

ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΜΑΤΙΚΗΣ
Π.Μ.Σ.: «ΠΛΗΡΟΦΟΡΙΚΗ & ΤΗΛΕΜΑΤΙΚΗ»
Κατεύθυνση: Υπολογιστικές και Διαδικτυακές Τεχνολογίες και Εφαρμογές

Διπλωματική Εργασία

ΕΞΑΤΟΜΙΚΕΥΣΗ ΠΕΡΙΕΧΟΜΕΝΟΥ ΜΕ ΑΞΙΟΠΟΙΗΣΗ ΓΝΩΜΗΣ ΚΑΙ ΚΟΙΝΩΝΙΚΟΥ ΔΙΚΤΥΟΥ

Επιβλέπων Καθηγητής
Βαρλάμης Ηρακλής, Επίκουρος Καθηγητής

Μέλη Τριμελούς Επιτροπής
Νικολαΐδου Μάρα, Καθηγήτρια
Μιχαήλ Δημήτριος, Λέκτορας

Σαρδιανός Χρήστος
Α.Μ.: 12112

Αθήνα, Φεβρουάριος 2014

"Μία είναι η απόδειξη του μπορώ,
Το κάνω."

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα της διπλωματικής μου εργασίας και Επίκουρο Καθηγητή του Τμήματος Πληροφορικής & Τηλεματικής του Χαροκοπείου Πανεπιστημίου κ.Βαρλάμη Ηρακλή για την εξαιρετική συνεργασία, την πολύτιμη καθοδήγηση και γνώση, καθώς και για την αμέριστη υποστήριξη που έδειξε καθ' όλη την διάρκεια εκπόνησης της παρούσας μελέτης. Επίσης, θα ήθελα να τον ευχαριστήσω για την εμπιστοσύνη που δείχνει προς το πρόσωπό μου, δίνοντάς μου την ευκαιρία να ασχοληθώ ερευνητικά με τα επιστημονικά μου ενδιαφέροντα.

Θα ήθελα επίσης να ευχαριστήσω την οικογένειά μου και όλους όσους ήταν εμφανείς και αφανείς συμπαραστάτες σε αυτό το βήμα μου, για την κατανόηση, την υπομονή και την στήριξή τους μέχρι να φτάσω στην ολοκλήρωση αυτού του στόχου.

Πίνακας Περιεχομένων

1 Εισαγωγή 10
1.1 Σκοπός 10
2 Θεωρητικό Υπόβαθρο 12
2.1 Βιβλιογραφική Επισκόπηση 12
2.2 Αλγόριθμοι Collaborative Filtering 19
2.2.1 User-User 19
2.2.2 FunkSVD 21
2.2.3 Item-Item 23
2.3 Συναφείς Εργασίες 27
2.4 Εργαλεία 29
3 Προτεινόμενη Προσέγγιση 32
3.1 Περιγραφή Συστήματος 34
3.2 Κλάσεις 38
3.3 Μεθοδολογία 40
3.4 Μετρικές Αξιολόγησης 41
3.4.1 MAE (Mean Absolute Error) 41
3.4.2 RMSE (Root Mean Squared Error) 42
3.4.3 nDCG (Normalized Discounted Cumulative Gain) 42
3.4.4 Top-N nDCG (Normalized Discounted Cumulative Gain for Top-N items) 43
4 Υλοποίηση & Αξιολόγηση του μοντέλου 44
4.1 Σύνολα Δεδομένων & Πλατφόρμα Πειραμάτων 44
4.2 Υλοποίηση 46
4.3 Αξιολόγηση της Πειραματικής Διαδικασίας 59
4.3.1 Εκτέλεση Collaborative Filtering στον πλήρη γράφο 60
4.3.2 Εκτέλεση Collaborative Filtering στους πλήρεις υπογράφους 60
4.3.3 Εκτέλεση Collaborative Filtering στους υπογράφους με K-nn γείτονες 61
4.4 Αποτελέσματα 62
4.4.1 Αποτελέσματα για διαφορετικό πλήθος υπογράφων – s 62
4.4.2 Συγκριτικά Αποτελέσματα 76
4.4.3 Αποτελέσματα για διαφορετικό αριθμό πλησιέστερων γειτόνων – knn 78
5 Συμπεράσματα 84
6 Μελλοντικές Εργασίες 85
7 Παραρτήματα 86
7.1 Παράρτημα I (Πίνακες Αποτελεσμάτων Εκτέλεσης των δύο datasets) 86
7.2 Παράρτημα II (Διάγραμμα Επικοινωνίας των Components) 88
8 Βιβλιογραφία 89

