ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Υλοποίηση και βελτίωση αλγορίθμων διάχυσης σε συστήματα προτάσεων

Κωνσταντίνα Ελευθερία Γ. Κολιοπούλου

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

ΑΘΗΝΑ

ΦΕΒΡΟΥΑΡΙΟΣ 2017

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Υλοποίηση και βελτίωση αλγορίθμων διάχυσης σε συστήματα προτάσεων

Κωνσταντίνα Ελευθερία Γ. Κολιοπούλου

Α.Μ.: 1115201100036

ΕΠΙΒΛΕΠΩΝ: **Ιωάννης Χαμόδρακας,** μέλος του Εργαστηριακού Διδακτικού Προσωπικού του Τμήματος Πληροφορικής και Τηλεπικοινωνιών, υπό την εποπτεία του Καθηγητή **Ιωάννη Εμίρη**

ΠΕΡΙΛΗΨΗ

Λόγω της εκρηκτικής αύξησης του όγκου των πληροφοριών στο διαδίκτυο, της συνεπαγόμενης ανάγκης διαχείρισής τους και μετάδοσής τους στο κατάλληλο κοινό, ο ρόλος των συστημάτων προτάσεων καθίσταται κρίσιμος. Σε αυτό το πλαίσιο, τα τελευταία χρόνια έχουν σχεδιαστεί αλγόριθμοι διαφόρων ειδών με στόχο τη βελτίωση της απόδοσης και της αποτελεσματικότητάς τους. Η παρούσα πτυχιακή επικεντρώνεται στην ανάπτυξη συστημάτων προτάσεων με χρήση αλγορίθμων που βασίζονται σε μεθόδους διάχυσης με τη χρήση διμερών γράφων. Σκοπός της είναι η υλοποίηση, η αξιολόγηση, βελτίωση και σύγκριση των Weighted Slope One, Heat Spreading (HeatS) και Probabilistic Spreading (ProbS) αλγορίθμων. Αρχικά, γίνεται επεξήγηση της λειτουργίας κάθε αλγορίθμου και του τρόπου με τον οποίο διαχειρίζονται την πληροφορία που τους δίνεται. Στη συνέχεια, εξετάζεται ο αλγόριθμος Hybrid Spreading (HybridS) που αποτελεί συνδυασμό των αλγορίθμων HeatS και ProbS με στόχο την παραγωγή βελτιωμένων προτάσεων. Επίσης, χρησιμοποιείται ο αλγόριθμος πλησιέστερων γειτόνων LSH-Superbit για τη μείωση της πολυπλοκότητας του χρόνου του προβλήματος. Τέλος, οι αλγόριθμοι αξιολογούνται μέσα από την εκτέλεση πειραμάτων ως προς την αποδοτικότητα, την αποτελεσματικότητα και την εγκυρότητά τους με τη χρήση διάφορων γνωστών μετρικών αξιολόγησης.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Συστήματα προτάσεων

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Αλγόριθμοι διάχυσης, διμερής γράφος, Weighted Slope One, HeatS, ProbS, HybridS, παραγωγή προτάσεων, μετρικές αξιολόγησης, αποδοτικότητα, αποτελεσματικότητα, εγκυρότητα.

ABSTRACT

Due to the explosive growth of the amount of available information on the Internet, the need for management and transmission to the appropriate audience, namely the role of recommendation systems, becomes consequently critical. In this context, various kinds of algorithms have been designed the last years aiming to improve their performance and efficiency. This thesis focuses on the development of recommendation systems with the use of diffusion based algorithms which use bipartite networks. The primary goals of this thesis are the implementation, evaluation and comparison of Weighted Slope One (WSO), Heat Spreading (HeatS) and Probabilistic Spreading (ProbS) algorithms. Initially, the function of each algorithm is explained and the way with which they handle the information given. Afterwards, the Hybrid Spreading algorithm (HybridS) is examined which is a combination of HeatS and ProbS algorithms aiming to produce improved recommendations. Also, the nearest neighbor (locality-sensitive hashing) LSH-Superbit algorithm is used in order to reduce the time complexity of the problem. Last but not least, algorithms are evaluated in terms of efficiency, effectiveness and validity using various known evaluation metrics.

SUBJECT AREA: Recommendation systems.

KEYWORDS: Diffusion based algorithms, bipartite network, Weighted Slope One, HeatS, ProbS, predictions, evaluation metrics, efficiency, effectiveness, validity.

Στην οικογένειά μου

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω θερμά τον Δρ. Ιωάννη Χαμόδρακα που μου έδωσε τη δυνατότητα να ασχοληθώ με ένα τόσο ενδιαφέρον θέμα. Θα ήθελα επίσης να τον ευχαριστήσω για την συνεχή καθοδήγηση, την υποστήριξη, τις ουσιώδεις συμβουλές και την αφιέρωση πολύτιμου χρόνου ώστε να ολοκληρωθεί η πτυχιακή αυτή. Τέλος, θα ήθελα να ευχαριστήσω θερμά τον Καθηγητή κ. Ιωάννη Εμίρη για την εμπιστοσύνη που μου έδειξε στην ανάθεση της παρούσας πτυχιακής εργασίας και για το χρόνο που αφιέρωσε για τη συνολική εποπτεία της.

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

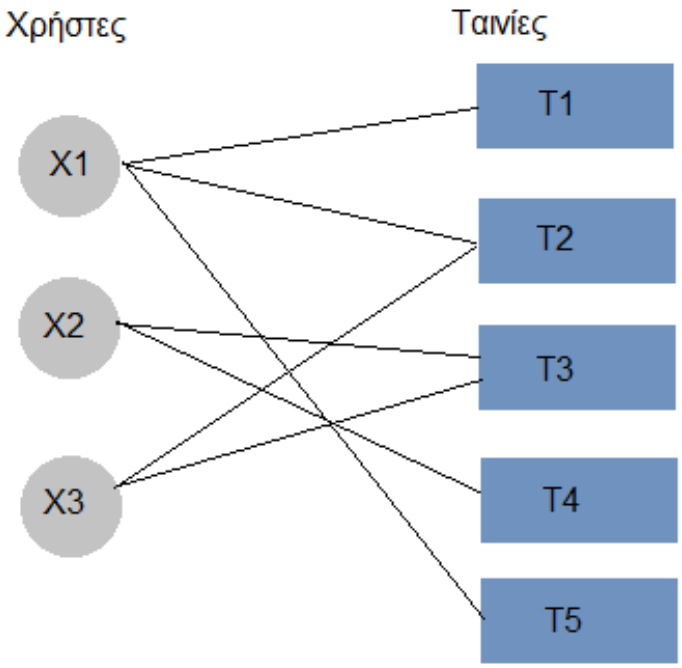
1. ΕΙΣΑΓΩΓΗ

Τα τελευταία χρόνια έχουν παρατηρηθεί αρκετά σοβαρά προβλήματα λόγω της συνεχόμενης αύξησης της πληροφορίας. Η ποσότητά της, ειδικά στο διαδίκτυο, αυξάνεται πολύ πιο γρήγορα από την ικανότητά μας να τη διαχειριστούμε. Για παράδειγμα υπάρχουν χιλιάδες ταινίες, εκατομμύρια βιβλία, δισεκατομμύρια ιστοσελίδες. Η συνεχιζόμενη ταχεία επέκταση των πόρων του διαδικτύου σε μεγάλο βαθμό εντείνει την ανάγκη ανάπτυξης όλο και πιο αποτελεσματικών εργαλείων για το φιλτράρισμα της άφθονης αυτής πληροφορίας. [1] Ως απόρροια των παραπάνω, έχουν προταθεί πολυάριθμοι αλγόριθμοι βασισμένοι ο καθένας σε διαφορετικές ιδέες και έννοιες. Οι συνηθέστεροι αυτών συναντώνται σε κατηγορίες που βασίζονται είτε στη μέθοδο του συνεργατικού φιλτραρίσματος (collaborative filtering) με το οποίο και θα ασχοληθούμε, είτε στη μέθοδο φιλτραρίσματος με βάση το περιεχόμενο (content based filtering).

Το συνεργατικό φιλτράρισμα αποτελεί τεχνική που χρησιμοποιείται ευρέως στα συστήματα προτάσεων. Αποτελεί μια μέθοδο σύμφωνα με την οποία εξάγονται αυτόματα προβλέψεις όσον αφορά τα ενδιαφέροντα και τις προτιμήσεις ενός χρήστη, συλλέγοντας τις προτιμήσεις άλλων χρηστών. Ουσιαστικά η βασική υπόθεση του συνεργατικού φιλτραρίσματος είναι ότι εάν σε κάποιον χρήστη Χ άρεσε το Α αντικείμενο και το ίδιο αντικείμενο άρεσε και στο χρήστη Ψ, τότε ο χρήστης Χ είναι πιο πιθανό να έχει ίδια γνώμη για ένα αντικείμενο Β με το χρήστη Ψ από ό,τι θα είχε με κάποιον τυχαίο χρήστη. Έτσι λοιπόν, δημιουργείται ένα προφίλ για το χρήστη βασισμένο σε παλαιότερες ενέργειές του, συνδυάζοντας τη συμπεριφορά του με τη συμπεριφορά άλλων χρηστών παρόμοιων με εκείνον, δηλαδή με παρόμοιες προτιμήσεις και ενδιαφέροντα.

Για την καλύτερη διαχείριση των διάφορων πληροφοριών που μπορεί να συλλεχθούν για ένα χρήστη, για παράδειγμα από κάποια ιστοσελίδα στο διαδίκτυο, όπως το Amazon ή το IMDB, η χρήση διμερών γράφων (ή διαφορετικά διμερών δικτύων) αποτελεί ευέλικτο εργαλείο. [1] Πιο συγκεκριμένα η παρούσα πτυχιακή θα εστιάσει σε δίκτυα τα οποία ονομάζονται χρήστη-αντικειμένου (web-based user-object networks). Τα συγκεκριμένα δίκτυα αντιπροσωπεύουν τις αλληλεπιδράσεις μεταξύ των χρηστών και των αντικειμένων κυρίως σε ιστοσελίδες στο διαδίκτυο, όπως το ιστορικό των αγορών αντικειμένων από το eBay ή τα video που έχουν παρακολουθήσει στο YouTube.

Ένα δίκτυο G(V,E) αποτελεί ένα διμερές δίκτυο αν υπάρχει ένα επιμέρους σύνολο (V1,V2) όπου $V1 \cup V2 = V$, $V1 \cap V2 = \emptyset$ και κάθε ακμή συνδέει έναν κόμβο του V1 με έναν κόμβο του V2. Υπάρχουν πολλά παραδείγματα στο φυσικό κόσμο τα οποία διαμορφώνονται ως διμερή δίκτυα, τέτοια είναι: το δίκτυο του μεταβολισμού το οποίο αποτελείται από χημικές ουσίες και χημικές αντιδράσεις, το δίκτυο συνεργασίας ηθοποιών που αποτελείται από ηθοποιούς και ρόλους κ.α. [1] Στην παρούσα πτυχιακή το διμερές δίκτυο που θα διαμορφωθεί αποτελείται από ταινίες και χρήστες. Για την ευκολότερη κατανόησή του δίδεται σχετικό παράδειγμα στην *Εικόνα 1 .*



Εικόνα 1: Παράδειγμα διμερή γράφου μεταξύ χρηστών και ταινιών, κάθε κόμβος χρήστη συνδέεται με τον αντίστοιχο κόμβο ταινίας η οποία του άρεσε.

Θα μπορούσε λοιπόν κανείς να επιδιώξει την εξαγωγή προτάσεων χρησιμοποιώντας μια αναπαράσταση των δεδομένων εισόδου, των προτιμήσεων του χρήστη, σε ένα τέτοιο δίκτυο. Οι αλγόριθμοι που θα παρουσιαστούν στη συνέχεια βασίζονται σε συγκεκριμένους μετασχηματισμούς ή αλλιώς προβολές των δεδομένων εισόδου σε ένα δίκτυο χρήστη-αντικειμένου. Οι προτάσεις για κάθε μεμονωμένο χρήστη εξάγονται χρησιμοποιώντας τις παλαιότερες προτιμήσεις του ως «πηγή» σε ένα δεδομένο δίκτυο ώστε στη συνέχεια να του προταθούν καινούρια αντικείμενα τα οποία δεν είχε αξιολογήσει. [1]

Πρόσφατα, έχουν προταθεί αλγόριθμοι προτάσεων βασισμένοι στη διάχυση. Οι συγκεκριμένοι αλγόριθμοι ουσιαστικά βασίζονται στην αναπαράσταση εκείνου του φαινομένου που ονομάζουμε στη φυσική ως θερμική διάχυση. Με αυτόν τον τρόπο και για να γίνει πιο κατανοητό το πως λειτουργούν οι συγκεκριμένοι αλγόριθμοι μπορούμε να φανταστούμε ότι κάθε χρηστής έχει μια συγκεκριμένη ποσότητα θερμότητας την οποία μεταφέρει στους γειτονικούς χρήστες, με αποτέλεσμα σταδιακά να προκαλείται μια διάχυση θερμότητας από τον έναν χρήστη στον άλλον και κατ' επέκταση σε όλο το δίκτυο. Στην παρούσα πτυχιακή οι αλγόριθμοι που θα αναλυθούν είναι ο HeatS ή αλλιώς αλγόριθμος θερμικής αγωγιμότητας (heat conduction) και ο ProbS ή αλλιώς αλγόριθμος πιθανοτικής εξάπλωσης (probability spreading). Στην περίπτωση μας κάθε χρήστης θα έχει κάποιο σύνολο πόρων (resources) αντί για θερμότητα, όπως αναφέραμε στο προηγούμενο παράδειγμα. Χρησιμοποιώντας αντίστοιχα τους προαναφερθέντες αλγορίθμους στο διμερές δίκτυο μεταξύ χρηστών ταινιών θα έχουμε μια μεταφορά πόρων μεταξύ χρηστών και ταινιών με απώτερο σκοπό τη μεταφορά πόρων από κάθε χρήστη στις ταινίες όπου και θα προκύψει το τελικό αποτέλεσμα από το οποίο θα εξάγουμε αντίστοιχα τις προτάσεις για κάθε χρήστη. Θα αναφερθούμε αναλυτικότερα στο κεφάλαιο 2 για τη λειτουργία κάθε αλγορίθμου ξεχωριστά.

