Combination of number of items”. Αξίζει σε αυτό το σημείο να επισημάνουμε μία διαφορά με την προηγούμενη πτυχιακή η οποία είναι αποτέλεσμα του περιορισμού που υπήρχε λόγω της γραμμικής παλινδρόμησης. Η διαφορά αυτή είναι ότι πριν, το πλήθος από τα βαθμολογημένα αντικείμενα του εξεταζόμενου χρήστη έπρεπε να είναι μεγαλύτερο από το k γιατί αλλιώς θα εξετάζαμε, σαν συνδυασμό αντικειμένων, το κενό σύνολο. Δηλαδή, αν ένας χρήστης έχει βαθμολογήσει 3 αντικείμενα και το k είναι 4, τότε δεν μπορούμε να τοποθετήσουμε 4 αντικείμενα στο καινούργιο dataset. Στην συγκεκριμένη πτυχιακή όμως, όπως αναφέραμε στο κεφάλαιο 3.1, οι χρήστες μας “δανείζονται” βαθμολογίες από όμοιους χρήστες και έτσι ο αριθμός των βαθμολογημένων αντικειμένων μπορεί να γίνει μικρότερος από το k.

Η παραγωγή των προτάσεων για κάθε χρήστη θα εκτελεστεί για κάθε συνδυασμό αντικειμένων που έχει βαθμολογήσει ο εκάστοτε χρήστης. Το πλήθος αυτών των συνδυασμών είναι διαφορετικό για κάθε χρήστη αφού εξαρτάται από τον αριθμό των αντικειμένων τα οποία έχει βαθμολογήσει (n). Η παραγωγή προτάσεων για κάθε χρήστη θα γίνει για n βαθμολογημένα αντικείμενα ανά το πλήθος των επιθυμητών βαθμολογημένων αντικειμένων (k) χωρίς επανάληψη. Επομένως το πλήθος των συνδυασμών είναι n!/(k!(n-k)!) το οποίο εκφράζει όλους τους δυνατούς συνδυασμούς μεγέθους k βαθμολογημένων αντικειμένων από n βαθμολογημένα αντικείμενα. Το k αρχικοποιείται από το χειριστή της γραφικής διεπαφής στο πεδίο “Combinations of number of items” και παραμένει σταθερό κατά την εκτέλεση. Το πλήθος των παραγόμενων προτάσεων εξαρτάται από τη τιμή του n η οποία δεν είναι γνωστή εκ των προτέρων και μπορεί να είναι διαφορετική για κάθε χρήστη. Έπεται ότι όσο περισσότερα αντικείμενα έχουν βαθμολογήσει οι χρήστες τόσο αυξάνεται ο χρόνος εκτέλεσης της εφαρμογής. Προκειμένου να αντιμετωπίσουμε αυτό το πρόβλημα χρησιμοποιούμε τη τιμή του πεδίου “Maximum combinations” το οποίο καθορίζει το μέγιστο αριθμό των συνδυασμών των βαθμολογημένων αντικειμένων που θα λάβουμε υπόψη κατά τον υπολογισμό για τον εκάστοτε χρήστη.

Για να δούμε σε ένα παράδειγμα τη χρήση αυτών τον μεταβλητών ας υποθέσουμε βλέπουμε πως δύο χρήστες έχουν βαθμολογήσει τα αντικείμενα {1,2,3} ο πρώτος και {4,5,6,7} ο δεύτερος. Υποθέτω ότι δίνω στην εφαρμογή στο πεδίο “Combinations of number of items”, δηλαδή το k, την τιμή 3, επίσης επιλέγω να δώσω στο πεδίο “Maximum combinations” την τιμή 1. Μπορούμε να υπολογίσουμε το πλήθος των δυνατών συνδυασμών των αντικειμένων του πρώτου χρήστη από τον τύπο 3!/(3!(3-3!)) το οποίο ισούται με 1. Για τον δεύτερο χρήστη υπολογίζουμε πως το πλήθος των δυνατών συνδυασμών του με 3 αντικείμενα είναι 4, το οποίο είναι αποτέλεσμα της πράξης 4!/(3!(4-3)!). Επιλέξαμε όμως το “Maximum combinations” να είναι 1, οπότε μόνο ένας συνδυασμός αντικειμένων, από τους τέσσερις, του δεύτερου χρήστη θα

εισαχθούν στο νέο dataset. Ο πρώτος χρήστης έχει μόνο ένα συνδυασμό αντικειμένων οπότε αυτός ο συνδυασμός θα εισαχθεί στο νέο dataset.

Εν κατακλείδι εκτελούνται όλα τα βήματα και παράγεται ο μέσος όρος του κάθε χρήστη ως προς όλους του πιθανούς συνδυασμούς των αντικειμένων του. Η διαδικασία αυτή πραγματοποιείται για όλους τους συνδυασμούς των επιλογών που έχει κάνει ο χειριστής της εφαρμογής στα βήματα ένα μέχρι πέντε. Τα τελικά αποτελέσματα των προβλέψεων αποθηκεύονται σε αρχεία στο φάκελο με όνομα Result. Οι τελικοί μέσοι όροι των μέσων σφαλμάτων αποθηκεύονται στο αρχείο average_errors.txt στον κατάλογο Result. Τα αρχεία με τις προβλέψεις αποθηκεύονται στα αρχεία με όνομα του τύπου α1_α2_α3_α4_α5.txt, το α_i εκπροσωπεί τον αριθμό επιλογής στις επιλογές του κάθε βήματος στη γραφική διεπαφή και ξεκινάνε από το ένα.

5. ΜΕΤΡΙΚΕΣ ΑΞΙΟΛΟΓΗΣΗΣ

Στο κεφάλαιο αυτό θα δούμε πως μπορούμε να αξιολογήσουμε την αποδόση και την επίδοση των αλγόριθμων που εξετάσαμε. Για τον λόγο αυτό θα πρέπει να χρησιμοποιηθούν ορισμένες μετρικές αξιολόγησης όπως είναι η ακρίβεια, η ανάκληση, η ορθότητα και το απόλυτο σφάλμα. Ως ακρίβεια (precision) θα εκφράσουμε την πιθανότητα μία πρόταση που παράχθηκε από το σύστημα να ικανοποιεί τις προτιμήσεις του χρήστη. Ως ανάκληση (recall) θα εκφράσουμε την πιθανότητα κάποια αντικείμενα να "αρέσουν" στους χρήστες, δηλαδή να ικανοποιεί τις προτιμήσεις του, όντως να προταθεί από το σύστημα. Ως ορθότητα (accuracy) θα εκφράσουμε το κατά πόσον ορθά έγινε η πρόβλεψη της βαθμολογίας για ένα αντικείμενο σε σχέση με τις προτιμήσεις των χρηστών, δηλαδή υπολογίζει πόσα αντικείμενα από αυτά τα οποία πρότεινε το σύστημα όντως αρέσουν στους χρήστες και πόσα αντικείμενα δεν πρότεινε όντως δεν ικανοποιούν τις προτιμήσεις των χρηστών. Τέλος το μέσο απόλυτο σφάλμα εκφράζει το μέσο συνολικό σφάλμα της πρόβλεψης σε σχέση με τα πραγματικά δεδομένα.