Κατάλογος Πινάκων – Εικόνων

Πίνακας 1: Πίνακας αξιολογήσεων (utility matrix) 13
Πίνακας 2: Χαρακτηριστικά των συνόλων δεδομένων 45
Πίνακας 3: Dominance Matrices for all three algorithms in Epinions Dataset 76
Πίνακας 4: Dominance Matrices for all three algorithms in Flixster Dataset 77

Εικόνα 1: Μεταβατικότητα της εμπιστοσύνης (πηγή: Varlamis et al., 2013) 18
Εικόνα 2: Gradient Descent κατανομή (πηγή: Principe et al., 1999) 23
Εικόνα 3: Διαδικασία υπολογισμού ομοιότητας αντικειμένων (πηγή: Sarwar et al., 2001) .. 24
Εικόνα 4: Αφαιρετικό διάγραμμα λειτουργίας του μοντέλου 33
Εικόνα 5: Περιγραφή της λειτουργίας του μοντέλου 34
Εικόνα 6: Παράδειγμα partitioning του αρχικού κοινωνικού γράφου και των items που έχουν βαθμολογήσει οι χρήστες 36
Εικόνα 7: Class Diagrams των βασικών κλάσεων του μοντέλου 39
Εικόνα 8: Δομή φακέλων του project 48
Εικόνα 9: Περιεχόμενα properties στο pom.xml 48
Εικόνα 10: Λίστα εξαρτήσεων στο pom.xml 50
Εικόνα 11: Compiler plugin στο pom.xml 50
Εικόνα 12: Evaluation plugin στο pom.xml 51
Εικόνα 13: Source Code Packages 52
Εικόνα 14: Χρόνος εκτέλεσης της αξιολόγησης κάθε αλγόριθμου για τους διαφορετικούς υπογράφους (Epinions) 62
Εικόνα 15: Απόδοση κατά nDCG όλων των αλγορίθμων στο Epinions για τους διαφορετικούς υπογράφους (s) 64
Εικόνα 16: Απόδοση κατά Top-N nDCG όλων των αλγορίθμων στο Epinions για τους διαφορετικούς υπογράφους (s) 65
Εικόνα 17: Απόδοση κατά MAE όλων των αλγορίθμων στο Epinions για τους διαφορετικούς υπογράφους (s) 66
Εικόνα 18: Απόδοση κατά RMSE όλων των αλγορίθμων στο Epinions για τους διαφορετικούς υπογράφους (s) 67
Εικόνα 19: Σύγκριση του μέσου πλήθους μοναδικών χρηστών ανά υπογράφο για τα δύο σύνολα δεδομένων 68
Εικόνα 20: Σύγκριση του μέσου πλήθους μοναδικών αντικειμένων ανά υπογράφο για τα δύο σύνολα δεδομένων 69

Εικόνα 21: Σύγκριση του μέσου πλήθους βαθμολογιών ανά αντικείμενο (in-Degree), ανά υπογράφο για τα δύο σύνολα δεδομένων 71

Εικόνα 22: Σύγκριση του μέσου σφάλματος των αλγορίθμων User-User_ByRating & SVD ως προς τον μέσο αριθμό χρηστών σε κάθε υπογράφο για τα δύο datasets 71