Τέλος, στην παρούσα πτυχιακή πέραν της μελέτης των παραπάνω αλγορίθμων θα παρουσιαστεί πως αυτοί βελτιώνονται αν συνδυαστούν σε έναν υβριδικό αλγόριθμο, τον

HybridS. Θα ακολουθήσει σύγκριση μεταξύ τους, αλλά και με έναν ακόμη γνωστό αλγόριθμο τον Weighted Slope One (WSO). Επιπλέον, για την περαιτέρω βελτίωση των αποτελεσμάτων τους θα προταθεί η χρήση Local Sensitive Hashing (LSH) αφού όπως θα διαπιστώσουμε στα επόμενα κεφάλαια με τη χρήση του LSH σημειώνεται βελτίωση των αλγορίθμων τόσο σε αποδοτικότητα όσο και σε αποτελεσματικότητα. Πιο συγκεκριμένα ο HybridS με τη χρήση LSH θα αποδειχτεί ότι αποτελεί την καλύτερη επιλογή μεταξύ των άλλων αλγορίθμων αφού, όπως θα δούμε και στο Κεφάλαιο 4 πιο αναλυτικά, σημειώνει τις υψηλότερες επιδόσεις σε ακρίβεια, ανάκληση αλλά και στη διαφορετικότητα των προτάσεων του.

2. ΑΝΑΛΥΣΗ ΑΛΓΟΡΙΘΜΩΝ

Αρχικά πριν ξεκινήσουμε την ανάλυση κάθε αλγορίθμου θα ήταν ορθό για την καλύτερη κατανόηση από τον αναγνώστη να γίνει μια σύντομη αναφορά στον τύπο των δεδομένων που θα διαχειριστούμε.

Όπως ήδη έχουμε αναφέρει τα δεδομένα μας αφορούν χρήστες και ταινίες. Το σύνολο των δεδομένων που θα χρησιμοποιηθεί για να διερευνήσουμε πως συμπεριφέρεται κάθε αλγόριθμος είναι το MovieLens. [2] Το σύνολο αυτό αποτελείται από 943 χρήστες και 1.682 ταινίες. Κάθε χρήστης έχει βαθμολογήσει τουλάχιστον 20 ταινίες. Συνολικά το σύνολο των δεδομένων MovieLens περιέχει 100.000 βαθμολογίες χρηστών σε μια διαβάθμιση από το 1 μέχρι το 5.

Στη συνέχεια ακολουθεί περιγραφή των αλγορίθμων που θα χρησιμοποιηθούν καθώς και σχετικά παραδείγματα για την καλύτερη κατανόηση τους.

2.1 Heat Spreading algorithm (HeatS)

Ο παρών αλγόριθμος αναπαριστά και βασίζεται σε μια διαδικασία ανάλογη της θερμικής αγωγιμότητας. Γενικά, όπως αναφέραμε ένα σύστημα προτάσεων μπορεί να αναπαρασταθεί ως ένας διμερής γράφος, στον οποίον θα έχουμε δύο κόμβους: χρήστες U και αντικείμενα (ταινίες) O. Υποθέτοντας ότι έχουμε m αντικείμενα $\mathrm{O} = \{\mathrm{o}_1,\mathrm{o}_2, \ldots ,\mathrm{o}_\mathrm{m}\}$ και n χρήστες $\mathrm{U} = \{\mathrm{u}_1,\mathrm{u}_2,\ldots,\mathrm{u}_\mathrm{n}\}$ το σύστημά μας μπορεί να αναπαρασταθεί από ένα $\mathrm{m}\times\mathrm{n}$ μητρώο γειτνίασης $\mathrm{A} = \{\ \alpha_{\mathrm{i}\alpha}\ \}_{\mathrm{m,n}}$ όπου το στοιχείο $\alpha_{i\alpha} = 1$ αν στον χρήστη i άρεσε η ταινία α, διαφορετικά ισούται με 0. [3] Στην περίπτωση του παρόντος dataset όπου έχουμε βαθμολογίες από 1 έως 5, θεωρούμε ότι για βαθμολογίες ≥3 η ταινία άρεσε στο χρήστη, όποτε και συμπληρώνουμε το μητρώο γειτνίασης με 1, διαφορετικά για βαθμολογίες μικρότερες ή σε περίπτωση που δεν έχει δοθεί καν βαθμολογία παραμένει 0.

Κάθε χρήστης θεωρούμε ότι έχει ένα σύνολο πόρων, οι οποίοι προκύπτουν ανάλογα από τις ταινίες που του άρεσαν. Η διαδικασία ανακατανομής των πόρων αυτών μπορεί να πραγματοποιηθεί χρησιμοποιώντας την εξής απλή εξίσωση:

$$f = W f_0 \quad (1)$$

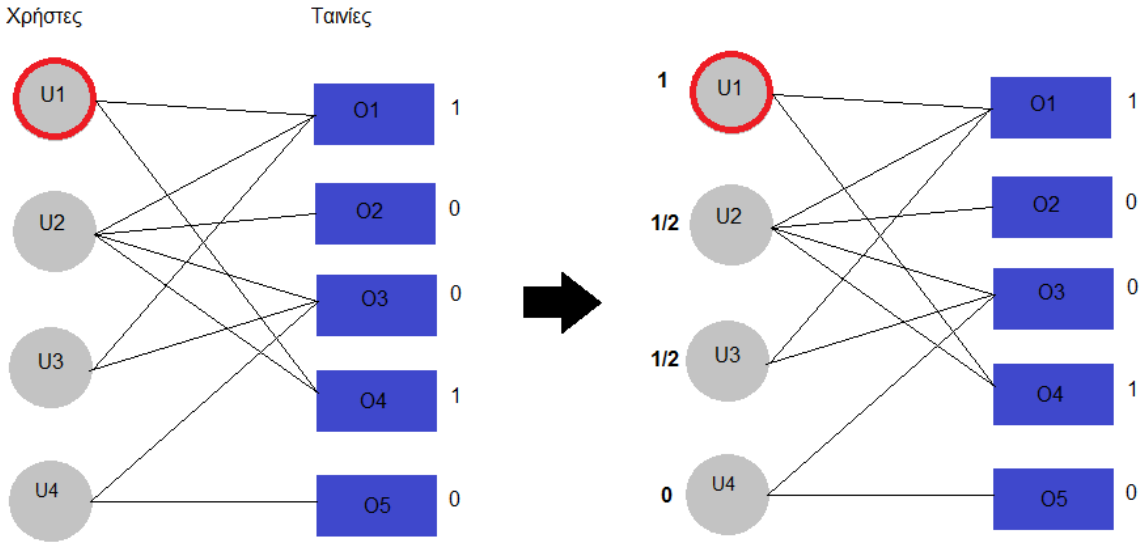
όπου $f_0 = [f_{1,0}^i , \ldots , f_{n,0}^i]$ η αρχική διαμόρφωση των πόρων στις ταινίες, W το μητρώο ανακατανομής των πόρων και $f = [f_1^i, \ldots , f_n^i]$ η τελική διαμόρφωση των πόρων στις ταινίες. Στη συνέχεια τα αντικείμενα ταξινομούνται σε φθίνουσα σειρά όπου επιλέγεται ένα συγκεκριμένο σύνολο αυτών που αντιστοιχεί στις ταινίες που τελικά διαθέτουν τους υψηλότερους πόρους και δεν έχουν ήδη βαθμολογηθεί από τον χρήστη ώστε να παραχθούν οι προτάσεις. Ξέροντας επομένως το μητρώο ανακατανομής των πόρων W και την αρχική διαμόρφωση αυτών f_0 μπορούμε να υλοποιήσουμε τον εν λόγω αλγόριθμο.

Πιο συγκεκριμένα στον heat spreading αλγόριθμο, το μητρώο ανακατανομής των πόρων διαμορφώνεται ως εξής:

$$W_{\alpha\beta}^{H} = \frac{1}{\kappa_\alpha}\sum_{i=1}^{m}\frac{\alpha_{i\alpha}\alpha_{i\beta}}{\kappa_i} \quad (2)$$

όπου κ_α είναι ο βαθμός του o_α, δηλαδή το πλήθος των χρηστών που τους άρεσε το αντικείμενο a, και κ_i ο βαθμός του u_i, δηλαδή το πλήθος των αντικειμένων που άρεσαν στον χρήστη i. Επομένως θέλοντας να εξάγουμε προτάσεις για έναν συγκεκριμένο χρήστη i καθορίζουμε το αρχικό διάνυσμα των πόρων του f^i σύμφωνα με τις ταινίες που ήδη του αρέσουν, δηλαδή για κάποια ταινία β θέτουμε $f_\beta^i = \alpha_{i\beta}$ και στη συνέχεια από την (1) και εξάγοντας το μητρώο ανακατανομής των πόρων προκύπτει η λίστα με τις αντίστοιχες προτάσεις ταινιών. Για παράδειγμα, αν είχαμε τέσσερις ταινίες και έναν στόχο-χρήστη, για τον οποίον θα θέλαμε να εξάγουμε προτάσεις, στον οποίον άρεσαν οι δύο από αυτές τις τέσσερις ταινίες, τότε αν σε έναν άλλον διαφορετικό χρήστη άρεσαν δυο ταινίες, εκ των οποίων η μία άρεσε και στο χρήστη-στόχο, και δεν του άρεσαν οι άλλες δυο ταινίες, τότε οι πόροι που θα του ανατίθεντο θα ήταν ίσοι με ½. Στη συνέχεια, ο αλγόριθμος για κάθε ταινία θα υπολόγιζε το μέσο όρο των πόρων των χρηστών που συνδέονται με αυτή, αναθέτοντας με αυτό τον τρόπο και την τελική τιμή των πόρων της, δηλαδή αν είχαμε μια ταινία η οποία άρεσε σε δυο χρήστες όπου ο πρώτος είχε πόρους ίσους με ½ και ο δεύτερος ίσους με 0, τότε στην ταινία θα ανατίθεντο πόροι ίσοι με ¼ ($\frac{\frac{1}{2}+0}{2} = \frac{1}{4}$). Η συμπεριφορά του αλγορίθμου εξετάζεται στη συνέχεια με ένα πιο λεπτομερές παράδειγμα.

Είναι σημαντικό να σημειωθεί ότι ο HeatS αποτελεί αλγόριθμο ο οποίος προωθεί μη δημοφιλή αντικείμενα, με αποτέλεσμα ταινίες που έχουν προτιμηθεί από λιγότερους χρήστες να αποκτούν μεγαλύτερη τιμή πόρων και να ανέρχονται ψηλά στη λίστα των προτάσεων. Ακολουθεί μια σύντομη απεικόνιση της λειτουργίας του αλγορίθμου για την καλύτερη επεξήγησή του.



Εικόνα 2: Πρώτο βήμα διαμοιρασμού των πόρων στον αλγόριθμο HeatS.

Σύμφωνα με την παραπάνω απεικόνιση έχουμε τέσσερις χρήστες, με τον U1 ως το χρήστη για τον οποίον θα εξάγουμε προτάσεις, και πέντε ταινίες. Σε κάθε ταινία την οποία ο στόχος-χρήστης μας έχει προτιμήσει σημειώνουμε 1 διαφορετικά 0. Η διαδικασία αναδιανομής των πόρων από και προς στις ταινίες και στους χρήστες γίνεται μέσω μιας διαδικασίας μέσου όρου. Στο παρόν βήμα οι χρήστες λαμβάνουν ένα επίπεδο πόρων ίσο με τη μέση τιμή της ποσότητας που κατέχεται από τις γειτονικές

ταινίες, και όπως θα δούμε παρακάτω στη συνέχεια οι ταινίες λαμβάνουν πίσω τη μέση τιμή των πόρων των γειτονικών τους χρηστών. Πιο αναλυτικά για το πρώτο βήμα εργαζόμαστε σύμφωνα με τον τύπο:

$$f(U_i) = \sum_{a=1}^{n} \frac{a_{ia}}{\kappa_i} f_a^j \quad (3)$$

όπου i ο χρήστης στον οποίον θα γίνει η ανάθεση των πόρων, j ο στόχος-χρήστης και $f_a^j = \alpha_{ja}$ οι αρχικοί πόροι του στόχου-χρήστη για το αντικείμενο α. Για παράδειγμα, ο χρήστης U2 έχει τιμή πόρων ίση με $\frac{1}{2}$ αφού από το σύνολο των τεσσάρων ταινιών που του άρεσαν οι δύο από αυτές άρεσαν και στο στόχο-χρήστη U1. Ουσιαστικά το σύνολο των πόρων που ανατίθεντο στο χρήστη U2 προκύπτει από το μέσο όρο των πόρων των ταινιών που του άρεσαν. Δηλαδή θα έχουμε:

$$f(U_2) = \frac{a_{21}}{\kappa_2} a_{11} + \frac{a_{22}}{\kappa_2} a_{12} + \frac{a_{23}}{\kappa_2} a_{13} + \frac{a_{24}}{\kappa_2} a_{14} = \frac{1}{4} \times 1 + \frac{1}{4} \times 0 + \frac{1}{4} \times 0 + \frac{1}{4} \times 1 = \frac{2}{4} = \frac{1}{2}$$

Στη συνέχεια, οι πόροι ρέουν πίσω στις ταινίες βρίσκοντας για κάθε ταινία το μέσο όρο των πόρων των χρηστών στους οποίους άρεσε. Προκειμένου να καταλήξουμε στο αποτέλεσμα της *Εικόνας 3* εργαζόμαστε σύμφωνα με τον τύπο:

$$f(O_a) = \sum_{i=1}^{m} \frac{a_{ia} f(U_i)}{\kappa_\alpha} \quad (4)$$

Αναλύοντας τον τύπο (4) και σύμφωνα με [7] :

$$f(O_a) = \sum_{i=1}^{m} \frac{a_{ia} f(U_i)}{\kappa_\alpha} \underset{(3)}{\Rightarrow} f(O_a) = \sum_{i=1}^{m} \frac{a_{ia}}{\kappa_a} \sum_{\beta=1}^{n} \frac{a_{i\beta} f_\beta^j}{\kappa_i} \underset{(2)}{\Rightarrow}$$

$$f(O_a) = \sum_{\beta=1}^{n} W_{\alpha\beta} f_\beta^j$$

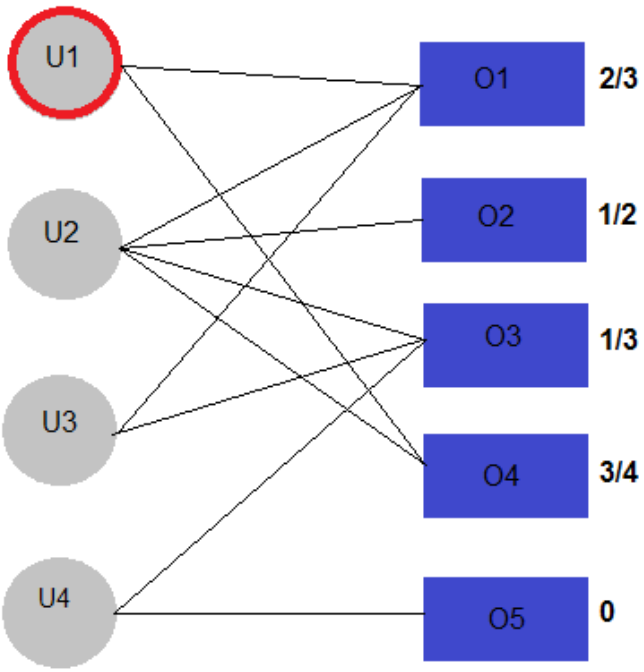
Όπου όπως ήδη έχει αναφερθεί $W_{\alpha\beta}$ το μητρώο ανακατανομής των πόρων όπως ορίζεται σύμφωνα με την εξίσωση (2) και $f_\beta^j = \alpha_{j\beta}$ η αρχική τιμή των πόρων του χρήστη j για το αντικείμενο β. Γράφοντας την παραπάνω εξίσωση σε μορφή μητρώων βλέπουμε πως εύκολα καταλήγουμε στην εξίσωση (1). [7] Επομένως γίνεται εύκολα αντιληπτό πως η εξίσωση (1), μέσω της (2), αναλύεται στις εξισώσεις (3) και (4).