Ένα πολύ σημαντικό πράγμα που πρέπει να αναφέρουμε είναι το πως οριοθετούμε την τιμή της βαθμολογίας η οποία θα είναι ο δείκτης μας για να προτείνει το σύστημα ένα αντικείμενο στον χρήστη. Στον αλγόριθμο αυτό έχουμε επιλέξει το όριο αυτό να είναι το 70% της μέγιστης βαθμολογίας έτσι ώστε να μπορούμε να αξιολογήσουμε καλύτερα τους αλγόριθμους. Για τις πιο υψηλές απαιτήσεις των χρηστών σαφώς και θα ήταν καλύτερα το ποσοστό αυτό να ήταν πιο υψηλό και για τον λόγο αυτό έχουμε εξετάσει αποτελέσματα με το όριο αυτό να είναι 80% και 90%.

Σύμφωνα με τα παραπάνω μία πρόβλεψη για ένα αντικείμενο μπορούμε να την αντιστοιχίσουμε σε μία από τις 4 εξής κατηγορίες :

1. Ένα αντικείμενο να έχει προταθεί από το σύστημα και να ικανοποιεί τις προτιμήσεις του χρήστη.
2. Ένα αντικείμενο να έχει προταθεί από το σύστημα αλλά να μην ικανοποιεί τις προτιμήσεις του χρήστη.
3. Ένα αντικείμενο να μην έχει προταθεί από το σύστημα και να ικανοποιεί τις προτιμήσεις του χρήστη.
4. Ένα αντικείμενο να μην έχει προταθεί από το σύστημα και να μην ικανοποιεί τις προτιμήσεις του χρήστη.

Μπορούμε εύκολα τώρα με βάση τις κατηγορίες αυτές να εξηγήσουμε με μαθηματικούς τύπους τις μονάδες μετρήσεων . Θα πρέπει δηλαδή να κρατήσουμε τέσσερις μετρητές που αντιστοιχούν σε αυτές τις κατηγορίες. Έστω :

- Nrs : Το πλήθος των αντικειμένων που ανήκουν στην κατηγορία 1.
- Nis : Το πλήθος των αντικειμένων που ανήκουν στην κατηγορία 2.
- Nrn : Το πλήθος των αντικειμένων που ανήκουν στην κατηγορία 3.
- Nin : Το πλήθος των αντικειμένων που ανήκουν στην κατηγορία 4.

5.1 Μέσο Απόλυτο Σφάλμα (Mean absolute error, MAE)

Με το μέσο απόλυτο σφάλμα μετράμε πόσο κοντά έπεσαν οι προβλέψεις του συστήματος σε σχέση με τις πραγματικές βαθμολογίες. Ο τύπος που το περιγράφει είναι

$$MAE = \frac{\left(\sum \left| \left(r_i - p_i \right) \right| \right)}{N} \quad , (5.1)$$

Όπου r_i η πραγματική βαθμολογία που έχει δώσει ο χρήστης και

p_i η βαθμολογία που πρόβλεψε το σύστημα, το N είναι ο αριθμός των αντικειμένων για τα οποία κάναμε τις προβλέψεις.

5.2 Ακρίβεια

Εδώ μετράμε κατά πόσο το σύστημα πρόβλεψε σωστά σύμφωνα με τις ανάγκες ενός χρήστη. Ο τύπος της ακρίβειας είναι

$$P = \frac{Nrs}{\left(Nrs + Nis \right)} \quad , (5.2)$$

Για παράδειγμα αν το σύστημα έκανε πρόβλεψη για 10 αντικείμενα αλλά μόνο τα 7 ικανοποιούν τις προτιμήσεις του χρήστη τότε η ακρίβεια είναι 70%.

5.3 Ανάκληση

Εδώ μετράμε κατά τα πόσο τα αντικείμενα που ικανοποιούν τις ανάγκες κάποιου χρήστη όντως προτάθηκαν από το σύστημα. Έχουμε δηλαδή

$$R = \frac{Nrs}{\left(Nrs + Nrn \right)} \quad , (5.3)$$

Για παράδειγμα αν για κάποιον χρήστη ικανοποιούν τις ανάγκες του 10 αντικείμενα και το σύστημα του πρότεινε 8 από αυτά τότε η ανάκληση είναι 80%.

5.4 Ορθότητα

Εδώ μετράμε κατά πόσο το σύστημα αποφάσισε σωστά αν ένα αντικείμενο μπορεί να προταθεί ή όχι σε κάποιο χρήστη. Η ορθότητα υπολογίζεται όπως φαίνεται παρακάτω.

$$A = \frac{\left(Nrs + Nin \right)}{\left(Nrs + Nis + Nrn + Nin \right)} \quad , (5.4)$$

5.5 Σταθμισμένος Αρμονικός Μέσος

Με αυτή την μέθοδο ουσιαστικά μετράμε την συσχέτιση που έχει η ακρίβεια με την ανάκληση. Ο στόχος ενός συστήματος προτάσεων είναι να έχει αυτά τα δύο ποσοστά όσον το δυνατό μεγαλύτερα. Έχει παρατηρηθεί ότι όσο μεγαλύτερο πλήθος από αντικείμενα έχουμε για να προτείνουμε στους χρήστες τόσο αυξάνεται το ποσοστό της ανάκλησης, μειώνεται όμως το ποσοστό της ακρίβειας. Αντιθέτως αν έχουμε μικρό όγκο δεδομένων στα αντικείμενα τότε παρατηρείται το αντίθετο. Μπορούμε λοιπόν να υποθέσουμε ότι με βάση το πλήθος των αντικειμένων η σχέση της ακρίβειας με την ανάκληση είναι αντιστρόφως ανάλογη.

Για τον λόγο αυτό χρησιμοποιούμε τον σταθμισμένο αρμονικό μέσο (F-Measure), με τον οποίο μπορούμε να μετρήσουμε αυτήν την σχέση. Η μέθοδος υπολογισμού δίνεται από τον εξής τύπο[11]

$$F_{Measure} = \left(1 + b^2\right) \frac{PR}{\left(b * P + R\right)} \quad , (5.5)$$

όπου το b είναι ένας συντελεστής βαρύτητας ανάμεσα στην ακρίβεια P και την ανάκληση R. Εμείς θεωρούμε ότι οι δύο μετρικές έχουν τον ίδιο σημαντικό ρόλο στις μετρήσεις μας και για τον λόγο αυτό θέτουμε b = 1.