Εικόνα 23: Σύγκριση του κανονικοποιημένου κέρδους των αλγορίθμων User-User_ByRating & SVD ως προς τον μέσο αριθμό χρηστών σε κάθε υπογράφο για τα δύο datasets 72

Εικόνα 24: Απόδοση κατά nDCG όλων των αλγορίθμων στο Flixster για τους διαφορετικούς υπογράφους (s) 74

Εικόνα 25: Απόδοση κατά Top-N nDCG όλων των αλγορίθμων στο Flixster για τους διαφορετικούς υπογράφους (s) 74

Εικόνα 26: Απόδοση κατά MAE όλων των αλγορίθμων στο Flixster για τους διαφορετικούς υπογράφους (s) 75

Εικόνα 27: Απόδοση κατά Top-N nDCG όλων των αλγορίθμων στο Flixster για τους διαφορετικούς υπογράφους (s) 76

Εικόνα 28: Χρόνος εκτέλεσης της αξιολόγησης για διαφορετικό Neighborhood Size και s={4, 65,1000} στο Epinions 79

Εικόνα 29: Απόδοση κατά RMSE για διαφορετικό Neighborhood Size (k) 79

Εικόνα 30: Απόδοση κατά MAE για διαφορετικό Neighborhood Size (k) 81

Εικόνα 31: Απόδοση κατά nDCG για διαφορετικό Neighborhood Size (k) 82

Εικόνα 32: Απόδοση κατά Top-N nDCG για διαφορετικό Neighborhood Size (k) 83

Περίληψη

Είναι γεγονός πως λόγω της ραγδαίας ανάπτυξης των κοινωνικών δικτύων και της αύξησης του περιεχομένου κυρίως λόγω των Web 2.0 εφαρμογών, κύριο χαρακτηριστικό πλέον αποτελεί η βελτίωση της εμπειρίας των χρηστών (user experience) στις διάφορες σελίδες ενημέρωσης, ηλεκτρονικών καταστημάτων και κοινωνικών δικτύων.

Συνεπώς, αυξάνεται διαρκώς και η έρευνα γύρω από τους τομείς που ασχολούνται με την αλληλεπίδραση των χρηστών με τις ίδιες τις εφαρμογές. Τα Συστήματα Συστάσεων είναι ένας από τους βασικότερους τομείς έρευνας που ασχολείται με την βελτίωση της εμπειρίας του χρήστη μέσα από την παραγωγή εξατομικευμένων συστάσεων, είτε για πιθανά ενδιαφέροντα προϊόντα αγοράς, είτε για ενδιαφέροντα άρθρα, σελίδες κτλ. Μία από τις γνωστότερες τεχνικές στα συστήματα συστάσεων είναι η τεχνική του Συνεργατικού Φιλτραρίσματος (Collaborative Filtering – CF). Στόχος της τεχνικής αυτής είναι η αναζήτηση σχέσεων μεταξύ των χρηστών, από τις οποίες μπορεί να προκύψει κάποιο συμπέρασμα για πιθανά κοινά χαρακτηριστικά, ενδιαφέροντα, προτιμήσεις κλπ και κατά συνέπεια η πρόταση αντικειμένων που εκτιμάται ότι ενδιαφέρουν τον χρήστη.