Συνεχίζοντας το παράδειγμά μας για την ταινία O_3 έχουμε:

$$f(O_3) = \frac{a_{23} f(U_2)}{\kappa_3} + \frac{\alpha_{33} f(U_3)}{\kappa_3} + \frac{\alpha_{43} f(U_4)}{\kappa_3} = \frac{\frac{1}{2}}{3} + \frac{\frac{1}{2}}{3} + \frac{0}{3} = \frac{2}{6} = \frac{1}{3}$$

Τέλος, αφού σε κάθε ταινία έχουν διαμοιραστεί και οι αντίστοιχοι πόροι το μόνο που μένει είναι η διαλογή εκείνων που δεν έχουν ήδη προτιμηθεί από τον στόχο-χρήστη και

έχουν τη μεγαλύτερη ποσότητα πόρων. Επομένως οι προτάσεις που θα εξάγονταν για το χρήστη U_1 θα ήταν οι ταινίες O_2 και O_3.



Εικόνα 3: Τελικό βήμα αλγορίθμου HeatS.

2.2 Probability Spreading algorithm (ProbS)

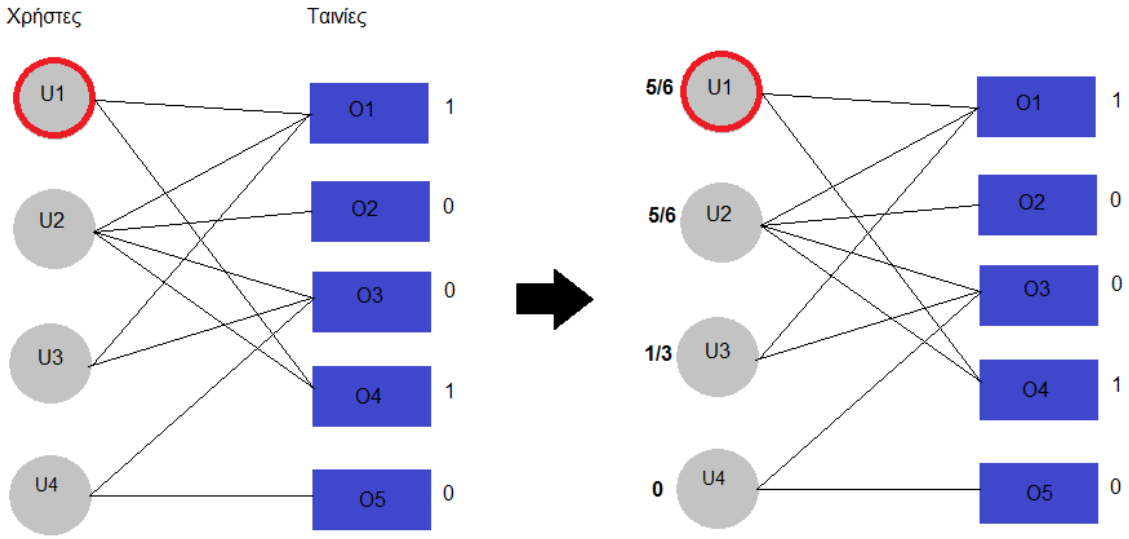
Όπως αναφέραμε και στον αλγόριθμο HeatS το σύστημα προτάσεων μας μπορεί να αναπαρασταθεί ως ένας διμερής γράφος, στον οποίον θα έχουμε δύο κόμβους: χρήστες U και αντικείμενα (ταινίες) O. Έτσι υποθέτοντας ότι έχουμε m αντικείμενα $\mathrm{O} = \{\mathrm{o}_1,\mathrm{o}_2,\ldots,\mathrm{o}_\mathrm{m}\}$ και n χρήστες $\mathrm{U} = \{\mathrm{u}_1,\mathrm{u}_2,\ldots,\mathrm{u}_\mathrm{n}\}$ το σύστημά μας μπορεί να αναπαρασταθεί από ένα $\mathrm{m}\times\mathrm{n}$ μητρώο γειτνίασης $\mathrm{A} = \{\ \alpha_{\mathrm{i}\alpha}\ \}_{\mathrm{m,n}}$ όπου το στοιχείο $\alpha_{i\alpha} = 1$ αν στον χρήστη i άρεσε η ταινία α, διαφορετικά ισούται με 0. [3]

Ο αλγόριθμος ProbS αποτελεί μέθοδο που προσπαθεί να διαμοιράσει τους πόρους μεταξύ των γειτονικών χρηστών ομοιόμορφα. Αρχικά εκχωρούμε για κάθε χρήστη ένα αρχικό επίπεδο πόρων στα αντικείμενα που αυτοί ήδη έχουν επιλέξει συμβολιζόμενο από το διάνυσμα f (οπού f_β είναι ο πόρος που κατέχει το αντικείμενο β). Στην περίπτωση μας το αρχικό επίπεδο πόρου είναι ίσο με 1. [4] Η διαδικασία ανακατανομής των πόρων αυτών πραγματοποιείται χρησιμοποιώντας την εξίσωση (1), όπου όμοια ισχύει ότι όπου $f_0 = [f_{1,0}^i, \ldots, f_{n,0}^i]$ είναι η αρχική διαμόρφωση των πόρων στις ταινίες, W το μητρώο ανακατανομής των πόρων και $f = [f_1^i, \ldots, f_n^i]$ η τελική διαμόρφωση των πόρων στις ταινίες.

Η διάφορα έγκειται στη μορφή του μητρώου ανακατανομής των πόρων W, το οποίο και διαμορφώνεται σύμφωνα με τον παρακάτω τύπο:

$$W_{\alpha\beta}^{P} = \frac{1}{\kappa_\beta}\sum_{i=1}^{m}\frac{\alpha_{i\alpha}\alpha_{i\beta}}{\kappa_i} \quad (5)$$

όπου κ_β είναι ο βαθμός του o_β, δηλαδή το πλήθος των χρηστών που προτίμησαν το αντικείμενο β και κ_i ο βαθμός του u_i, δηλαδή το πλήθος των αντικειμένων που άρεσαν στο χρήστη i. Στον ProbS, μια δημοφιλής ταινία με υψηλό βαθμό θα έχει περισσότερους τελικούς πόρους, έχοντας υψηλή θέση στη λίστα προτάσεων του χρήστη. Ωστόσο είναι προφανές ότι αλγόριθμοι όπως ο HeatS και ο ProbS εξαρτώνται αρκετά από τη μορφολογία του διμερούς δικτύου στο οποίο απευθύνονται. Στον ProbS τα δημοφιλή στοιχεία έχουν το προβάδισμα ενώ εκείνα τα μη δημοφιλή είναι πολύ πιθανό να μην προταθούν καθόλου, αφού η τελική διαμόρφωση των πόρων τους μπορεί να είναι μηδενική. Προκειμένου να γίνει ακόμα πιο κατανοητή η λειτουργία του αλγορίθμου ακολουθεί σχετικό παράδειγμα:



Εικόνα 4: Πρώτο βήμα διαμοιρασμού των πόρων στον ProbS.

Όπως και προηγουμένως έχουμε τέσσερις χρήστες, πέντε ταινίες και ως στόχο-χρήστη τον U_1. Παρατηρούμε ότι συγκριτικά με τα αποτελέσματα του HeatS ο χρήστης 1 έχει ίδιο αριθμό πόρων με εκείνων του 2, στον οποίον άρεσαν εξίσου οι ίδιες ταινίες. Επομένως οι πόροι του δεύτερου δε διαμοιράζονται ανάλογα με το πλήθος των ταινιών που έχει προτιμήσει, αλλά διαμοιράζονται ως προς την ομοιότητά του με τον στόχο-χρήστη. Πιο αναλυτικά, προκειμένου να βρεθεί το σύνολο των πόρων κάθε χρήστη στο παρόν βήμα ακολουθούμε τον εξής τύπο:

$$f(U_i) = \sum_{a=1}^{n}\frac{a_{ia}}{\kappa_a}f_a^j \quad (6)$$

όπου j ο χρήστης για τον οποίο θα εξάγουμε προτάσεις και $f_a^j = \alpha_{ja}$ οι αρχικοί πόροι του στόχου-χρήστη για το αντικείμενο α.

Επομένως για κάθε χρήστη αντίστοιχα θα έχουμε:

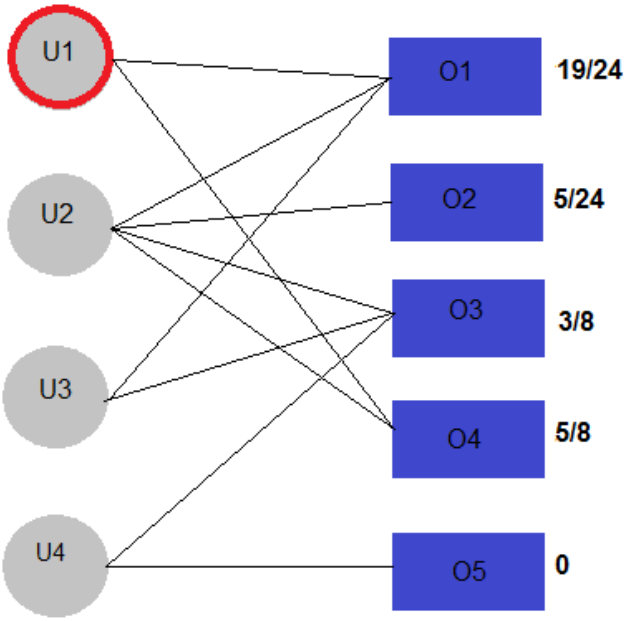
$$f(U_1) = \frac{a_{11}}{\kappa_1}a_{11} + \frac{a_{14}}{\kappa_4}a_{14} = \frac{1}{3}\times 1 + \frac{1}{2}\times 1 = \frac{5}{6}$$

$$f(U_2) = \frac{a_{21}}{\kappa_1}a_{11} + \frac{a_{22}}{\kappa_2}a_{12} + \frac{a_{23}}{\kappa_3}a_{13} + \frac{a_{24}}{\kappa_4}a_{14} = \frac{1}{3}\times 1 + \frac{1}{1}\times 0 + \frac{1}{3}\times 0 + \frac{1}{2}\times 1 = \frac{5}{6}$$

$$f(U_3) = \frac{a_{31}}{\kappa_1}a_{11} + \frac{a_{33}}{\kappa_3}a_{13} = \frac{1}{3}\times 1 + \frac{1}{3}\times 0 = \frac{1}{3}$$

$$f(U_4) = \frac{a_{43}}{\kappa_3}a_{13} + \frac{a_{45}}{\kappa_5}a_{15} = \frac{1}{3}\times 0 + \frac{1}{1}\times 0 = 0$$

Στη συνέχεια το σύνολο των πόρων κάθε χρήστη ρέει πίσω στις ταινίες (*Εικόνα 5)* όπου και διαμορφώνεται η τελική τιμή του καθενός προκειμένου να εξαχθεί και η ζητούμενη λίστα προτάσεων.



Εικόνα 5: Τελικό βήμα αλγορίθμου ProbS.

Προκειμένου να καταλήξουμε στο αποτέλεσμα της *Εικόνας 5* εργαζόμαστε σύμφωνα με τον τύπο: [7]

$$f(O_a) = \sum_{i=1}^{m} \frac{a_{ia} f(U_i)}{\kappa_i} \quad (7)$$

Επομένως αντίστοιχα για κάθε ταινία θα έχουμε :

$$f(O_1) = \frac{a_{11} f(U_1)}{\kappa_1} + \frac{\alpha_{21} f(U_2)}{\kappa_2} + \frac{\alpha_{31} f(U_3)}{\kappa_3} = \frac{\frac{5}{6}}{2} + \frac{\frac{5}{6}}{4} + \frac{\frac{1}{3}}{2} = \frac{19}{24}$$

$$f(O_2) = \frac{\alpha_{22} f(U_2)}{\kappa_2} = \frac{\frac{5}{6}}{4} = \frac{5}{24}$$

$$f(O_3) = \frac{a_{23} f(U_2)}{\kappa_2} + \frac{\alpha_{33} f(U_3)}{\kappa_3} + \frac{\alpha_{43} f(U_4)}{\kappa_4} = \frac{\frac{5}{6}}{4} + \frac{\frac{1}{3}}{2} + \frac{0}{2} = \frac{3}{8}$$

$$f(O_4) = \frac{a_{14} f(U_1)}{\kappa_1} + \frac{\alpha_{24} f(U_2)}{\kappa_2} = \frac{\frac{5}{6}}{2} + \frac{\frac{5}{6}}{4} = \frac{5}{8}$$

$$f(O_5) = \frac{\alpha_{45} f(U_4)}{\kappa_4} = \frac{0}{2} = 0$$

Τέλος, με την ολοκλήρωση και του τελευταίου βήματος (*Εικόνα 5)* καταλήγουμε και στην τελική αποτίμηση των πόρων κάθε ταινίας. Ο ProbS συγκριτικά με τον HeatS παρατηρούμε ότι θα συνιστούσε πρώτα την ταινία O_3 η οποία είναι δημοφιλέστερη, αφού προτιμάται από μεγαλύτερο αριθμό χρηστών, συγκριτικά με την O_2 που προτιμάται απο τον HeatS και δεν είναι τόσο δημοφιλής.