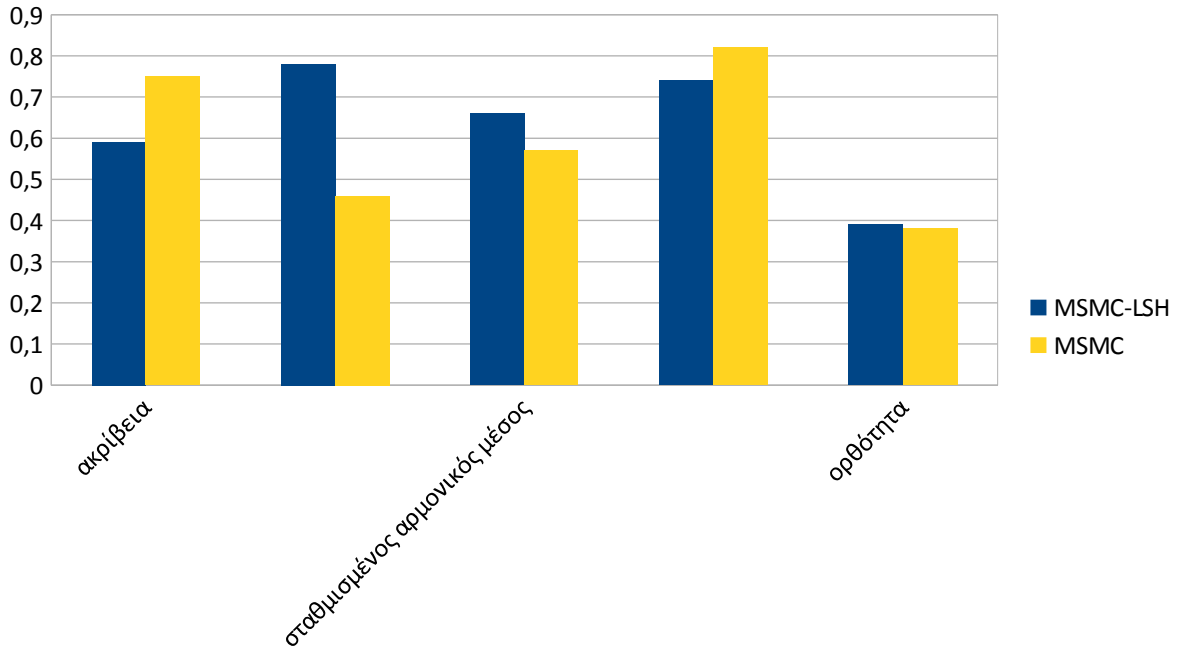
6. ΑΠΟΤΕΛΕΣΜΑΤΑ ΑΛΓΟΡΙΘΜΩΝ

Για να αξιολογήσουμε τα αποτελέσματα θα πρέπει να εξετάσουμε τους αλγόριθμους του συστήματος ως προς την απόδοσή τους και ως προς την επίδοσή τους. Η απόδοση έχει να κάνει με το πόσο πετυχημένες είναι οι αποφάσεις που παίρνει το σύστημα για πρόταση αντικειμένων στους χρήστες, ανάλογα με τις προτιμήσεις τους. Η επίδοση έχει να κάνει με την ταχύτητα του συστήματος για να πάρει μία απόφαση, δηλαδή με το πόσο γρήγορα εκτελούνται οι αλγόριθμοι που χρησιμοποιεί. Παρακάτω θα αναλύσουμε αυτούς τους τρόπους αξιολόγησης συγκρίνοντας τον αλγόριθμο που εξετάσαμε σε αυτήν την πτυχιακή αρχικά με τον προηγούμενο αλγόριθμο(MSMC) αλλά και σε σχέση με τους ήδη γνωστούς αλγορίθμους(DMC,WSO) που έχουμε αναφέρει.

6.1 Πλεονεκτήματα και μειονεκτήματα αλγορίθμου Multi-Step Multi-Criteria with LSH σε σχέση με τον αλγόριθμο Multi-Step Multi-Criteria

Ο αλγόριθμος που επεκτάθηκε σε αυτήν την πτυχιακή χρησιμοποιεί νέες τεχνικές και μεθόδους στις οποίες πρέπει να επισημάνουμε τα πλεονεκτήματα και τα μειονεκτήματα που επέφεραν σε σχέση με τον προηγούμενο αλγόριθμο.

Στο σχήμα 1 φαίνονται οι μετρικές αξιολόγησης για τον νέο (MSMC-LSH) και τον αρχικό (MSMC) αλγόριθμο.



Σχήμα 1: Σύγκριση απόδοσης των αλγορίθμων MSMC-LSH και MSMC.

Ο νέος αλγόριθμος όπως παρατηρούμε έχει μία άνοδο στο σταθμισμένο αρμονικό μέσο, ο οποίος όπως έχουμε περιγράψει συνδυάζει την ακρίβεια με την ανάκληση. Ένα πλεονέκτημα αυτού του αλγορίθμου είναι ότι εξετάζει πιο πολλά ενδεχόμενα από τον προηγούμενο. Για να το εξηγήσουμε αυτό θα πρέπει να κάνουμε μία αναφορά στο κεφάλαιο 3. Στον προηγούμενο αλγόριθμο είχαμε πρόβλημα όταν εφαρμόζαμε την μέθοδο της γραμμικής παλινδρόμησης για να βρούμε τι βάρος δίνει ο χρήστης σε κάθε

επιμέρους βαθμολογία. Για τον λόγο αυτό δεν μπορούσαμε να λάβουμε υπόψη όλους τους συνδυασμούς αντικειμένων που είχε βαθμολογήσει ένας χρήστης, ο οποίος είχε βαθμολογήσει λίγα αντικείμενα, με αποτέλεσμα να παίρνουμε ένα μικρό δείγμα από το προφίλ του για να κάνουμε πρόβλεψη. Για παράδειγμα ένας χρήστης που έχει βαθμολογήσει εννιά αντικείμενα θα έπρεπε να εξετάσουμε μόνο έναν συνδυασμό, αφού ο αλγόριθμος της γραμμικής παλινδρόμησης περιόριζε τον χρήστη να έχει βαθμολογήσει τουλάχιστον δύο παραπάνω αντικείμενα από των αριθμό των επιμέρους κριτηρίων, που στην περίπτωσή μας είναι εφτά.