Σε πολλές όμως περιπτώσεις, σε έναν ιστότοπο υπάρχει συνδυασμός πληροφορίας περιεχομένου και κοινωνικής αλληλεπίδρασης. Στόχος λοιπόν, της παρούσας διπλωματικής μελέτης είναι να εξερευνήσει την δυνατότητα παραγωγής εξατομικευμένων συστάσεων για τον χρήστη, αξιοποιώντας πληροφορία κοινωνικού περιεχομένου που συνοδεύει τους χρήστες σε ένα κοινωνικό δίκτυο. Για τον σκοπό αυτό προτείνεται ένα μοντέλο παραγωγής συστάσεων το οποίο λαμβάνει υπόψη του την συνδυαζόμενη πληροφορία και το οποίο σύμφωνα με τα πειράματα που εκτελέστηκαν μπορεί να παράγει συστάσεις με αποδοτικότερο τρόπο. Χαρακτηριστικό του μοντέλου είναι η εφαρμογή των αλγορίθμων Collaborative Filtering σε υπογράφους σε αντίθεση με την εκτέλεσή του σε ολόκληρο τον γράφο, όπως στις παραδοσιακές τεχνικές. Οι υπογράφοι δημιουργούνται με τη διαίρεση του αρχικού γράφου με βάση τη δομή του κοινωνικού δικτύου. Επιπλέον, μέσω του μοντέλου που περιγράφεται, προτείνεται

μία αξιόπιστη λύση στο πρόβλημα δημιουργίας συστάσεων για νέους χρήστες, χωρίς προηγούμενο ιστορικό, που είναι γνωστό ως Cold-Start problem.

Λέξεις-κλειδιά: Συστάσεις, Συστήματα Συστάσεων, Κοινωνικά Δίκτυα, Εξατομίκευση Περιεχομένου, Συσταδοποίηση Γράφου, Συνεργατικό Φιλτράρισμα

Abstract

The increased popularity of social networking applications and the explosion in user provided content which is mainly due to Web 2.0 applications has increased the interest of researchers and companies to the analysis of social networks and the generation of a personalized user experience. One interesting area of research in this direction are Recommender Systems, which aim to predict the interests of the social networking application users and to recommend them items that may be of interest to them. Key goal of this field is to improve user-experience in these applications and subsequently to create profit, for example for commercial sites by increasing the number of items bought, or for advertisement funded sites (e.g. blogs and news sites) by increasing the users' loyalty and the daily traffic.

In this environment, there is a large spectrum of applications, such as customer review sites, in which users can express their opinions about something (e.g. a product, a film, etc.) by providing a rating or writing a full-text review. What has recently changed in this is an overlay of social interaction between the users, which takes the form of mutual friendship, trust or distrust, and can be exploited to provide better recommendations.

Taking the above mentioned into account, the research question that arises is if there can be a way to combine the users opinions and their social interaction information, so that the resulting recommendation can be more precise and personalized. This thesis, proposes a model that takes into account both the social networking graph of users and their explicit opinions about items or products and generates personalized recommendations by applying Collaborative Filtering (CF) techniques on subgraphs of the original graph. The partitioning of the original graph is based on the social information.

For the purpose of our study, we tested a large number of scenarios and performed the corresponding experiments in two popular datasets (Epinions and Flixster) that provide both social and rating information. The results show that partitioning the social graph and performing Collaborative Filtering over

subpartitions can perform faster and improve the performance of the existing CF approaches. In addition to the innovation of this research, we propose a solution to the Cold-Start problem that concerns the generation of recommendations for new users.

Keywords: Recommendations, Recommender Systems, Social Networks, Content Personalization, Graph Partitioning, Collaborative Filtering