2.3 Hybrid Spreading algorithm (HybridS)

Σε μια προσπάθεια βελτίωσης των παραπάνω προτάθηκε ένας υβριδικός αλγόριθμος [6], ο οποίος συνδυάζει χαρακτηριστικά και από τους δυο αλγορίθμους. Βασικός σκοπός του αλγορίθμου να εκμεταλλευτεί την ακρίβεια που προσφέρει ο ProbS και την ποικιλομορφία των αποτελεσμάτων του HeatS.

Όπως αναφέρθηκε ο HeatS έχει την τάση να προωθεί αντικείμενα λιγότερο δημοφιλή σε αντίθεση με τον ProbS. Ως αποτέλεσμα, οι προτάσεις εκείνες του HeatS καθίστανται πολλές φορές αρκετά ιδιόμορφες, ενώ του ProbS παρουσιάζουν έλλειψη ως προς την ποικιλομορφία τους. Για αυτό το λόγο προτάθηκε ο HybridS, συνδυάζοντας και τους δυο αλγορίθμους στο μητρώο ανακατανομής των πόρων W, έχοντας την εξής μορφή:

$$W_{\alpha\beta}^{H+P} = \frac{1}{\kappa_\alpha^{1-\lambda} \kappa_\beta^{\lambda}} \sum_{i=1}^{m} \frac{\alpha_{i\alpha} \alpha_{i\beta}}{\kappa_i} \quad (8)$$

Όπου για $\lambda = 1$ έχουμε τον αλγόριθμο ProbS, ενώ για $\lambda = 0$ τον HeatS. Ρυθμίζοντας κατάλληλα την τιμή του λ ο αλγόριθμος προσαρμόζεται ανάλογα με τις απαιτήσεις κάθε συστήματος. Η τιμή του λ δεν είναι απαραίτητο να είναι πάντα ίδια και μπορεί να διαφέρει για κάθε χρήστη ή αντικείμενο ανάλογα με την κατάσταση του συστήματος και κατά πόσο τα αποτελέσματα που παράγονται είναι βέλτιστα.

2.4 Weighted Slope One (WSO)

Αποτελεί έναν από τους πιο δημοφιλής αλγορίθμους στο συνεργατικό φιλτράρισμα, έχοντας ως πλεονέκτημα την απλότητα και την ευκολία υλοποίησής του. Ο Weighted

Slope One βασίζεται στην ιδέα της πρόβλεψης βαθμολογιών για τους χρήστες συνδυάζοντας παρόμοιες προηγούμενες βαθμολογίες τους με αυτές άλλων χρηστών.

Θα μπορούσαμε να θεωρήσουμε ότι προσεγγίζει το πρόβλημα σε δυο μέρη. Αρχικά γίνεται ο υπολογισμός των αποκλίσεων μεταξύ κάθε ζεύγους ταινιών και στη συνέχεια γίνεται η πρόβλεψη των βαθμολογίων.

Προκειμένου να γίνει ο υπολογισμός των αποκλίσεων εργαζόμαστε ως εξής:

$$dev_{i,j} = \sum_{u \in S_{i,j}(X)} \frac{u_i - u_j}{card(S_{i,j}(X))} \quad (9)$$

όπου X είναι ο πίνακας με τις βαθμολογίες κάθε χρήστη από το 1 έως 5, $card(S)$ είναι το σύνολο των στοιχείων στο S και επομένως $card(S_{i,j}(X))$ είναι το σύνολο των χρηστών που έχουν βαθμολογήσει εξίσου και την ταινία j και την i. Δεδομένων των ειπωθέντων ο υπολογισμός των αποκλίσεων για τον *Πίνακα 1* διαμορφώνεται ως εξής:

Πίνακας 1: Βαθμολογίες χρηστών ανά ταινία.

	Movie 1	Movie 2	Movie 3
User 1	3	4	3
User 2	4	2	?
User 3	3	?	5
User 4	?	5	2

Θέλοντας να υπολογίσουμε την απόκλιση της ταινίας 2 ως προς την ταινία 1, θα έχουμε στον παρανομαστή $card\left(S_{i,j}(X)\right) = 2$ αφού 2 χρήστες έχουν βαθμολογήσει και τις 2 ταινίες. Ο αριθμητής στη συνέχεια είναι η διαφορά των μεταξύ τους βαθμολογιών, δηλαδή:

$$dev_{movie1,movie2} = \frac{3-4}{2} + \frac{4-2}{2} = 0.5$$

Έτσι η απόκλιση από την ταινία 2 ως προς την 1 είναι 0.5 πράγμα που σημαίνει πως οι χρήστες βαθμολόγησαν την πρώτη κατά 0.5 βαθμό καλύτερα. Αν τώρα θέλαμε να υπολογίσουμε την απόκλιση από την ταινία 1 ως προς την ταινία 2 θα είχαμε:

$$dev_{movie2,movie1} = \frac{4-3}{2} + \frac{2-4}{2} = -0.5$$

Εργαζόμενοι όμοια για κάθε συνδυασμό ταινίας καταλήγουμε στον *Πίνακα 2.*

Πίνακας 2: Αποκλίσεις κάθε ζεύγους ταινιών.

	Movie 1	Movie 2	Movie 3
Movie 1	0	0.5	-1
Movie 2	-0.5	0	2
Movie 3	1	-2	0

Έχοντας υπολογίσει το σύνολο των αποκλίσεων, επόμενο βήμα είναι ο υπολογισμός των προβλέψεων των βαθμολογιών για κάθε χρήστη σε ταινίες που δεν έχει βαθμολογήσει. Αυτό εύκολα επιτυγχάνεται ακολουθώντας τον παρακάτω τύπο:

$$P(U)_j = \frac{\sum_{i \in S(U)-\{j\}} (dev_{j,i} + u_i) card(S_{j,i}(X))}{\sum_{i \in S(U)-\{j\}} card(S_{j,i}(X))} \quad (10)$$

όπου $P(U)_j$ η πρόβλεψη βαθμολογίας για τον χρήστη U για την ταινία j. Αναλύοντας τον αριθμητή ο όρος $\sum_{i \in S(U)-\{j\}} (dev_{j,i} + u_i)$ σημαίνει πως για κάθε ταινία που ο χρήστης U έχει βαθμολογήσει (εκτός από την ταινία j) θα ψάξουμε την απόκλιση της ταινίας j σε αυτή την ταινία και θα την προσθέσουμε στη βαθμολογία του χρήστη U για την ταινία i. Έπειτα, πολλαπλασιάζουμε με $card(S_{j,i}(X))$, δηλαδή με το πλήθος των χρηστών που έχουν βαθμολογήσει και τις δυο ταινίες i και j.

Έστω λοιπόν ότι θέλουμε να εξάγουμε μια πρόβλεψη για τη βαθμολογία του χρήστη 2 για την ταινία 3. Σύμφωνα με τον αριθμητή, για κάθε ταινία που ο χρήστης 2 έχει βαθμολογήσει, πέραν προφανώς της ταινίας 3 της οποίας και τη βαθμολογία θέλουμε να προβλέψουμε, θα προσθέσουμε την απόκλιση της ταινίας αυτής ως προς την ταινία 3 με τη βαθμολογία του χρήστη 2 για αυτή και στη συνέχεια θα πολλαπλασιάσουμε με το πλήθος των χρηστών που έχουν βαθμολογήσει και τις δύο ταινίες. Ο χρήστης 2 έχει βαθμολογήσει την ταινία 1 με $u_1 = 4$, η απόκλιση της ταινίας 3 ως προς την 1 σύμφωνα με τον *Πίνακα 2* είναι ίση με $dev_{3,1} = 1$ και παρατηρώντας τον *Πίνακα 1* βλέπουμε πως το πλήθος των χρηστών που βαθμολόγησαν και τις δυο ταινίες 1 και 3 είναι ίσο με $card\left(S_{1,3}(X)\right) = 2$. Επιπλέον, ο χρήστης 2 έχει βαθμολογήσει και την ταινία 2 με $u_2 = 2$, η απόκλιση της ταινίας 3 ως προς την 2 σύμφωνα με τον *Πίνακα 2* είναι ίση με $dev_{3,2} = -2$ και από τον *Πίνακα 1* βλέπουμε πως το πλήθος των χρηστών που βαθμολόγησαν και τις δυο ταινίες 3 και 2 είναι ίσο με $card\left(S_{2,3}(X)\right) = 2$. Άρα από τα παραπάνω προκύπτει ότι:

$$P(U_2)_{movie3} = \frac{(dev_{3,1} + u_1) card\left(S_{1,3}(X)\right) + (dev_{3,2} + u_2) card\left(S_{2,3}(X)\right)}{card\left(S_{1,3}(X)\right) + card\left(S_{2,3}(X)\right)}$$
$$= \frac{(1+4)2 + (-2+2)2}{4} = 2.5$$

Επομένως ο χρήστης 2 με βάση τον WSO θα βαθμολογούσε την ταινία με 2.5.

2.5 Local Sensitive Hashing (LSH)

Ο αλγόριθμος LSH στοχεύει στη μείωση της διάστασης πολυδιάστατων δεδομένων. Αυτό επιτυγχάνεται στο γεγονός ότι κατακερματίζει τα δεδομένα τοποθετώντας παρόμοια δείγματα αυτών, με ίδια τιμή κατακερματισμού, στους λεγόμενους κάδους. Υπάρχουν διάφορα είδη αλγορίθμων LSH για την προσέγγιση διαφορετικών αποστάσεων όπως π.χ. ο bit-sampling LSH, ο LSH με απόσταση Hamming, ο min-hash LSH κ.α. Ανάμεσα σε αυτούς υπάρχουν κάποια είδη του αλγορίθμου που επεξεργάζονται δυαδικά δεδομένα. Αυτό συνήθως γίνεται με την προσέγγιση κάποιας συγκεκριμένης απόστασης ή ομοιότητας μεταξύ των δειγμάτων που εξετάζουμε, τα οποία αναπαρίστανται ως διανύσματα σε δυαδική μορφή. Αυτό προϋποθέτει κυρίως bitwise πράξεις καθιστώντας τη διαδικασία υπολογιστικά γρήγορη. Ειδικά σε εφαρμογές όπου ερχόμαστε αντιμέτωποι με μεγάλο πλήθος δεδομένων ο LSH αποτελεί αποδοτική λύση τόσο από άποψη χρονικής βελτίωσης της επεξεργασίας των δεδομένων, αλλά και της παραγωγής των αποτελεσμάτων.

Στην παρούσα πτυχιακή θα χρησιμοποιηθεί ο αλγόριθμος LSH-SuperBit, ο οποίος προτάθηκε από τους Jianqiu Ji, Jianmin Li, Shuicheng Yan, Bo Zhang, Qi Tian [5] και αντίστοιχη υλοποίησή του βρέθηκε https://github.com/tdebatty/java-LSH.

Σύμφωνα με [8], ένα σύνολο συναρτήσεων κατακερματισμού ανήκει σε ένα σύνολο συναρτήσεων κατακερματισμού τοπικής ευαισθησίας F αν για δύο στοιχεία δεδομένων x, y:

$$P_{h \in F}[h(x) = h(y)] = sim(x, y) \quad (11)$$

όπου P η πιθανότητα και $sim(x, y)$ η συνάρτηση ομοιότητας που θα χρησιμοποιηθεί στο σύνολο των δεδομένων.

Ο SuperBit-LSH αποτελεί μια βελτιωμένη έκδοση εκείνου του Random Projection-LSH και υπολογίζει μια εκτίμηση της συνημιτονοειδούς απόστασης μεταξύ των δεδομένων. Ο Random Projection-LSH λειτουργεί προσεγγίζοντας την συνημιτονοειδή απόσταση μεταξύ διανυσμάτων. Ο αλγόριθμος επιλέγει ένα τυχαίο υπερεπίπεδο (hyperplane) το οποίο χρησιμοποιεί ως σημείο αναφοράς για τον κατακερματισμό των διανυσμάτων που θα δοθούν ως είσοδος, εξετάζοντας σε ποια πλευρά του υπερεπιπέδου βρίσκεται το καθένα από αυτά. Δοθέντος λοιπόν ενός διανύσματος v και ενός υπερεπιπέδου r ορίζουμε ως συνάρτηση κατακερματισμού:

$$h(v) = sgn(v \cdot r) = \begin{cases} 1, & \alpha v\ r \cdot v \geq 0 \\ 0, & \alpha v\ r \cdot v < 0 \end{cases} \quad (12)$$