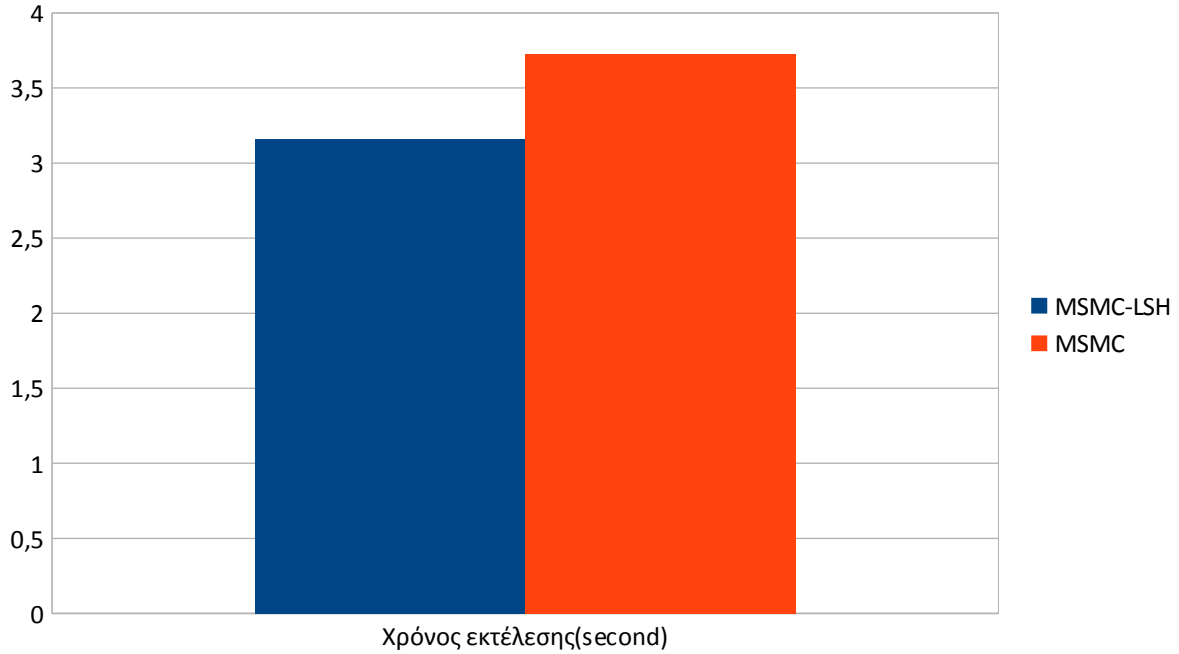
Η τεχνική που εφαρμόστηκε για την επίλυση αυτού του προβλήματος παρατηρούμε ότι μας δίνει καλύτερη ανάκληση αλλά όχι καλύτερη ακρίβεια, δηλαδή αυξάνεται το ποσοστό οι προτιμήσεις των χρηστών να ικανοποιούνται από τις συγκεκριμένες προτάσεις, αλλά αυξάνεται και το ποσοστό οι προτάσεις αυτές να αφορούν αντικείμενα που δεν ικανοποιούν τις προτιμήσεις των χρηστών. Θα μπορούσαμε δηλαδή να πούμε ότι το σύστημα προτείνει παραπάνω αντικείμενα με πριν, έχοντας στην διάθεσή του την βαρύτητα που δίνει ο χρήστης σε κάθε κριτήριο ενός αντικειμένου. Το μειονέκτημα όμως είναι η πτώση της ακρίβειας, η οποία οφείλεται κυρίως στο γεγονός ότι ο αλγόριθμος με την χρήση του LSH έχει γίνει πιο ευριστικός. Για παράδειγμα, αν ένας χρήστης δεν έχει αρκετές αξιολογήσεις, τότε εκτιμάται ότι αυτός ο χρήστης θα βαθμολογούσε όπως ένας άλλος με τον οποίον έχουν παρόμοια συμπεριφορά ως προς τις προτιμήσεις τους.

Επίσης αξίζει να σημειώσουμε ότι στον MSMC-LSH έχει μειωθεί το μέσο απόλυτο σφάλμα. Αν και ο παλιός αλγόριθμος εξέταζε όλες τις ομοιότητες μεταξύ των χρηστών θα μπορούσε να πει κανείς ότι το σφάλμα της πρόβλεψης που θα περίμενε, θα ήταν μικρότερο. Αυτό δεν συμβαίνει διότι οι χρήστες που λαμβάνουμε υπόψη για να προβλέψουμε τις βαθμολογίες ενός χρήστη είναι μόνο οι πραγματικά πολύ παρόμοιοι, με αποτέλεσμα να αγνοούμε χρήστες οι οποίοι δεν έχουν κοινό προφίλ με τον χρήστη που εξετάζουμε και μπορούν έτσι να αλλοιώσουν το τελικό αποτέλεσμα. Επίσης όπως έχουμε αναφέρει, για να κάνουμε πρόβλεψη εκτός από τις αξιολογήσεις των χρηστών λαμβάνουμε υπόψη και την βαρύτητα που δίνουν σε κάθε επιμέρους κριτήριο. Το μειονέκτημα του προηγούμενου αλγορίθμου ήταν ότι η γραμμική παλινδρόμηση αγνοούσε αξιολογήσεις που δεν ήταν γραμμικά ανεξάρτητες μεταξύ τους. Άρα μειωνόταν το πλήθος των αξιολογήσεων με αποτέλεσμα αν δεν ίσχυε ο περιορισμός της γραμμικής παλινδρόμησης να επέστρεφε λανθασμένα αποτελέσματα που στην ουσία ήταν ένα μηδενικό διάνυσμα από βάρη. Συνεπώς, αυτά τα βάρη όταν λαμβάνονταν στην πρόβλεψη των αξιολογήσεων το αποτέλεσμα επέφερε μεγαλύτερο σφάλμα.

Θα πρέπει επίσης να εστιάσουμε και στις επιδόσεις των αλγορίθμων. Ο MSMC-LSH επιβαρύνεται στο πρώτο βήμα του αλγορίθμου γιατί όπως έχουμε αναφέρει χρησιμοποιείται ο αλγόριθμος LSH σε περίπτωση που δεν ικανοποιούνται οι περιορισμοί της γραμμικής παλινδρόμησης. Αυτό το κόστος δεν υπάρχει στον προηγούμενο αλγόριθμο. Στα επόμενα βήματα, που ουσιαστικά ασχολούμαστε με την εύρεση ομοιοτήτων μεταξύ χρηστών, η διαφορά των δύο αλγορίθμων είναι ότι ο MSMC έχει τετραγωνική πολυπλοκότητα σε σχέση με τον αριθμό των χρηστών ενώ ο MSMC-LSH ξαναχρησιμοποιεί τον LSH για την εύρεση κοντινότερων γειτόνων, με αποτέλεσμα η πολυπλοκότητα χρόνου να μειώνεται αφού δεν χρειάζεται να εξετάσουμε όλους τους χρήστες.

Στην δικιά μας περίπτωση, η βάση δεδομένων είχε να εξετάσει ένα μικρό πλήθος από χρήστες, με αποτέλεσμα η διαφορά χρόνου υπέρ του καινούργιου αλγόριθμου να μην

είναι εμφανής (σχήμα 2). Να επισημάνουμε επίσης, ότι το κόστος στο πρώτο βήμα, όπου αλγόριθμος MSMC-LSH θα εξετάσει παραπάνω περιπτώσεις αν δεν πληρούνται οι περιορισμοί της γραμμικής παλινδρόμησης αλλά και η ένταξη των χρηστών στους κάδους δεν συνυπολογίζονται στην παράμετρο του χρόνου διότι αυτό που μας ενδιαφέρει είναι ο υπολογισμός ομοιοτήτων μεταξύ των χρηστών με σκοπό την πρόβλεψη της αξιολόγησης των αντικειμένων. Δηλαδή δεν λαμβάνουμε υπόψη την προ-επεξεργασία που γίνεται με σκοπό την εύρεση ομοιοτήτων.