1 Εισαγωγή

1.1 Σκοπός

Η ανάπτυξη του Web 2.0 και των εφαρμογών του δημιούργησε ένα ευρύ περιβάλλον έρευνας και ανάλυσης. Από τα σημαντικότερα χαρακτηριστικά του Web 2.0 ήταν η δημιουργία των κοινωνικών δικτύων και των παραλλαγών τους, που έχουν πλέον κυριαρχήσει στο διαδίκτυο. Σημαντικό λοιπόν μερίδιο στην καλύτερη αξιοποίηση αυτών των τεχνολογιών αποτελεί η ανάλυση των κοινωνικών δικτύων αλλά και των υπόλοιπων όμοιων δικτύων (blogs, εφαρμογές κοινωνικής δικτύωσης και σελίδων αξιολόγησης προϊόντων). Μια από τις κυριότερες κατηγορίες ανάλυσης τέτοιων δικτύων είναι οι Web εφαρμογές που έχουν σκοπό την πρόβλεψη των πιθανών επιλογών του χρήστη, γνωστές και ως Συστήματα Συστάσεων (Σ.Σ. - Recommendation Systems). Δύο από τις πιο χαρακτηριστικές περιπτώσεις όπου βρίσκουν εφαρμογή τα Συστήματα Συστάσεων είναι, για παράδειγμα, η πρόταση στους αναγνώστες ηλεκτρονικών εφημερίδων νέων άρθρων τα οποία θα βρίσκουν ενδιαφέροντα ή η πρόταση στους επισκέπτες ηλεκτρονικών καταστημάτων προϊόντων τα οποία ταιριάζουν περισσότερο στα ενδιαφέροντα και τις προτιμήσεις τους και τα οποία κατά συνέπεια είναι περισσότερο πιθανό να αγοράσουν. Έτσι, *"ένα Σύστημα Συστάσεων είναι ένας μηχανισμός – υποκατηγορία των συστημάτων «φιλτραρίσματος» πληροφοριών – με σκοπό την παραγωγή συστάσεων με χρήσιμο περιεχόμενο για τους χρήστες (π.χ. προϊόντα, άρθρα κλπ)" (Anand et al., 2012).*

Η σύγχρονη όμως τάση στις διάφορες σελίδες αγορών, είτε αυτά είναι τα ίδια τα ηλεκτρονικά καταστήματα μεγάλων αλυσίδων, είτε είναι απλά portal ενημέρωσης των καταναλωτών για τα χαρακτηριστικά ενός προϊόντος, είναι ο χρήστης να έχει τη δυνατότητα να εκφράσει τη γνώμη του για κάθε προϊόν, βαθμολογώντας ξεχωριστά τα διάφορα χαρακτηριστικά του. Για παράδειγμα, στο δημοφιλή ιστότοπο ταξιδιών TripAdvisor ο επισκέπτης βαθμολογεί το ξενοδοχείο χωριστά για κάθε υπηρεσία που προσφέρει (reception, καθαριότητα, φαγητό κλπ). Ταυτόχρονα όμως, στις ιστοσελίδες αυτές υπάρχει και πληροφορία κοινωνικού δικτύου για τους χρήστες η οποία μπορεί να αξιοποιηθεί. Μπορεί για παράδειγμα ο χρήστης να χαρακτηρίσει μια κριτική χρήσιμη ή μη, ή μπορεί να εκφράσει την εμπιστοσύνη του ή μη, προς κάποιον άλλο χρήστη.

Στόχος της παρούσας διπλωματικής μελέτης είναι η σχεδίαση ενός μοντέλου που θα λαμβάνει υπόψη του τόσο τις γνώμες που καταθέτουν οι χρήστες για τις διαφορετικές ιδιότητες ενός αντικειμένου (aspect based opinion mining) αλλά και το κοινωνικό δίκτυο των χρηστών ώστε να καταλήγει στις καλύτερες εναλλακτικές προτάσεις.

2 Θεωρητικό Υπόβαθρο

2.1 Βιβλιογραφική Επισκόπηση

Στη διεθνή βιβλιογραφία, τα Συστήματα Συστάσεων διακρίνονται στις ακόλουθες κατηγορίες με κριτήριο τον τρόπο λειτουργίας και παραγωγής των συστάσεών τους (Melville and Sindhwani, 2010).