Δηλαδή, $h(v) = \pm 1$ ανάλογα σε ποια πλευρά του υπερεπιπέδου βρίσκεται το v διάνυσμα. Έχοντας λοιπόν δύο διανύσματα u, v εξετάζουμε το βαθμό ομοιότητάς τους χρησιμοποιώντας την εξίσωση:

$$\Pr[h(u) = h(v)] = 1 - \frac{\theta(u,v)}{\pi} \quad (13)$$

όπου $\theta(u,v)$ η γωνία που σχηματίζουν μεταξύ τους τα διανύσματα u,v και $1-\frac{\theta(u,v)}{\pi}$ η συνημιτονοειδής ομοιότητα αφού από [9] ισχύει:

$$\cos(\theta(u,v)) = \cos(\pi(1 - P_{h\in F}[h(u) = h(v)])) \text{ (14)}$$

Αντίστοιχα, στον Superbit-LSH επιλέγονται K ανεξάρτητα τυχαία διανύσματα $\{v_1, v_2, \dots, v_K\}$ με Γκαουσιανή κατανομή (Gaussian or normal distribution) τα οποία χωρίζονται σε L παρτίδες αποτελούμενες από N διανύσματα η καθεμία. Στη συνέχεια με την εκτέλεση ορθογωνιοποίησης Gram-Schmidt (Gram-Schmidt Orthogonalization) σε αυτές τις L παρτίδες N διανυσμάτων αντίστοιχα, παίρνουμε $K = N \times L$ προβολές διανυσμάτων $\{w_1, w_2, \dots w_K\}$. Αυτό έχει ως αποτέλεσμα τη δημιουργία K συναρτήσεων του Superbit-LSH $\left(h_{w_1}, h_{w_2,} \dots, h_{w_K}\right)$, όπου αντίστοιχα με την εξίσωση του Random Projection-LSH (12) δοθέντος ενός διανύσματος x έχουμε [5]:

$$h_{w_i}(x) = sgn(w_i^T \cdot x) \text{ (15)}$$

Με αυτό τον τρόπο εξάγουμε για κάθε ένα από τα διανύσματα που έχουμε ως είσοδο μια υπογραφή (signature), ως προς τις προβολές των διανυσμάτων που παράχθηκαν από των Superbit-LSH. Στη συνέχεια, για κάθε ζεύγος διανυσμάτων γίνεται σύγκριση των υπογραφών του βρίσκοντας τη μεταξύ τους συνημιτονοειδή ομοιότητα από όπου και θα προκύψει σε τελική φάση ο διαχωρισμός των διανυσμάτων σε κάδους. Η συνημιτονοειδής ομοιότητα είναι η γωνιακή διαφορά μεταξύ δυο διανυσμάτων και εκφράζεται από τον τύπο:

$$Sim = \cos(\theta) = \frac{A*B}{||A||||B||} \text{ (16)}$$

Επομένως, δυο διανύσματα με τον ίδιο προσανατολισμό θα έχουν συνημιτονοειδή ομοιότητα ίση με 1, εκείνα που θα σχηματίζουν γωνία 90° ίση με 0 και τέλος εκείνα που θα είναι αντίθετα διαμετρικά μεταξύ τους ίση με -1. Πιο συγκεκριμένα το σύνολο των διανυσμάτων το οποίο θα επεξεργαστούμε θα είναι της μορφής:

$$\sum_{i=1}^{n} d_i = \{d_{1,} d_2, \dots, d_m\} \text{ (17)}$$

όπου n είναι το πλήθος των χρηστών και m το πλήθος των ταινιών. Κάθε χρήστης λοιπόν αντικατοπτρίζεται από ένα διάνυσμα αποτελούμενο από 0 και 1 ανάλογα με το ποιες ταινίες του άρεσαν ή όχι. Με τη σύγκριση αυτών των διανυσμάτων χρησιμοποιώντας τον Superbit-LSH καταφέρνουμε να ομαδοποιήσουμε τους όμοιους χρήστες μεταξύ τους και να τους τοποθετήσουμε σε ίδιους κάδους ανάλογα με την υπογραφή του καθενός, ως συνέπεια την επίτευξη μιας πιο γρήγορης εκτέλεσης του προβλήματος αλλά και πιο ακριβείς προβλέψεις αφού το αποτέλεσμα προκύπτει λαμβάνοντας υπόψη παρόμοιους χρήστες.

3. ΜΕΤΡΙΚΕΣ ΑΞΙΟΛΟΓΗΣΗΣ

Στο παρόν κεφάλαιο θα εξετάσουμε με ποιο τρόπο μπορούμε να αξιολογήσουμε τους προαναφερθέντες αλγορίθμους ως προς την αποδοτικότητα, την εγκυρότητα και την αποτελεσματικότητά τους. Για το σκοπό αυτό έχουν προταθεί διάφορες μετρικές αξιολόγησης. Ωστόσο αυτές που θα χρησιμοποιηθούν είναι: η ακρίβεια (precision), η ανάκληση (recall), η μέση ακρίβεια (average precision) και ο σταθμισμένος αρμονικός μέσος (F1 metric).

Παρακάτω θα αναλύσουμε κάθε μια από τις μετρικές αυτές ξεχωριστά, ωστόσο για να γίνει πιο κατανοητή η συμπεριφορά τους χρειάζεται να διαχωρίσουμε το είδος κάθε πρόβλεψης που πρόκειται να εξαχθεί από το σύστημα προτάσεων όπως φαίνεται στον Πίνακα 3.

Πίνακας 3: Είδη προβλέψεων.

		Πραγματικότητα		
Πρόβλεψη		Πραγματικά καλή	Πραγματικά κακή	
	Έχει προβλεφθεί ως καλή	True positive (tp)	False positive (fp)	**Όλες οι προτάσεις**
	Έχει προβλεφθεί ως κακή	False negative (fn)	True negative (tn)	
		Όλες οι καλές προτάσεις		

Από τον Πίνακα 3 επομένως μπορούμε να συμπεράνουμε πως μια ταινία η οποία θα προταθεί από το σύστημά μας θα ανήκει σε μια από τις εξής τέσσερις κατηγορίες:

- Να έχει προταθεί και να αρέσει στο χρήστη: *True Positive (tp).*
- Να έχει προταθεί και να μην αρέσει στο χρήστη: *False Positive (fp).*
- Να μην έχει προταθεί και να είναι στις προτιμήσεις του χρήστη: *False Negative (fn).*
- Να μην έχει προταθεί και να μην προτιμάται από το χρήστη: *True Negative (tn).*

Για κάθε χρήστη i το σύστημα προτάσεων θα συγκεντρώσει και θα ταξινομήσει όλες τις ταινίες που δεν έχει δει με φθίνουσα σειρά προτίμησης και θα του προτείνει τις πρώτες καλύτερες. Προκειμένου να γίνει η αξιολόγηση αυτών των προτάσεων έχουμε χωρίσει το δείγμα μας σε δύο μέρη: το πρώτο μέρος (train set) με το οποίο θα εκπαιδεύσουμε τους αλγορίθμους μας ώστε να δημιουργηθεί ένα μοντέλο συμπεριφοράς για κάθε χρήστη και το δεύτερο μέρος (test set) πάνω στο οποίο θα εξετάσουμε κατά πόσο οι προβλέψεις μας ήταν σωστές. Το δείγμα μας έχει χωριστεί σε 5 τέτοια ζεύγη όπου το 80% αποτελεί το σύνολο των δεδομένων που θα δοθούν για εκπαίδευση στους

αλγορίθμους και το 20% το σύνολο εκείνων ως προς τα οποία θα τους αξιολογήσουμε. Σημαντικό είναι να σημειωθεί ότι τα δύο σύνολα αυτά των δεδομένων, train και test set, είναι ξένα μεταξύ τους με σκοπό την αποφυγή υπερπροσαρμογής των αλγορίθμων (overfitting) σε αυτά και εξαγωγής πιο αξιόπιστων αποτελεσμάτων.

Ο λόγος ο οποίος έχουμε διασπάσει τα δεδομένα μας σε αυτά τα 5 ζεύγη οφείλεται στην εφαρμογή μιας τεχνικής η οποία ονομάζεται 5-fold cross validation. Σύμφωνα με αυτή για κάθε ένα από αυτά τα 5 ζεύγη θα εκτελέσουμε τους αλγορίθμους αρχικά δίνοντάς τους το train set, όπου κάθε αλγόριθμους θα εκπαιδευτεί για τις προτιμήσεις κάθε χρήστη και θα εξάγει για τον καθένα από μια λίστα με προτάσεις ταινιών, την οποία στη συνέχεια θα συγκρίνουμε με το test set. Θα συγκρίνουμε κατά πόσο οι ταινίες που προβλέψαμε με βάση το train set συμπίπτουν με αυτές του test set, δηλαδή άρεσαν όντως στο χρήστη. Εν ολίγοις, με τη μέθοδο του cross validation συνδυάζονται τα σφάλματα προβλέψεων και οι μέσοι όροι των μετρικών προκειμένου να αποκομίσουμε μια πιο ακριβή εκτίμηση της απόδοσης του μοντέλου πρόβλεψης.

3.1 Ακρίβεια (Precision)

Η ακρίβεια αποτελεί ένα μέτρο εγκυρότητας, καθορίζεται από το λόγο των σχετικών αντικειμένων που προτείνονται, δηλαδή των αντικειμένων που άρεσαν στο χρήστη, ως προς όλο το σύνολο των αντικειμένων που προτείνονται. Δηλαδή στην περίπτωσή μας ουσιαστικά είναι το σύνολο των ταινιών που ήταν πραγματικά καλές, άρεσαν στο χρήστη.

$$Precision = \frac{tp}{tp+fp} = \frac{|\text{όλες οι ταινιές που προτάθηκαν και ήταν καλές}|}{|\text{όλες οι ταινίες που προτάθηκαν}|} \quad (18)$$

Για παράδειγμα αν προβλέπαμε για κάποιο χρήστη 20 ταινίες εκ των οποίων οι 15 ήταν σύμφωνες με τις προτιμήσεις του τότε το σύστημά μας θα είχε ακρίβεια ίση με 75%.

3.2 Ανάκληση (Recall)

Η ανάκληση αποτελεί ένα μέτρο πληρότητας, καθορίζεται από το λόγο των σχετικών αντικειμένων που προτάθηκαν ως προς όλα τα σχετικά αντικείμενα. Συγκεκριμένα είναι ο λόγος των ταινιών που προτάθηκαν και ήταν καλές ως προς όλο το σύνολο καλών ταινιών, δηλαδή το σύνολο των ταινιών που αρέσουν στο χρήστη.

$$Recall = \frac{tp}{tp+fn} = \frac{|\text{όλες οι ταινιές που προτάθηκαν και ήταν καλές}|}{|\text{όλες οι καλές ταινίες}|} \quad (19)$$

Για παράδειγμα αν σε ένα χρήστη αρέσουν 20 ταινίες και από αυτές τις 20 το σύστημα μας του πρότεινε τις 13 από αυτές τότε θα είχαμε ανάκληση ίση με 65%.

Συνήθως όταν ένα σύστημα προτάσεων ρυθμίζεται έτσι ώστε να αυξάνεται η ακρίβεια του, ως αποτέλεσμα μειώνεται η ανάκλησή του και το αντίστροφο.

3.3 Μέση ακρίβεια (Average Precision)

Η μέση ακρίβεια αποτελεί μια μετρική ακρίβειας της κατάταξης (ranked precision metric) των προβλέψεων η οποία δίνει έμφαση στη θέση εμφάνισης των σωστών προβλέψεων στη λίστα των προτάσεων. Οι υψηλά καταταγμένες αυτές προβλέψεις εκφράζονται με τον όρο *hits*.

$$Average\ Precision = \frac{1}{n}(\frac{1}{i} + \cdots + \frac{1}{i+j}) \quad (20)$$

όπου n είναι το πλήθος των σωστών προβλέψεων και i, j ακέραιοι που αντικατοπτρίζουν τη θέση τους στη λίστα προτάσεων.

Έστω δηλαδή ότι έχουμε τις προβλέψεις του Πίνακα 4:

Πίνακας 4: Κατάταξη σωστών προβλέψεων στη λίστα προτάσεων.

Κατάταξη	Hit
1	
2	**X**
3	**X**
4	
5	**X**

Στον Πίνακα 4 με **X** συμβολίζεται η σωστή πρόβλεψη στη λίστα και δίπλα ο αριθμός της θέσης στην οποία βρίσκεται. Έχοντας αυτά τα δεδομένα μπορούμε να υπολογίσουμε τη μέση ακρίβεια η οποία θα είναι:

$$Average\ Precision = \frac{1}{3}\left(\frac{1}{2} + \frac{2}{3} + \frac{3}{5}\right) \simeq 0.58$$

3.4 Σταθμισμένος αρμονικός μέσος (F_1 measure)

Ο σταθμισμένος αρμονικός μέσος συνδυάζει την ακρίβεια και την ανάκληση σε μια ενιαία τιμή. Χρησιμοποιείται για την απόκτηση μιας πιο ισορροπημένης εικόνας της απόδοσης του συστήματος. Στόχος κάθε συστήματος είναι να κρατά όσο το δυνατόν γίνεται αυτά τα ποσοστά σε υψηλό επίπεδο, αν και όπως προαναφέρθηκε συνήθως η ακρίβεια και η ανάκληση παρουσιάζουν μια αντίστροφη σχέση.

$$F_\beta = (1+\beta^2)\cdot\frac{precision\cdot recall}{\beta^2\cdot precision+recall} \text{ (21)}$$

Ο F_1 σταθμισμένος αρμονικός μέσος δίνει ισόποση βαρύτητα στην ακρίβεια και στην ανάκληση και για αυτό το λόγο η τιμή της παραμέτρου β είναι ίση με 1. Ωστόσο συνηθισμένες είναι και οι περιπτώσεις όπου χρησιμοποιούνται οι $F_2, F_{0.5}$ με τον πρώτο να δίνει μεγαλύτερη βαρύτητα στην ανάκληση και τον δεύτερο να δίνει έμφαση στην ακρίβεια, θέτοντας τις τιμές του β ίσες με 2 και 0.5 αντίστοιχα.

3.5 Ποικιλομορφία (Diversity)

Η ποικιλομορφία στα συστήματα προτάσεων εξετάζει το πόσο διαφορετικά είναι μεταξύ τους τα αντικείμενα που προτείνονται. Στο παρόν σύστημα προτάσεων θα εξετάσουμε κατά πόσο οι αλγόριθμοί μας είναι δυνατόν να παρέχουν ποικίλα αντικείμενα σε κάθε χρήστη.

Η ποικιλομορφία χαρακτηρίζεται από το μέσο όρο της διαφορετικότητας των αντικειμένων που βρίσκονται στη λίστα προτάσεων του χρήστη. Η ομοιότητα μεταξύ δυο αντικειμένων μπορεί να μετρηθεί με τη χρήση του ευρετηρίου Sørensen.

$$S_{\alpha\beta} = \frac{2a}{2a+b+c} \text{ (22)}$$

Όπου α και β το σύνολο των αντικειμένων που συγκρίνουμε μεταξύ τους, a το πλήθος των αντικειμένων που βρίσκονται και στα δυο δείγματα, b το πλήθος των αντικειμένων που βρίσκονται στο δείγμα B αλλά όχι στο A, c το πλήθος των αντικείμενων που βρίσκονται στο A και όχι στο B και τέλος d εκείνα που λείπουν και από τα δυο δείγματα.

Η ποικιλομορφία των αντικειμένων του χρήστη i ορίζεται ως εξής:

$$D_i = \frac{1}{L(L-1)}\sum_{\alpha\neq\beta}(1-S_{ab}) \text{ (15)}$$

Όπου L το μήκος της λίστας προτάσεων και S_{ab} ο συντελεστής Sørensen.