Σχήμα 2: Διάγραμμα μετρήσεων επίδοσης των αλγορίθμων

6.2 Σύγκριση απόδοσης αλγορίθμου Multi-Step Multi-Criteria με LSH σε σχέση με τους ήδη γνωστούς, Decomposing Multi Criteria και Weighted Slope One

Όσον αφορά την απόδοση υπάρχουν αρκετές παρατηρήσεις που μπορούμε να κάνουμε διότι έχουμε μεταβολές ανάλογα με τον τρόπο που θα αποφασίσουμε να δουλέψει το σύστημα. Αρχικά θα πρέπει να ορίσουμε πιο είναι το όριο της βαθμολογίας στο οποίο θα αποφασίσουμε αν πρέπει να προτείνουμε ένα αντικείμενο σε κάποιον χρήστη. Η τιμή αυτής της μεταβλητής που έχουμε δοκιμάσει στο σύστημα, είναι με 70%, 80% και 90% της μέγιστης βαθμολογίας. Όσον αυξάνεται το ποσοστό αυτό, τόσο πιο αυξημένες και απαιτητικές είναι οι προτιμήσεις του χρήστη, άρα και το σύστημα θα πρέπει να του προσφέρει πιο επιλεγμένα αντικείμενα. Για παράδειγμα αν ορίσουμε το όριο αυτό στο 90% της μέγιστης βαθμολογίας τότε θα προτείνουμε αντικείμενα στον χρήστη τα οποία το σύστημα θα έχει προβλέψει τουλάχιστον 4.5 (αν η μέγιστη βαθμολογία είναι 5). Σύμφωνα με αυτό μπορούμε να συμπεράνουμε ότι αν το ποσοστό αυτό μειώνεται, τότε περιορίζονται οι απαιτήσεις του χρήστη και επομένως το πλήθος αντικειμένων που μπορούμε να του προτείνουμε είναι μεγαλύτερο.

6.2.1 Σύγκριση με βάση το Σταθμισμένο Αρμονικό Μέσο και το Μέσο Απόλυτο Σφάλμα

Θα επικεντρωθούμε περισσότερο στον σταθμισμένο αρμονικό μέσο διότι είναι αυτό το

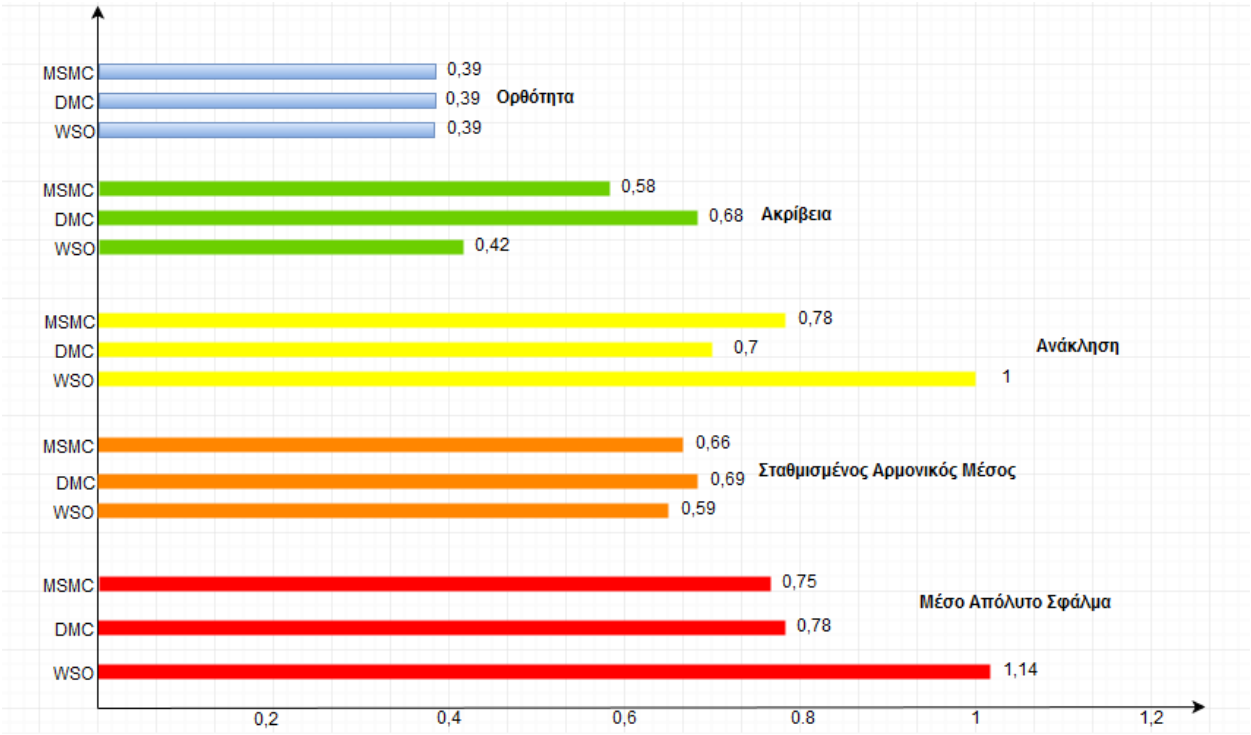
οποίο ένα σύστημα θέλει να μεγιστοποιεί για να έχει την ακρίβεια και την ανάκληση σε αρκετά σταθμισμένο βαθμό εφόσον οι δύο αυτές μετρικές είναι αντιστρόφως ανάλογες μεταξύ τους. Και ο λόγος που δεν θέλουμε να έχουμε ή την μία ή την άλλη σε πολύ μικρό βαθμό είναι διότι αν έχουμε πολύ μεγάλη ακρίβεια (και έχουμε μικρή ανάκληση) τότε αυτό σημαίνει ότι το σύστημα προτείνει ένα σύνολο από αντικείμενα τα οποία ικανοποιούν τις προτιμήσεις του χρήστη αλλά αυτά δεν είναι όλα τα αντικείμενα που είναι στις προτιμήσεις του χρήστη. Αν ισχύει το αντίθετο, δηλαδή μεγάλη ανάκληση και μικρή ακρίβεια, σημαίνει ότι για ένα σύνολο αντικειμένων που είναι στις προτιμήσεις του χρήστη, το σύστημα πρόβλεψε σωστά γι' αυτά τα αντικείμενα και όντως θα του προτείνει, το μειονέκτημα όμως είναι ότι θα του προτείνει και άλλα αντικείμενα τα οποία δεν βρίσκονται στην λίστα με τις προτιμήσεις του.

Με βάση το σχήμα 3 θα παρατηρήσουμε ότι ο Decomposing Multi Criteria είναι ο πιο αποδοτικός με βάση το F-Measure , και οριακά πιο πίσω βρίσκεται ο Multi Step Multi Criteria with LSH και τελευταίο το Weighted Slope One το οποίο παρουσιάζει την καλύτερη ανάκληση σε σχέση με τους προηγούμενους αλλά έχει πολύ μικρή ακρίβεια. Η χαμηλή ακρίβεια κυρίως οφείλεται διότι οι προβλέψεις έχουν γίνει με βάση μόνο την ολική βαθμολογία χωρίς να λαμβάνονται καθόλου υπόψη τα επιμέρους κριτήρια.