- Τα **Content-based** συστήματα, τα οποία βασίζονται κυρίως στα χαρακτηριστικά του περιεχομένου για το οποίο παράγουν συστάσεις. Για παράδειγμα, για ένα content-based σύστημα συστάσεων ταινιών σημαντικό ρόλο στην παραγωγή προτάσεων ταινιών για τον χρήστη μπορεί να διαδραματίζουν τα είδη ταινιών που έχουν καταχωρηθεί καθώς και τα είδη ταινιών που ο εν λόγω χρήστης έχει δηλώσει ότι τον ενδιαφέρουν.
- Τα **Collaborative filtering** (**CF**) συστήματα, τα οποία παράγουν συστάσεις βασιζόμενα σε κάποιο μέτρο ομοιότητας ανάμεσα στους χρήστες και τα υπό έλεγχο προϊόντα. Με αυτόν τον τρόπο, ουσιαστικά, τα προϊόντα που προτείνονται σε κάποιο χρήστη έχουν να κάνουν με τα προϊόντα που επιλέχθηκαν από χρήστες με όμοια χαρακτηριστικά με τον προς εξέταση χρήστη.
- Τα **Hybrid** συστήματα, τα οποία συνδυάζουν χαρακτηριστικά των δύο παραπάνω κατηγοριών, εκμεταλλευόμενα τις δυνατότητες τους για να πετύχουν καλύτερα αποτελέσματα προβλέψεων.

Σχετικά με τη γενική δομή των Συστημάτων Συστάσεων διακρίνονται δύο είδη οντοτήτων: (i) τους ***Χρήστες*** και τα (ii) ***Αντικείμενα***. Οι μεν χρήστες είναι εκείνοι που, με κάποιο έμμεσο ή άμεσο τρόπο, αξιολογούν κάποια από τα διαθέσιμα αντικείμενα, με αποτέλεσμα να εκφράζουν κάποιου είδους προτίμηση για κάποια αντικείμενα και τα χαρακτηριστικά τους. Τα δε αντικείμενα είναι ουσιαστικά το αντικείμενο (ειδήσεις, προϊόντα, άλλοι χρήστες) για το οποίο το σύστημα συστάσεων προσπαθεί να προβλέψει την προτίμηση του χρήστη και κατά συνέπεια να του το προτείνει καθώς εκτιμά ότι ταιριάζει περισσότερο στο προφίλ του.

Η ιδιαιτερότητα της εφαρμογής των συστημάτων συστάσεων και ο λόγος που κάνει απαραίτητη την υλοποίησή τους σε όλο και περισσότερους ιστότοπους και ηλεκτρονικές επιχειρήσεις, ανάμεσα στους οποίους συγκαταλέγονται μερικές από τις μεγαλύτερες εταιρίες πληροφορικής και παροχής υπηρεσιών και/ή προϊόντων, είναι η ανάγκη για παροχή εξατομικευμένου περιεχομένου ή πληροφορίας στον χρήστη-επισκέπτη. Το αποτέλεσμα θα κάνει την περιήγηση του χρήστη πιο άνετη και ευχάριστη και αφετέρου θα συμβάλλει στη συχνότερη επίσκεψη του χρήστη στον συγκεκριμένο ιστότοπο – ηλεκτρονικό κατάστημα.

Βασικό κομμάτι των συστημάτων συστάσεων είναι ο «πίνακας αξιολόγησης» (utility matrix) στον οποίο απεικονίζονται ζευγάρια χρήστη-αντικειμένου όπως στον παρακάτω πίνακα:

		Items					
		1	*2*	...	*i*	...	*m*
Users	*1*	5	3		1	2	
	2		2				4
	:			5			
	u	3	4		2	1	
	:					4	
	n			3	2		
	a	3	5		?	1	

Πίνακας 1: Πίνακας αξιολογήσεων (utility matrix)

Έτσι, για κάθε έναν από τους n χρήστες και m αντικείμενα, το κελί $r_{u,i}$ εκφράζει την αξιολόγηση που έχει δώσει ο χρήστης u για το αντικείμενο i. Θεωρητικά ο πίνακας αυτός είναι συνήθως αραιός καθώς κάθε χρήστης έχει αξιολογήσει ένα μικρό τμήμα των διαθέσιμων αντικειμένων. Σκοπός, λοιπόν, ενός συστήματος συστάσεων είναι δεδομένου του utility matrix να προβλέψει τις κενές τιμές του πίνακα, την αξιολόγηση δηλαδή που θα δώσει ο χρήστης στα αντικείμενα που δεν έχει ακόμα αξιολογήσει.