3.6 Καινοτομία (Novelty)

Η καινοτομία στα συστήματα προτάσεων αναφέρεται στο πόσο διαφορετικά τα αντικείμενα που προτείνονται είναι συγκριτικά με το τι έχουν ήδη προτιμήσει οι χρήστες. Αυτό μπορεί να υπολογιστεί αρκετά εύκολα μετρώντας τη μέση δημοτικότητα κάθε αντικειμένου που έχει προταθεί.

$$N = \frac{1}{ML}\sum_{i=1}^{M}\sum_{a\in R_i}\kappa_\alpha \text{ (16)}$$

Όπου M είναι το πλήθος των χρηστών, L το μήκος της λίστας προτάσεων, R_i η λίστα προτάσεων του χρήστη i και κ_α ο βαθμός του αντικείμενου α. Όσο πιο χαμηλή η δημοτικότητα των αντικειμένων τόσο πιο υψηλή είναι η καινοτομία των αποτελεσμάτων.

4. ΑΞΙΟΛΟΓΗΣΗ ΑΛΓΟΡΙΘΜΩΝ

Χρησιμοποιώντας τις μετρικές αξιολογήσεις που αναφέραμε στο κεφάλαιο 3 θα συγκρίνουμε τους αλγορίθμους ώστε να εξετάσουμε την απόδοση και αποτελεσματικότητά τους. Αρχικά να αναφέρουμε ότι η βάση στην οποία θα εξετάσουμε τους αλγορίθμους αποτελείται από 943 χρήστες, 1682 ταινίες και 100000 βαθμολογίες χρηστών σε ταινίες. Το δείγμα μας παρουσιάζει αραιότητα (sparsity) 93.6% αρχικά αλλά δεδομένου ότι οι αλγόριθμοί μας εξετάζουν δυαδικές τιμές οι βαθμολογίες των χρηστών μειώνονται αφού κρατάμε εκείνες που είναι μεγαλύτερες ή ίσες του 3. Έτσι η αραιότητα του δείγματός μας διαμορφώνεται όπως φαίνεται στον Πίνακα 5.

Πίνακας 5: Δομή του dataset.

	Movielens (ratings 1-5)	Movielens (binary ratings)
Items	1682	1682
Ratings	100000	66103
Sparsity	93.69%	95.83%
Ratings/user	106.04	70.09

4.1 Αξιολόγηση HeatS και ProbS

Οι δύο κύριοι αλγόριθμοι που υλοποιήθηκαν και έγινε προσπάθεια για βελτίωσή τους στην παρούσα πτυχιακή είναι ο HeatS και ο ProbS. Θα εξετάσουμε τη συμπεριφορά τους χωρίς κάποια βελτίωση και πως διαφοροποιούνται μεταβάλλοντας το μέγεθος της εξαγόμενης λίστας προτάσεων.

Πίνακας 6:Αξιολόγηση HeatS και ProbS.

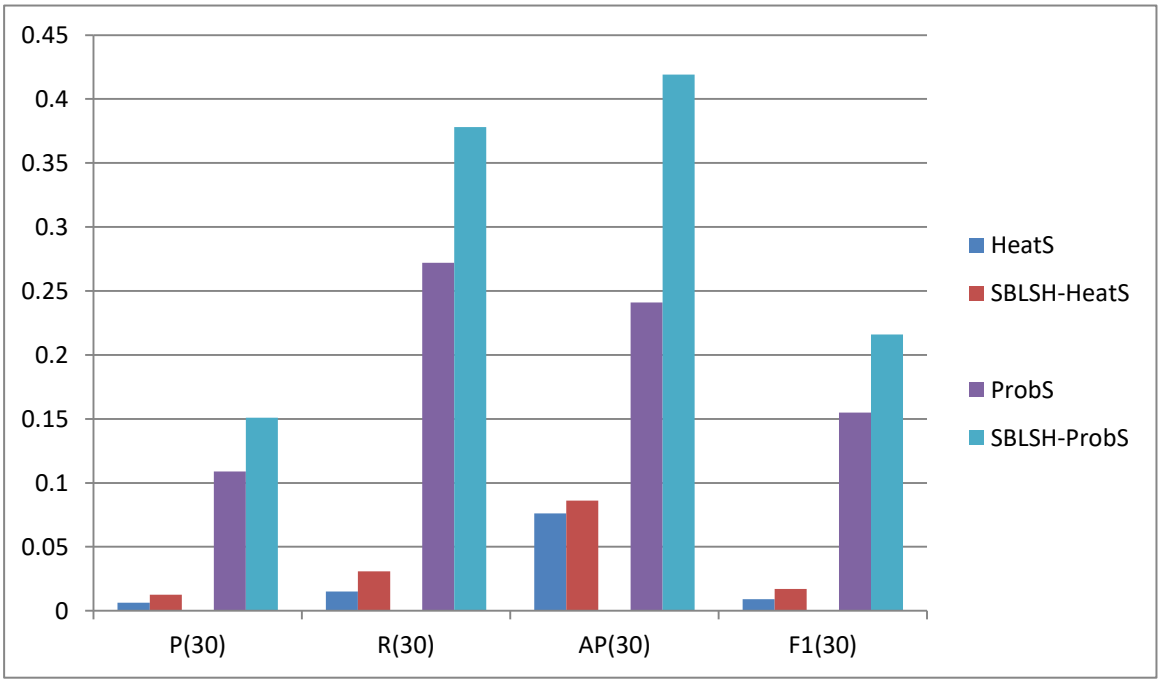
	P(30)	P(100)	R(30)	R(100)	AP(30)	AP(100)	F1(30)	F1(100)	D(30)
HeatS	0.0063	0.019	0.015	0.114	0.076	0.046	0.009	0.033	0.950
ProbS	0.109	0.071	0.272	0.419	0.373	0.241	0.155	0.121	0.907

Παρατηρώντας τον Πίνακα 6 συμπεραίνουμε ότι και οι δυο αλγόριθμοι δεν είναι τόσο αποδοτικοί στο παρόν δείγμα δεδομένων και χωρίς την προσθήκη κάποιας βελτίωσης σε αυτούς. Μην ξεχνάμε βέβαια ότι το δείγμα μας είναι αρκετά αραιό πράγμα που επηρεάζει σε μεγάλο βαθμό τη λειτουργικότητα των αλγορίθμων. Αυξάνοντας το μέγεθος της λίστας προτάσεων παρατηρούμε ότι αυξάνεται η ανάκληση που σημαίνει ότι πετυχαίνουμε περισσότερες ταινίες που άρεσαν στο χρήστη, αλλά μειώνεται η

ακρίβεια και η μέση ακρίβεια έχοντας ως αποτέλεσμα στο δείγμα να παρουσιάζονται αρκετά αποτελέσματα τα οποία δε θα ενδιαφέρουν το χρήστη. Ενδιαφέρον παρουσιάζει το γεγονός ότι ο HeatS εμφανίζει μεγαλύτερη ποικιλία στις προτάσεις του αφού ξεπερνάει τον ProbS κατά 5%. Αυτό σημαίνει ότι οι προτάσεις του αποτελούνται από ταινίες οι οποίες δεν είναι παρόμοιες μεταξύ τους. Ωστόσο ο ProbS εμφανίζεται σίγουρα πιο αποδοτικός από τον HeatS πράγμα που μας προδιαθέτει στο ότι οι δημοφιλέστερες ταινίες έχουν μεγαλύτερη αποδοχή από τους χρήστες του δείγματός μας συγκριτικά με τα πιο εξειδικευμένα αποτελέσματα που παράγονται από τον HeatS.

4.2 Βελτίωση HeatS και ProbS με τη χρήση SuperBit-LSH

Η πρώτη σκέψη για τη βελτίωση των παραπάνω αλγορίθμων ήταν ο περιορισμός του δείγματος σε χρήστες οι οποίοι παρουσιάζουν ομοιότητες μεταξύ τους. Αρχικά έγινε μια προσπάθεια χρήσης του K-Nearest Neighbor αλγορίθμου με τη χρήση συνημιτονοειδούς ομοιότητας. Ωστόσο παρόλη τη βελτίωση των αποτελεσμάτων, αποτέλεσε μια εξαιρετικά χρονοβόρα και εξαντλητική μέθοδο. Μετά από προτροπή του επιβλέποντα ο αλγόριθμος που προτιμήθηκε για να ικανοποιήσει το σκοπό αυτό είναι εκείνος του SuperBit-LSH. Έτσι αντί να εξετάζουμε ολόκληρο το δείγμα χρηστών περιοριζόμαστε σε όσους χρήστες βρίσκονται στον ίδιο κάδο.



Σχήμα 1: Σύγκριση HeatS, ProbS με SBLSH-HeatS και SBLSH-ProbS.

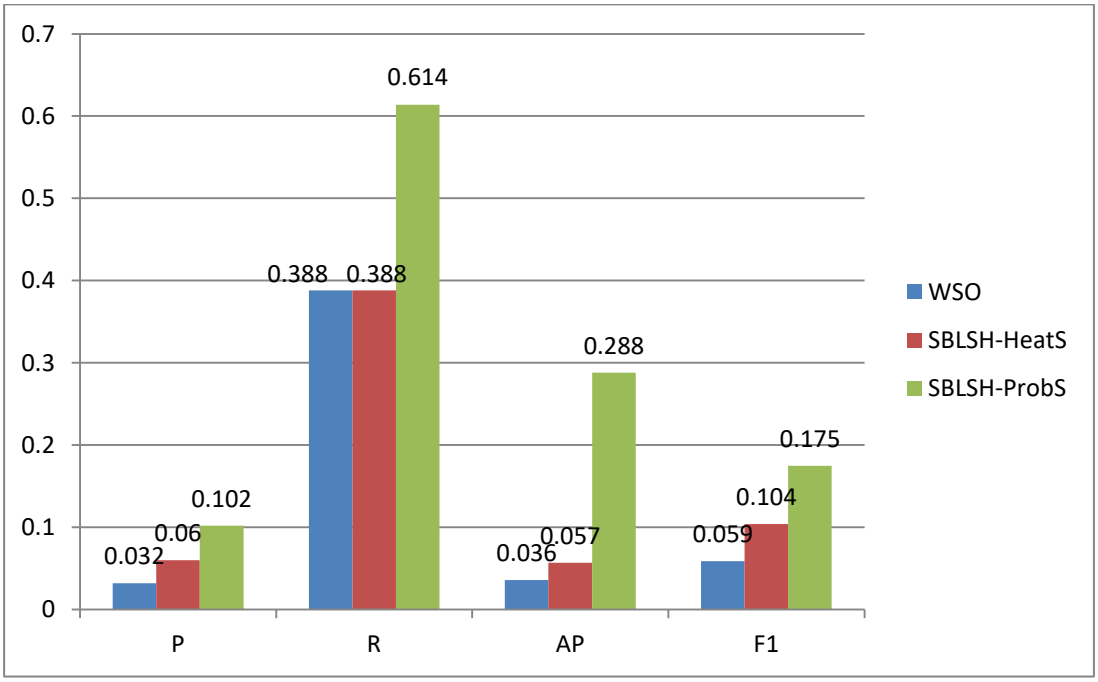
Πίνακας 7: Ποσοστά βελτίωσης αλγορίθμων HeatS και ProbS για 30 προτάσεις ταινιών.

	HeatS	SBLSH-HeatS	%	ProbS	SBLSH-ProbS	%
P(30)	0.0063	0.0125	98.4	0.109	0.151	38
R(30)	0.015	0.0308	105	0.272	0.378	38.9
AP(30)	0.076	0.086	13.1	0.241	0.419	73.8
F1(30)	0.009	0.017	88.8	0.155	0.216	39.3

Είναι προφανές πως και οι δυο αλγόριθμοι παρουσίασαν μεγάλη βελτίωση και σε ακρίβεια αλλά και σε ανάκληση, όπως άλλωστε φαίνεται και στον *Πίνακα 7*. Μάλιστα ο HeatS στην ακρίβεια και στην ανάκληση παρουσίασε διπλασιασμό της τιμής του, με αποτέλεσμα και το διπλασιασμό του σταθμισμένου αρμονικού μέσου. Όσον αφορά τον ProbS η ακρίβεια του βελτιώθηκε κατά 38% της αρχικής, η ανάκληση κατά 38.9% και ο σταθμισμένος αρμονικός μέσος 39.3%. Η μέση ακρίβεια επίσης βελτιώθηκε κατά 13.1% και 73.8% αντίστοιχα σε HeatS και ProbS. Με την ομαδοποίηση των χρηστών σε κάδους και διαχείρισή τους με βάση την ομοιότητά τους όχι μόνο μειώθηκε ο χρόνος εκτέλεσης του προγράμματος αλλά μπορέσαμε να εκμεταλλευτούμε καλύτερα και τις δυνατότητες κάθε αλγορίθμου. Αυτό οφείλεται στο γεγονός ότι η διάχυση των πόρων, όταν πρόκειται να εκτελεστεί σε κοντινούς γείτονες, εκμεταλλεύεται τους πόρους κάθε αντικειμένου καλύτερα εφόσον διατίθεται μεγαλύτερη ποσότητα αυτών για να διαμοιραστεί στα αντικείμενα χωρίς να τους καταναλώνουμε άσκοπα σε συνδέσεις χρηστών-ταινιών που δεν παρουσιάζουν ομοιότητα με το χρήστη στόχο.

4.3 Αξιολόγηση των SBLSH-HeatS και SBLSH-ProbS με τον WSO.

Ένας από τους δημοφιλέστερους αλγορίθμους στα συστήματα προτάσεων είναι ο Weighted Slope One ο οποίος αντιμετωπίζει ευρεία χρήση κυρίως λόγω της ταχύτητας του. Είναι σημαντικό να επισημανθεί ότι ο παρών αλγόριθμος διαχειρίζεται βαθμολογίες σε αντίθεση με τους HeatS και ProbS. Θα εξετάσουμε τα αποτελέσματα του WSO ως προς την ακρίβεια, ανάκληση, μέση ακρίβεια και σταθμισμένο αρμονικό μέσο για την εξαγωγή προτάσεων οι οποίες θα προκύπτουν από το σύνολο των ταινιών που ο αλγόριθμος προέβλεψε βαθμολογία μεγαλύτερη ή ίση του 3. Το μέγεθος της λίστας προτάσεων των SBLSH-HeatS και SBLSH-ProbS θα περιοριστεί ανάλογα με το σύνολο των ταινιών που αρέσουν και θα πρέπει να προβλεφθούν για κάθε χρήστη.