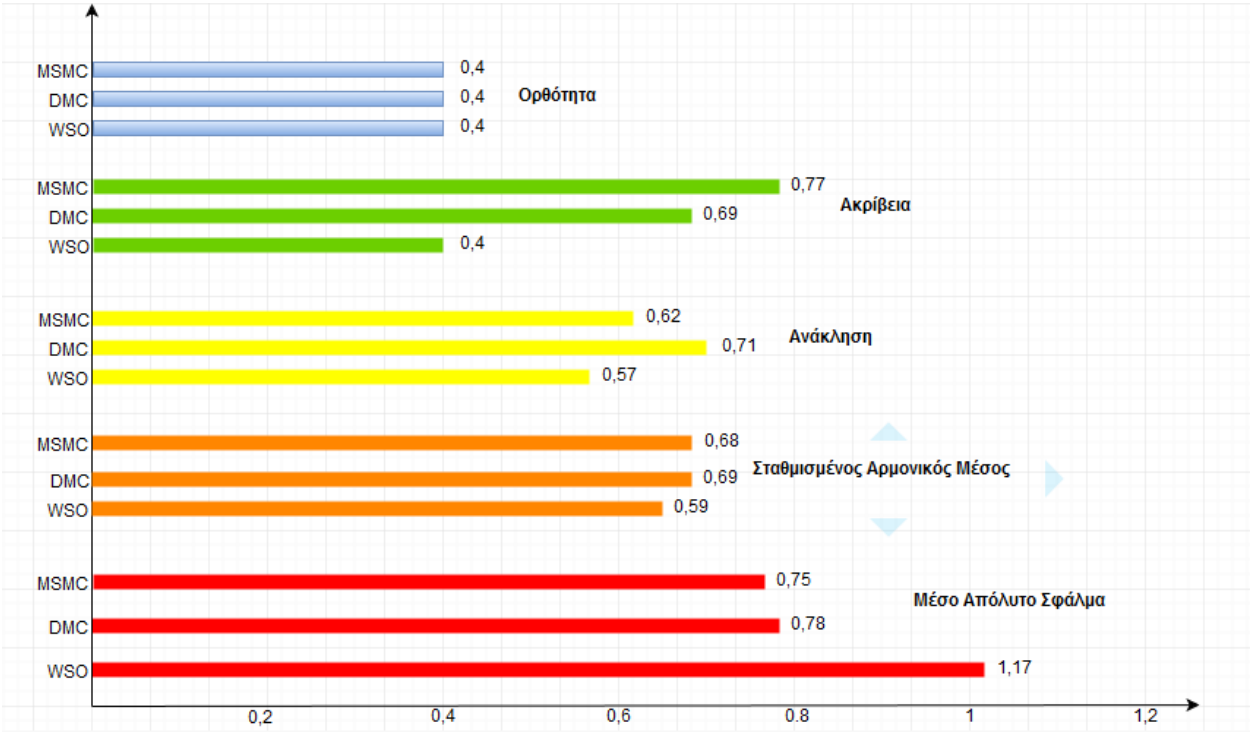
Όταν αυξήσουμε το ποσοστό του ορίου πρότασης ενός αντικειμένου που αναφέραμε παραπάνω, που αρχικά ξεκινάμε με το 70% της μέγιστης βαθμολογίας, σε 90% (σχήμα 5) ο Multi Step Multi Criteria with LSH επιφέρει καλύτερα αποτελέσματα από τον Decomposing Multi Criteria, με βάση πάντα το F-Measure. Όταν η τιμή αυτή είναι στο 80% (σχήμα 4) τότε δεν έχουμε μεγάλες διαφορές με την αρχική κατάσταση.

Αξίζει να εξετάσουμε αυτές τις μεταβολές στα αποτελέσματα για να κατανοήσουμε ποια πλεονεκτήματα προσφέρει ο κάθε αλγόριθμος. Όπως παρατηρήσαμε στην περίπτωση που το όριο για να αποφασίσει το σύστημα αν θα προτείνει ένα αντικείμενο είναι στο 90% της μέγιστης βαθμολογίας τότε ο MSMC-LSH είναι πιο αποδοτικός από τον DMC. Δηλαδή προτιμάται όταν έχουμε να προτείνουμε λίγα αντικείμενα αλλά αυτά θα είναι πολύ κοντά στις προτιμήσεις του. Ο λόγος οφείλεται στο ότι στον MSMC-LSH εξετάζουμε και βρίσκουμε ομοιότητες ως προς τα βάρη που δίνουν οι χρήστες στα επιμέρους κριτήρια, δηλαδή η πρόβλεψη εξαρτάται όχι μόνο από την βαθμολογία αλλά και στον τρόπο που βαθμολογούν οι χρήστες, το οποίο ισχυροποιεί την ομοιότητα ανάμεσα στους χρήστες. Αν βέβαια δεν θέλουμε να είμαστε τόσο απόλυτοι και θέλουμε να προτείνουμε πιο πολλά αντικείμενα στους χρήστες μπορούμε να χρησιμοποιήσουμε το DMC, ο οποίος όπως θα περιγράψουμε παρακάτω μειονεκτεί στον χρόνο εκτέλεσης σε σχέση με τους άλλους δύο αλγορίθμους αφού εξετάζει ομοιότητες για κάθε ένα επιμέρους κριτήριο ξεχωριστά.

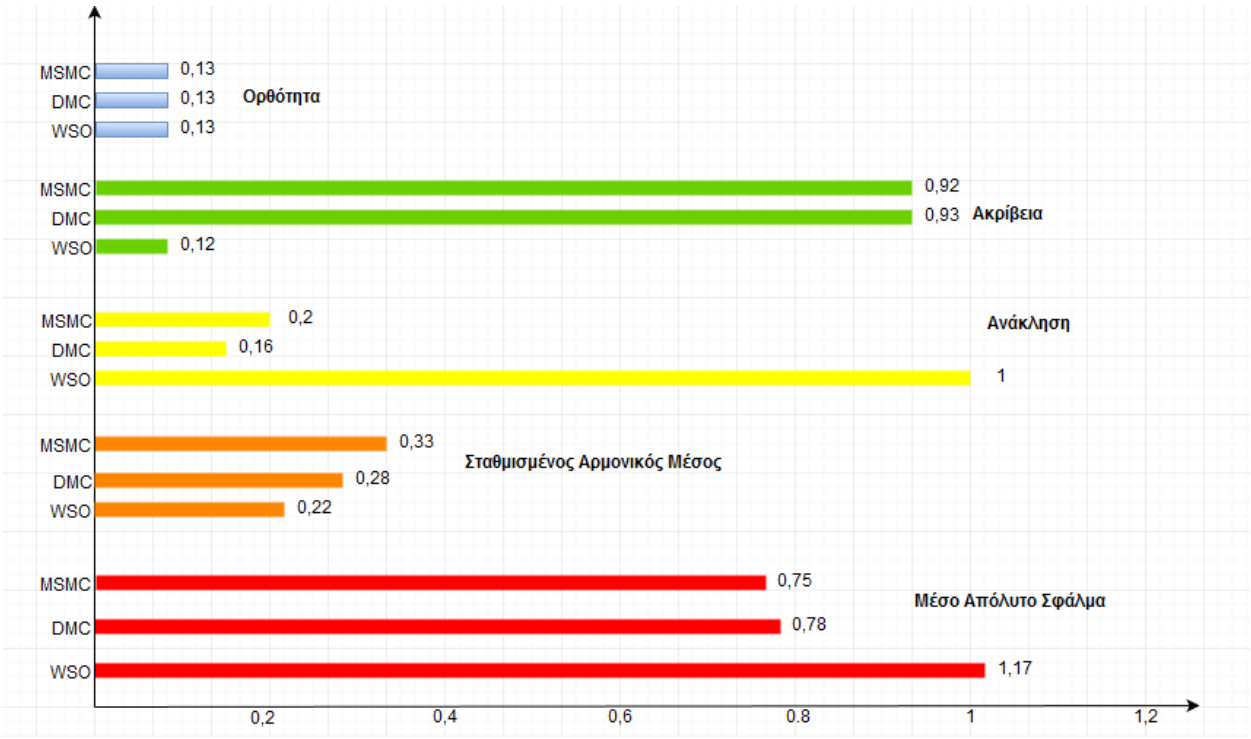
Τέλος να αναφέρουμε και το μέσο απόλυτο σφάλμα που μας απεικονίζει τις προβλέψεις του συστήματος σε σχέση με τις πραγματικές βαθμολογίες. Το WSO όπως ήταν αναμενόμενο παρουσιάζει το υψηλότερο σφάλμα αφού στην πρόβλεψη λαμβάνεται υπόψη μόνο η ολική βαθμολογία. Ο MSMC-LSH παρουσιάζει ένα ικανοποιητικό σφάλμα, το οποίο ανάλογα με τις επιλογές που θα κάνουμε για τις διάφορες τεχνικές που υλοποιείται ο αλγόριθμος, έχει μικρότερο σφάλμα από τον DMC. Οι τεχνικές αυτές είναι στα βήματα 2 και 3 η υλοποίηση με τις ομοιότητες Pearson, στο βήμα 4 η επιλογή intersections connectives using: p=1 και στο βήμα 5 επιλέγοντας την adjusted weighted sum aproach ελαχιστοποιούν το μέσο απόλυτο σφάλμα.