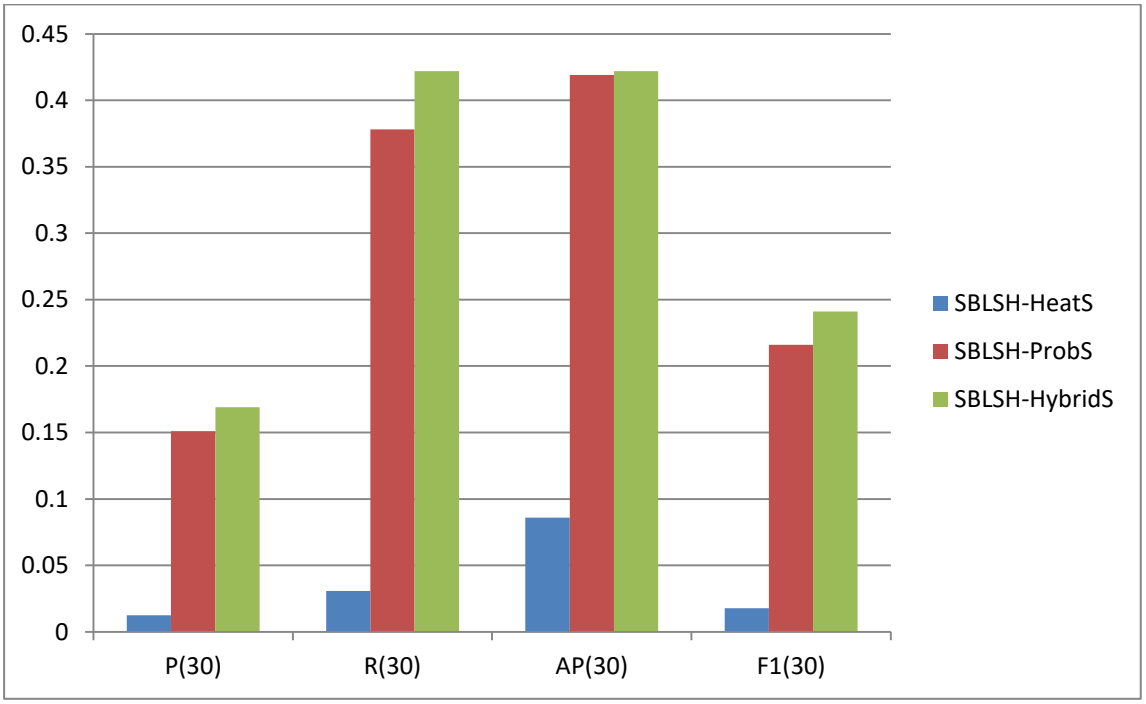


Σχήμα 2: Σύγκριση WSO, SBLSH-HeatS και SBLSH-ProbS.

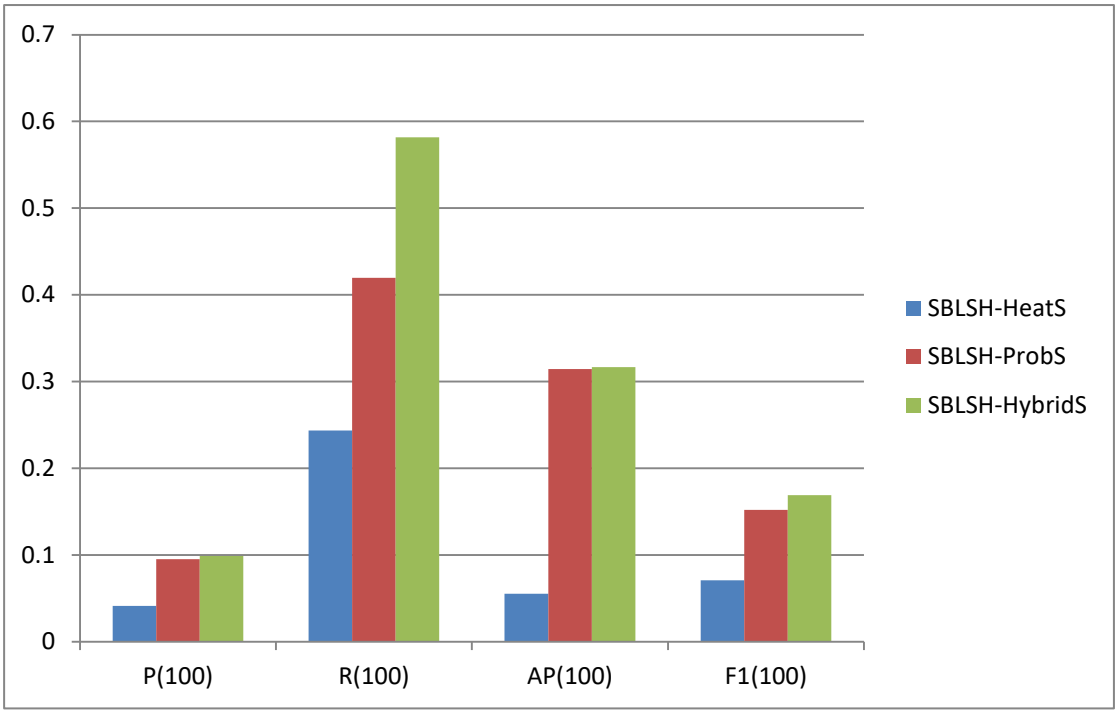
Ο WSO αν και παρουσιάζει μια ανταγωνιστική ανάκληση με εκείνη του SBLSH-HeatS η ακρίβειά του παραμένει αρκετά χαμηλή πράγμα που σημαίνει ότι εύστοχες προτάσεις χάνονται σε πληθώρα από προτάσεις που δεν ενδιαφέρουν το χρήστη. Το ίδιο ισχύει και για τον SBLSH-HeatS ο οποίος και αυτός με τη σειρά του αποτελεί μια αναξιόπιστη λύση για το σύστημά μας. Καλύτερος και από τους τρεις εμφανίζεται ο SBLSH-ProbS ο οποίος και αυτός έχει χαμηλή ακρίβεια αλλά παρουσιάζει αρκετά υψηλή ανάκληση. Επιπλέον, μεγάλη διαφορά παρατηρείται και στο ποσοστό της μέσης ακρίβειας πράγμα που σημαίνει πως οι σωστές προβλέψεις που εξάγονται από τους SBLSH-HeatS και SBLSH-ProbS βρίσκονται σε καλύτερες, υψηλότερες θέσεις στη λίστα προτάσεων συγκριτικά με αυτές του WSO. Τέλος, ο WSO παρόλο που έχει το πλεονέκτημα ότι διαχειρίζεται βαθμολογίες, με αποτέλεσμα να έχει σαν είσοδο δεδομένων ένα λιγότερο αραιό δείγμα συγκριτικά με αυτό των SBLSH-HeatS και SBLSH-ProbS, το οποίο έχει διαμορφωθεί σε τιμές 0 και 1 για βαθμολογίες μεγαλύτερες ή ίσες του 3, δεν πετυχαίνει καλύτερα αποτελέσματα από τους υπόλοιπους και δεν είναι αξιόπιστος για το παρόν σύστημα.

4.4 Βελτίωση των SBLSH-HeatS και SBLSH-ProbS συνδυάζοντάς τους στον SBLSH-HybridS

Ως επέκταση των αλγορίθμων HeatS και ProbS προτάθηκε ο συνδυασμός τους με τη δημιουργία του υβριδικού αλγορίθμου HybridS. Με το συνδυασμό του με τον SBLSH τα αποτελέσματα βελτιώθηκαν αισθητά καθώς επίσης και με την κατάλληλη ρύθμιση της παραμέτρου λ εξασφαλίσαμε τη μέγιστη απόδοση του αλγορίθμου. Μετά από αρκετές δοκιμές η καλύτερη τιμή για την παράμετρο λ είναι ίση με 0.26.



Σχήμα 3: Σύγκριση SBLSH-HeatS, SBLSH-ProbS και SBLSH-HybridS για top-30 προτάσεις.



Σχήμα 4: Σύγκριση SBLSH-HeatS, SBLSH-ProbS και SBLSH-HybridS για top-100 προτάσεις.

Όπως βλέπουμε στο Σχήμα 3 και στο Σχήμα 4 ο αλγόριθμος HybridS παρουσιάζει πολύ καλύτερα αποτελέσματα και από τον HeatS και από τον ProbS τόσο στην ακρίβεια αλλά και στην ανάκληση. Αυτό επιτυγχάνεται λόγω του γεγονότος της ύπαρξης της παραμέτρου λ με την οποία καταφέρνουμε να συνδυάσουμε τις δημοφιλείς προτάσεις του ProbS με αυτές τις λιγότερο δημοφιλείς του HeatS κάνοντας τη λίστα προτάσεων περισσότερο προσωποποιημένη για κάθε χρήστη. Πιο συγκεκριμένα όσον αφορά τη λίστα προτάσεων μήκους 30 και συγκριτικά με τον SBLSH-ProbS που αποφέρει τα αμέσως καλύτερα αποτελέσματα στην ακρίβεια παρουσιάστηκε βελτίωση κατά 11.9%,

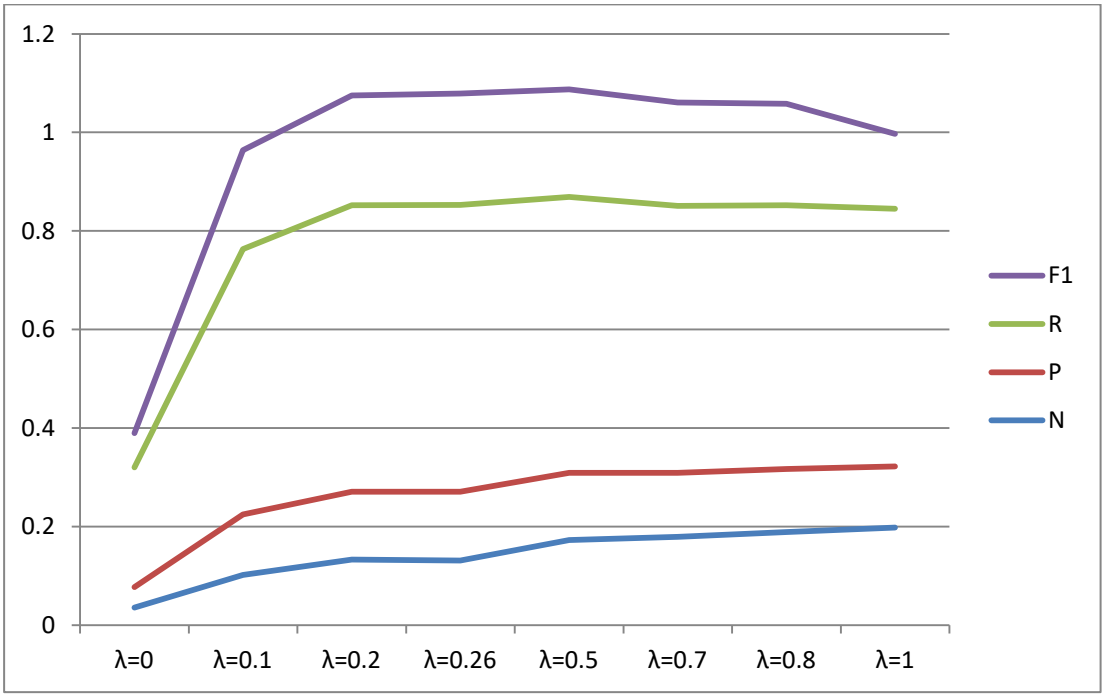
στην ανάκληση κατά 11.6% και στον σταθμισμένο αρμονικό μέσο κατά 11.5%, με τον τελευταίο να αποτελεί βασικό κριτήριο επιλογής του αλγορίθμου εφόσον θέλουμε να μας παρέχει τόσο ακρίβεια όσο και ανάκληση.

Πίνακας 8: Ποικιλομορφία και καινοτομία των SBLSH-HeatS, SBLSH-ProbS, SBLSH-HybridS για top-30 προτάσεις

	D(30)	N(30)
SBLSH-HeatS	0.959	0.009
SBLSH-ProbS	0.907	0.251
SBLSH-HybridS	0.932	0.190

Αρκετό ενδιαφέρον παρουσιάζει η ποικιλομορφία και καινοτομία των προτάσεων που παράγουν οι αλγόριθμοι. Σε πολλά συστήματα προτάσεων η ύπαρξη καινοτομίας και ποικιλίας στις προτάσεις κρίνεται απαραίτητη και πολλές φορές δίνεται περισσότερη σημασία ακόμα και από την ακρίβεια η οποία θεωρείται υπερεκτιμημένη. Εξάλλου σίγουρα κανείς δε θα ήθελε όταν βλέπει προτεινόμενες ταινίες οι προτάσεις να αποτελούνται μόνο από ένα είδος ταινίας. Μεγαλύτερη ποικιλία στις προτάσεις του, όπως άλλωστε ήταν αναμενόμενο εμφανίζει ο SBLSH-HeatS, πράγμα που οφείλεται στο γεγονός της ιδιαιτερότητας του να προωθεί μη δημοφιλή αντικείμενα. Αυτό διακρίνεται και από το εξαιρετικά μικρό ποσοστό στην καινοτομία το οποίο σημαίνει πως τα αντικείμενα που προτείνει δεν έχουν υψηλή δημοτικότητα και δεν έχουν προτιμηθεί από πολλούς χρήστες. Σε αντίθεση ο SBLSH-ProbS ο οποίος είναι ένας αλγόριθμος που στηρίζεται στην προώθηση των δημοφιλέστερων αντικειμένων εμφανίζεται όπως άλλωστε ήταν αναμενόμενο ο χειρότερος από άποψη ποικιλίας και καινοτομίας των προτάσεων του. Ο SBLSH-HybridS για $\lambda = 0.26$ παρουσιάζει και εκείνος μικρότερη ποικιλία και καινοτομία προτάσεων από τον SBLSH-HeatS αλλά όχι τόσο χαμηλή συγκριτικά με τον SBLSH-ProbS λόγω της ύπαρξης στοιχείων του HeatS στον αλγόριθμο.

Όσον αφορά την επιλογή της παραμέτρου λ στον SBLSH-HybridS παρατηρούμε στο *Σχήμα 5* ότι καθώς αυξάνεται η τιμή του πάνω από 0.26 η ακρίβεια, η ανάκληση και κατ' επέκταση και ο σταθμισμένος αρμονικός μέσος ακολουθούν καθοδική πορεία. Είναι φανερό λοιπόν πως για $\lambda = 0.26$ εξασφαλίζουμε τα βέλτιστα αποτελέσματα που μπορεί να μας δώσει ο αλγόριθμος εξισορροπώντας ταυτόχρονα την ανάκληση και την ακρίβεια του. Ενδιαφέρον παρουσιάζει και η διαφοροποίηση των τιμών του όσον αφορά την καινοτομία των προτάσεων αφού καθώς αυξάνεται το λ, όπου η επίδραση του ProbS είναι μεγαλύτερη παρατηρούμε ότι οι προτάσεις που εξάγονται δεν είναι τόσο καινοτόμες συγκριτικά με την περίπτωση όπου το λ μειώνεται και αντίστοιχα η επίδραση του HeatS γίνεται πιο αισθητή.