Σχήμα 3: Διάγραμμα μετρήσεων απόδοσης με όριο πρότασης 70% της μέγιστης βαθμολογίας .



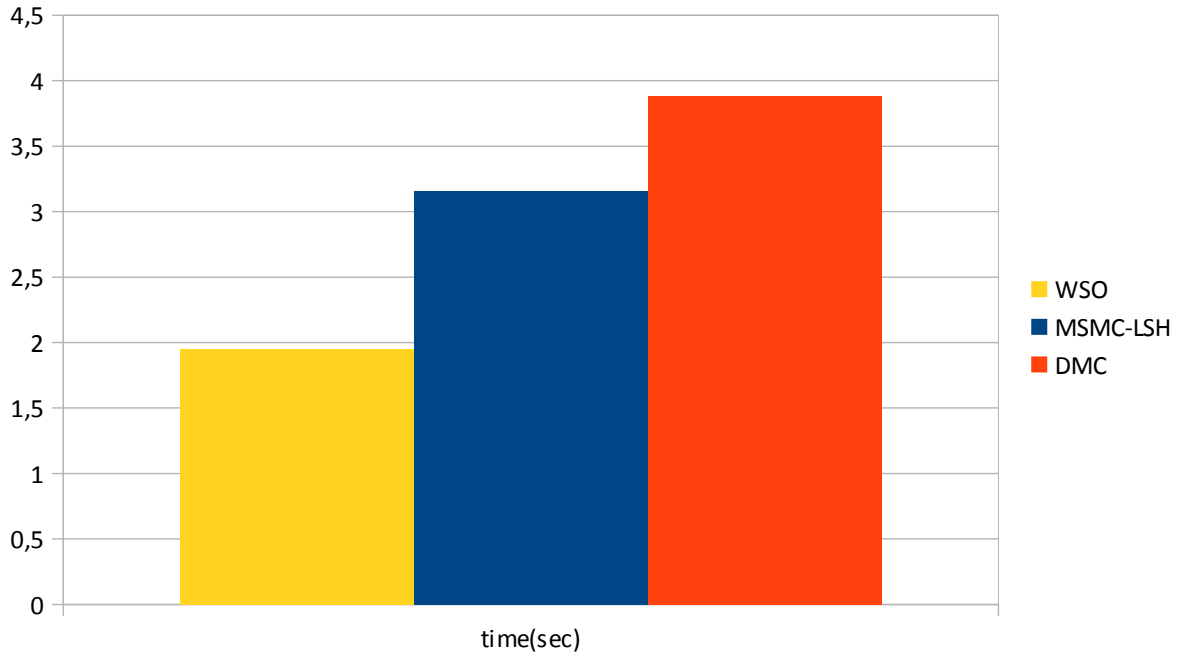
Σχήμα 4: Διάγραμμα μετρήσεων απόδοσης με όριο πρότασης 80% της μέγιστης βαθμολογίας .



Σχήμα 5: Διάγραμμα μετρήσεων απόδοσης με όριο πρότασης 90% της μέγιστης βαθμολογίας .

6.3 Σύγκριση επίδοσης αλγόριθμου Multi-Step Multi-Criteria with LSH σε σχέση με τους ήδη γνωστούς, Decomposing Multi Criteria και Weighted Slope One

Στην προηγούμενη ενότητα εξετάζαμε ουσιαστικά ποιος αλγόριθμος είναι πιο αποτελεσματικός και μας δίνει τα καλύτερα αποτελέσματα. Είναι σημαντικό βέβαια τα αποτελέσματα αυτά , ένα σύστημα να μας τα επιστρέφει έγκαιρα και ειδικά τα τελευταία χρόνια που έχουμε να εξετάσουμε πολύ μεγάλο όγκο δεδομένων. Για τον λόγο αυτό ο WSO χρησιμοποιείται πολύ συχνά στα συστήματα προτάσεων χωρίς να είναι ιδιαίτερα αποδοτικός. Ο λόγος που είναι πιο γρήγορος από τους άλλους αλγορίθμους οφείλεται στο ότι επικεντρώνεται σε μία μόνο διάσταση, αυτή της ολικής βαθμολογίας. Σε αντίθεση από τον αλγόριθμο αυτό, ο DMC είναι ο ποιο αργός από τους άλλους δύο όπως φαίνεται στο σχήμα 6. Η πολυπλοκότητα του αλγορίθμου αυτού θα μπορούσαμε να πούμε ότι είναι U^2 όπου U ο αριθμός των χρηστών, για τους οποίους χρήστες καλούμαστε να υπολογίσουμε τις αποστάσεις μεταξύ τους. Τέλος ο MSMC-LSH είναι πιο αργός από τον WSO αλλά πιο βελτιωμένος από τον DMC. Θα μπορούσαμε να πούμε ότι ο αλγόριθμος DMC μοιάζει από θέμα χρόνου αναζήτησης ομοιοτήτων με τον παλιό αλγόριθμο MSMC στην τετραγωνική πολυπλοκότητα ως προς το πλήθος των χρηστών. Επιπλέον στον αλγόριθμο MSMC-LSH δεν υπολογίζουμε αποστάσεις με όλους τους χρήστες αλλά με τους πιο κοντινούς γείτονες που έχει κάθε χρήστης, επομένως έχουμε σαφώς και πιο μικρή πολυπλοκότητα από το U^2 .