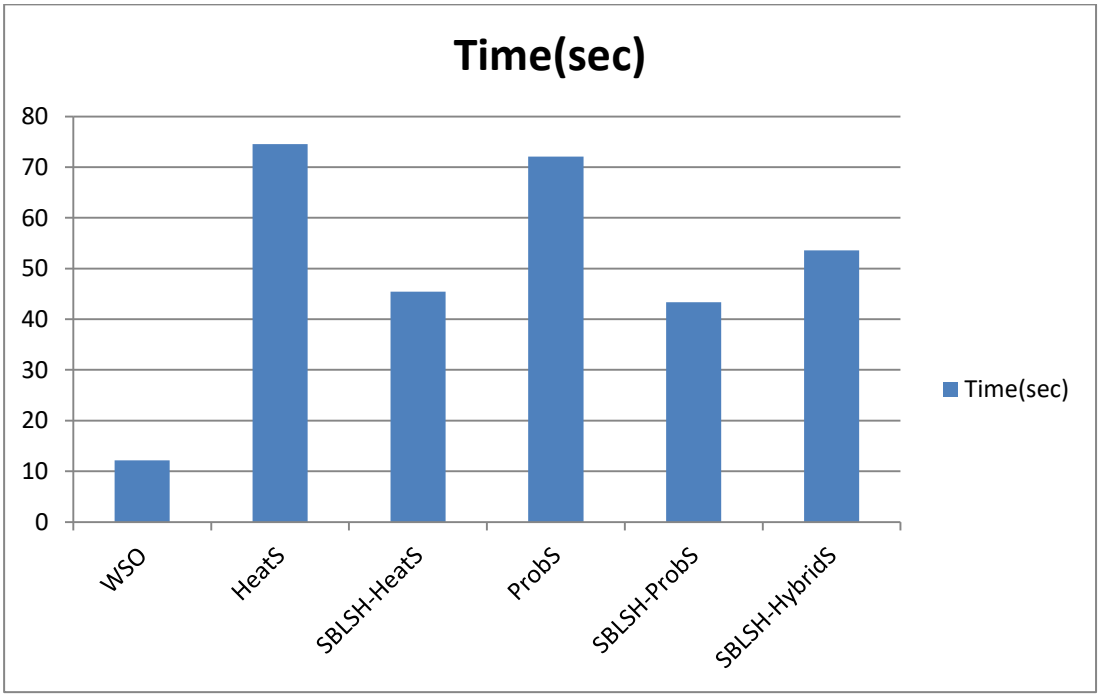


Σχήμα 5: Σύγκριση μετρήσεων για διαφορετικές τιμές του λ στον SBLSH-HybridS.

Τέλος, αξίζει να τονίσουμε για ακόμη μια φορά ότι το δείγμα μας είναι αρκετά αραιό, πράγμα φυσικά που έχει αντίκτυπο και στα αποτελέσματα των αλγορίθμων, αφού για παράδειγμα ο SBLSH-HybridS εκτελώντας τον για τοπ-100 προτάσεις για ένα δείγμα μπορεί να προέβλεψε 7850 προτάσεις σωστές από τις 15474, ωστόσο οι 85579 από τις 94300 που παρήγαγε αποτελούν προτάσεις για τις οποίες δε γνωρίζουμε από το test set αν άρεσαν στο χρήστη ή όχι και για αυτό το λόγο αντιμετωπίζονται αναγκαστικά ως λανθασμένες. Επιπλέον, συμπεραίνουμε ότι ο SBLSH-HybridS αποτελεί σίγουρα καλύτερη λύση και από τον HeatS αλλά και από τον ProbS. Λόγω της παραμέτρου λ ο HybridS εκμεταλλεύεται τα πλεονεκτήματα και των δυο αλγορίθμων εξίσου. Επιτυγχάνει το συνδυασμό εξαγωγής μια λίστας προτάσεων που χαρακτηρίζεται τόσο από ακρίβεια, λόγω της συμβολής του ProbS, αλλά και από ποικιλομορφία των προτάσεων, λόγω της επίδρασης του HeatS, καθώς δίνεται η δυνατότητα με κατάλληλη ρύθμιση της παραμέτρου λ να γίνει ενίσχυση σε οποιοδήποτε από αυτά τα δυο χαρακτηριστικά.

4.5 Αξιολόγηση χρόνων εκτέλεσης

Μέχρι στιγμής έχουμε εξετάσει κατά πόσο οι αλγόριθμοι είναι αποτελεσματικοί και εξάγουν σωστά αποτελέσματα. Είναι σημαντικό βέβαια τα αποτελέσματα αυτά να δίνονται έγκαιρα σε όσο το δυνατόν μικρότερο χρονικό διάστημα γίνεται. Βέβαια η ύπαρξη μεγάλου όγκου δεδομένων και η διαρκής αύξησή τους αποτελεί σημαντικό εμπόδιο στην επίτευξη χαμηλών χρόνων εκτέλεσης. Για αυτό το λόγο ο αλγόριθμος WSO χρησιμοποιείται πολύ συχνά παρά του γεγονότος ότι δεν είναι πολύ αποτελεσματικός.



Σχήμα 6: Χρόνοι εκτέλεσης αλγορίθμων WSO, SBLSH-HeatS, SBLSH-ProbS, SBLSH-HybridS.

Στο *Σχήμα 6* βλέπουμε τους χρόνους όλων των αλγορίθμων που εξετάσαμε. Ο WSO εμφανίζεται με διαφορά ο ταχύτερος, έχοντας το πλεονέκτημα της δημιουργίας του πίνακα αποκλίσεων για όλους τους χρήστες σε σύγκριση με τους HeatS και ProbS όπου ο πίνακας ανακατανομής των πόρων δημιουργείται κάθε φορά για κάθε χρήστη ξεχωριστά. Επιπλέον, είναι αισθητή η βελτίωση του χρόνου των HeatS και ProbS μετά τη χρήση του SBLSH ο οποίος μειώνεται σχεδόν κατά το ήμισυ. Αν ορίσουμε ως $\langle \kappa_u \rangle$ το μέσο βαθμό των χρηστών, δηλαδή το μέσο πλήθος αντικειμένων που οι χρήστες έχουν προτιμήσει, τότε η πολυπλοκότητα όμοια στους SBLSH-HeatS, SBLSH-ProbS και SBLSH-HybridS ισούται με $O(m\langle k_u^2 \rangle + mn\langle k_u \rangle)$ όπου m το πλήθος των χρηστών και n των αντικειμένων, όπου ο πρώτος όρος αναφέρεται στη δημιουργία του μητρώου ανακατανομής και ο δεύτερος στην τελική κατανομή των πόρων. Σαφώς, $\langle k_u^2 \rangle < n\langle k_u \rangle$, με αποτέλεσμα η πολυπλοκότητα των αλγορίθμων να είναι ίση με $O(mn\langle k_u \rangle)$. [7] Ο SBLSH-HybridS εμφανίζεται ελάχιστα πιο αργός, λόγω των επιπλέον υπολογισμών που πραγματοποιεί με την παράμετρο λ, πράγμα που αντισταθμίζεται βέβαια από την ακρίβεια των αποτελεσμάτων του.

5. ΣΥΜΠΕΡΑΣΜΑΤΑ

Τα συστήματα προτάσεων αποτελούν μια πολλά υποσχόμενη τεχνολογία ως προς την αντιμετώπιση της άφθονης πληροφορίας που καλούμαστε να διαχειριστούμε. Πρόσφατα προτάθηκε μια υβριδική μέθοδος (HybridS) η οποία συνδυάζει τη Θερμική εξάπλωση (Heat Spreading, HeatS) και την Πιθανοτική εξάπλωση (Probability Spreading, ProbS).

Στην παρούσα πτυχιακή έγινε μελέτη των παραπάνω αλγορίθμων καθώς επίσης προτάθηκε για τη βελτίωσή τους η χρήση Local Sensitive Hashing (LSH). Προκειμένου να εξεταστεί και να κατανοηθεί η συμπεριφορά των αλγορίθμων αξιολογήθηκαν βάσει έξι μετρικών. Ο WSO αν και ταχύτερος και με ανταγωνιστική ανάκληση δεν παρουσίασε αρκετά ικανοποιητικές επιδόσεις. Ο ProbS έχει την τάση να συνιστά δημοφιλή αντικείμενα με αποτέλεσμα η σύσταση της λίστας των προτάσεων του να είναι πιο ακριβής. Ο HeatS αντίθετα τείνει να προτείνει πιο εξειδικευμένα αντικείμενα από την άποψη ότι είναι λιγότερο δημοφιλή και δεν έχουν προτιμηθεί από πολλούς χρήστες, παρουσιάζοντας ωστόσο μεγαλύτερη ποικιλία και διαφορετικότητα στις προτάσεις του. Σε αυτή την πτυχιακή παρουσιάστηκε πως οι δυο τελευταίοι αυτοί αλγόριθμοι βελτιώνονται αν συνδυαστούν με τον αλγόριθμο LSH, αλλά και πως ο HybridS με την κατάλληλη ρύθμιση των παραμέτρων του μπορεί να επιτύχει αρκετά υψηλές επιδόσεις τόσο σε ακρίβεια και ανάκληση αλλά και στη διαφορετικότητα των προτάσεών του. Έτσι λοιπόν ο HybridS παίρνοντας τη μορφή ενός γραμμικού συνδυασμού των ProbS και HeatS, καθώς και την επίδραση του καθενός, θα μπορούσε να ρυθμιστεί ανάλογα με τις προτιμήσεις κάθε χρήστη στο κατά πόσο θα τον ενδιέφερε να δει κάτι διαφορετικό από τις αναμενόμενες προτάσεις, παρέχοντας μια λίστα προτάσεων περισσότερο προσωποποιημένη για κάθε χρήστη.

Τέλος, με το συνδυασμό του αλγορίθμου HybridS με εκείνον του SBLSH η βελτίωση των προτάσεων σε σύγκριση με εκείνων των υπόλοιπων αλγορίθμων είναι αισθητή, με τον HybridS να αποτελεί τον καλύτερο μέχρι στιγμής αλγόριθμο για συστήματα προτάσεων βασισμένα σε δίκτυα, συνδυάζοντας ταυτόχρονα ακρίβεια και ποικιλομορφία. Η πολυπλοκότητα των συστημάτων προτάσεων που παρουσιάστηκε σε αυτή την πτυχιακή επισημαίνει τη δυνατότητα για μελλοντικές περαιτέρω βελτιώσεις στο σχεδιασμό αλγορίθμων.

ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ

Ξενόγλωσσος όρος	**Ελληνικός Όρος**
Heat Spreading	Θερμική διάχυση
Probability Spreading	Πιθανοτική διάχυση
Recommendations	Προτάσεις
Hybrid	Υβριδικός
Local Sensitive Hashing	Κατακερματισμός τοπικής ευαισθησίας
Precision	Ακρίβεια
Recall	Ανάκληση
F1-Measure	Σταθμισμένος αρμονικός μέσος
Diversity	Ποικιλομορφία
Novelty	Καινοτομία
Collaborative filtering	Συνεργατικό φιλτράρισμα
Content-based filtering	Φιλτράρισμα βασισμένο στο περιεχόμενο
Weighted slope one	Μέθοδος Σταθμισμένης Μοναδιαίας Κλίσης
Cosine similarity	Συνημιτονοειδής ομοιότητα
Sparsity	Αραιότητα
Overfitting	Υπερπροσαρμογή
Hyperplane	Υπερεπίπεδο

ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ

WSO	Weighted Slope One
HeatS	Heat Spreading
ProbS	Probability Spreading
HybridS	Hybrid Spreading
SBLSH	SuperBit-Local Sensitive Hashing
P	Precision
R	Recall
F1	F1-Measure
D	Diversity
N	Novelty

ΑΝΑΦΟΡΕΣ

[1] L. L¨u, C. H. Yueng, M. Medo, Y.-C. Zhang, Z.-K. Zhang and T. Zhou, “Recommender Systems”, Physics Reports Vol. 519, doi: 10.1016/j.physrep.2012.02.006, pp. 1-49, February 2012.

[2] F. Maxwell Harper and Joseph A. Konstan, “The MovieLens Datasets: History and Context”, ACM Transactions on Interactive Intelligent Systems (TiiS) Vol. 5 Issue 4, doi: 10.1145/2827872, Article No. 19, January 2016.

[3] Fuguo Zhang, “Robust Analysis of Network based Recommendation Algorithms against Shilling Attacks”, International Journal of Security and its Applications Vol.9 No.3, doi: 10.14257/ijseia.2015.9.3.03, pp. 13-24, March 2015

[4] Jinhu Liu, Chengcheng Yang, Zi-Ke Zhang, “A two-step Recommendation Algorithm via Iterative Local Least Squares”, Computing Research Repository, doi: CoRR abs/1206.3320, June 2012.

[5] Jianqiu Ji, Jianmin Li, Shuicheng Yan, Bo Zhang, Qi Tian, “Super-Bit Locality-Sensitive Hashing”, NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems, pp. 108-116, December 2012.

[6] T. Zhou, Z. Kuscsik, J.-G. Liu, M. Medo, J.R. Wakeling, Y.-C. Zhang, “Solving the apparent diversity–accuracy dilemma of recommender systems”, National Academy of Sciences of the United States of America 107, doi: 10.1073/pnas.1000488107, pp. 4511–4515, February 2010.

[7] Tao Zhou, Jie Ren, Mat´uˇs Medo and Yi-Cheng Zhang, “How to project a bipartite network?”, Physical Review E 76, 046115 (2007), doi: 10.1103/PhysRevE.76.046115, February 2013.

[8] Charikar, Moses, “Similarity Estimation Techniques from Rounding Algorithms”, 34th Annual ACM Symposium on Theory of Computing, doi: 10.1145/509907.509965, pp. 380-388, May 2002.

[9] G. Xue, Y. Jiang, Y. You and M. Li., “A topology-aware hierarchical structured overlay network based on locality sensitive hashing scheme”, Proceedings of the second workshop on Use of P2P, GRID and agents for the development of content networks, doi: 10.1145/1272980.1272985, pp. 3-8, June 2007.