Σχήμα 6: Διάγραμμα χρόνων αναζήτησης των αλγορίθμων WSO, MSMC-LSH και DMC.

7. Συμπεράσματα

Όπως αναφέραμε και προηγουμένως τα συστήματα προτάσεων αξιολογούνται με βάση την αποτελεσματικότητά τους και την επίδοσή τους. Οι τρεις αλγόριθμοι που εξετάσαμε είχαν πλεονεκτήματα και μειονεκτήματα ως προς αυτούς τους τρόπους αξιολόγησης. Ο αλγόριθμος αποσύνθεσης πολλαπλών κριτηρίων είχε το πλεονέκτημα της καλύτερης απόδοσης όταν προτείναμε αντικείμενα με όριο το 70% και 80% της μέγιστης βαθμολογίας του χρήστη, είχε όμως το πρόβλημα της μεγάλης πολυπλοκότητας εκτέλεσής του. Αντιθέτως ο αλγόριθμος σταθμισμένης μοναδιαίας κλήσης ήταν πιο αναξιόπιστος από πλευράς αποφάσεων αφού είχε πολύ μικρή ακρίβεια, δηλαδή το σύστημα πρότεινε αντικείμενα στον χρήστη τα οποία δεν ικανοποιούσαν τις προτιμήσεις του, το υπολογιστικό κόστος όμως ήταν αρκετά ικανοποιητικό και για τον λόγο αυτό εφαρμόζεται σε πολλά συστήματα. Κάτι ενδιάμεσο από αυτούς τους δύο είναι ο αλγόριθμος πολλαπλών βημάτων πολλαπλών κριτηρίων. Είναι πιο ταχύς από τον αλγόριθμο αποσύνθεσης πολλαπλών κριτηρίων και αυτό εξηγήσαμε ότι οφείλεται κυρίως στο ότι παίρνουμε αποστάσεις από ένα υποσύνολο των χρηστών και όχι από όλους τους χρήστες. Αν δηλαδή έχουμε ένα πολύ μεγάλο πλήθος από χρήστες υπάρχει εμφανής διαφορά στο κόστος του χρόνου εκτέλεσης. Αναφέραμε επίσης ότι όταν προτείναμε αντικείμενα με όριο το 90% της μέγιστης βαθμολογίας, δηλαδή αυξήσαμε τις απαιτήσεις του χρήστη, η μέθοδος αυτή ήταν πιο αποδοτική και η απάντηση που δώσαμε σε αυτό οφείλεται στο ότι εδώ λαμβάνουμε υπόψη και τον τρόπο με τον οποίο βαθμολογεί ο χρήστης. Σε σχέση με τον αλγόριθμο μοναδιαίας κλήσης αν και παρουσιάζει μικρότερη ανάκληση είναι ισχυρότερος από πλευρά απόδοσης διότι έχει μεγαλύτερο σταθμισμένο αρμονικό μέσο, δεν υπερτερεί όμως στον χρόνο εκτέλεσης.

Βάσει λοιπόν των απαιτήσεων και των προδιαγραφών μιας εφαρμογής σύστασης προτάσεων στους χρήστες υπάρχει η ευχέρεια της επιλογής από την ομάδα ανάπτυξης της εφαρμογής αναφορικά με τον αλγόριθμο που θα πρέπει να χρησιμοποιηθεί ανάλογα με τις απαιτήσεις της εφαρμογής.

ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ

Ξενόγλωσσος όρος	**Ελληνικός Όρος**
collaborative filtering	Συνεργατικό φιλτράρισμα
content-based filtering	Φιλτράρισμα βασισμένο στο περιεχόμενο
single-rating recommender systems	Συστήματα προτάσεων ενός κριτηρίου
multi-criteria recommender systems	Συστήματα προτάσεων πολλαπλών κριτηρίων
neural nets	Νευρωνικά δίκτυα
support vector machine	Μηχανή διανυσμάτων υποστήριξης
Gaussian	Γκαουσιανή (Κανονική) κατανομή
Cosine-base similarity	Ομοιότητα με βάση το συνημίτονο
Pearson correlation-based similarity	Ομοιότητα βασισμένη στη συσχέτιση Πίρσον
weighted sum approach	Μέθοδος σταθμισμένης προσέγγισης
adjusted weighted sum approach	Μέθοδος προσαρμοζόμενης σταθμισμένης προσέγγισης
precision	Ακρίβεια
Weighted slope one	Μέθοδος Σταθμισμένης Μοναδιαίας Κλίσης
recall	Ανάκληση
accuracy	Ορθότητα
mean absolute error	Μέσο απόλυτο σφάλμα
Linear regression	Γραμμική παλινδρόμηση
locality sensitive hashing	Κατακερματισμός τοπικής ευαισθησίας

ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ

LSH	Locality Sensitive Hashing
MAE	Mean absolute error
DMC	Decomposing Multi Criteria
WSO	Weighted Slope One
MSMC	Multi Step Multi Criteria
MSMC-LSH	Multi Step Multi Criteria - Locality Sensitive Hashing

Αναφορές

[1] M. Tim Jones, IBM,12 December 2013, http://www.ibm.com/developerworks/library/os-recommender1/ , 20 April 2016.

[2] Γ. Μπουτσουκλή, “Αξιολόγηση μεθόδων προτάσεων πολλαπλών κρητηρίων”, Πτυχική Εργασία, Τμήμα Πληροφορικής και Τηλεπικοινωνιών, Εθνικό Καποδιστριακό Πανεπιστήμιο Αθηνών, 2014

[3] D. Lemire, A. Maclachlan, Slope One Predictors for Online Rating-Based Collaborative Filtering, 7 February 2005, http://lemire.me/fr/documents/publications/lemiremaclachlan_sdm05.pdf , 25 April 2016.

[4] Gediminas Adomavicius, YoungOk Kwon, New Recommendation Techniques for Multicriteria Rating Systems, University of Minnesota, May/June 2007

[5] I. Chamodrakas, N. Alexopoulou, D. Martakos Customer evaluation for order acceptance using a novel class of fuzzy methods based on TOPSIS, 2008

[6] Walt Fair, Jr., “An Algorithm for Weighted Linear Regression” 17 Apr 2008, http://www.codeproject.com/Articles/25335/An-Algorithm-for-Weighted-Linear-Regression.

[7] Manan Mohan Goyal, Neha Agrawal, Manoj Kumar Sarma, Nayan Jyoti Kalita,Comparison Clustering using Cosine and Fuzzy set based Similarity Measures of Text Documents, 1 May 2016

[8] Segaran, Toby. Programming Collective Intelligence: Building Smart Web 2.0 Applications. Sebastopol, CA: O'Reilly Media, 2007.

[9] I. Chamodrakas, N. Alexopoulou, D. Martakos Customer evaluation for order acceptance using a novel class of fuzzy methods based on TOPSIS, 2008.

[10] W. Josephson, Z. Wang, M. Charikar, K. Li, MultiProbe LSH: Efficient Indexing for HighDimensional Similarity Search, 2007

[11] George Hripcsak MD,MS, Adam S.Rothschild MD, Agreement, the F-Measure, and Reliability in Information Retrieval, 1 May 2005.