Σύμφωνα με τους Melville and Sindhwani (Melville and Sindhwani, 2010), τα συστήματα Collaborative filtering συγκεντρώνουν τις προτιμήσεις των χρηστών για τα αντικείμενα με τη μορφή βαθμολογιών και αναζητούν για ομοιότητες στις

αξιολογήσεις των χρηστών, για να προβλέψουν την βαθμολογία των αντικειμένων που δεν έχουν ακόμα αξιολογηθεί από τους χρήστες. Με βάση τις προβλέψεις αυτές θα αποφασίσουν αν θα προτείνουν ή όχι ένα αντικείμενο στον χρήστη. Τα collaborative συστήματα συστάσεων μπορούν να κατηγοριοποιηθούν σε Neighborhood-based και Model-based, ανάλογα με την προσέγγιση που ακολουθούν για τις προβλέψεις τους. Τα Neighborhood-based συστήματα επιλέγουν ένα υποσύνολο των χρηστών με βάση την ομοιότητά τους με τον προς εξέταση χρήστη, όπως αυτή υπολογίζεται από παλαιότερες αξιολογήσεις, ενώ τα Model-based συστήματα υποθέτουν ότι υπάρχει κάποια χαμηλής διάστασης δομή στα δεδομένα η οποία προκαλεί την ομοιότητα των χρηστών με τα αντικείμενα (Melville and Sindhwani, 2010). Για παράδειγμα, η βαθμολογία ενός χρήστη για μία ταινία είναι πιθανό να εξαρτάται και από έμμεσους παράγοντες όπως η ιδιαίτερη προτίμησή του σε συγκεκριμένα είδη ταινιών. Όπως γίνεται κατανοητό, οι προσεγγίσεις των collaborative filtering συστημάτων χρησιμοποιούν μόνο τον πίνακα των αξιολογήσεων, ενώ χειρίζονται όλους τους χρήστες και τα αντικείμενα ως ξεχωριστές μονάδες, αγνοώντας τυχόν επιπλέον προκύπτουσα πληροφορία.

Στην περίπτωση αυτή βρίσκουν χρήση τα content-based συστήματα συστάσεων, τα οποία παρέχουν συστάσεις συγκρίνοντας το περιεχόμενο που περιγράφει ένα αντικείμενο με το περιεχόμενο που χαρακτηρίζει τις προτιμήσεις του χρήστη (Melville and Sindhwani, 2010). Πάνω σε αυτό το αντικείμενο μάλιστα έχει πραγματοποιηθεί αρκετή έρευνα, εστιάζοντας κυρίως στα περιεχόμενα που έχουν να κάνουν με αρχεία κειμένου, σελίδες με ειδησεογραφικό περιεχόμενο και γενικότερα σελίδες που βασίζονται σε προτάσεις σχετικές με λεκτικό περιεχόμενο (blogs, ηλεκτρονικές βιβλιοθήκες κλπ). Σε ένα content-based σύστημα, για κάθε αντικείμενο δημιουργείται ένα προφίλ το οποίο είναι μια συλλογή από τα πιο σημαντικά χαρακτηριστικά αυτού του αντικειμένου (Anand et al., 2012), (Pazzani and Billsus, 2007). Αν για παράδειγμα αναφερόμαστε σε ένα content-based σύστημα συστάσεων ταινιών, τότε τα χαρακτηριστικά αυτού του αντικειμένου θα μπορούσαν να είναι: (i) οι ηθοποιοί της ταινίας καθώς κάποιοι χρήστες μπορεί να προτιμούν κάποιους ηθοποιούς περισσότερο από κάποιους άλλους και άρα το συγκεκριμένο κριτήριο θα μπορούσε να χαρακτηρίσει και το περιεχόμενο που θα

περιέγραφε τους χρήστες, (ii) ο σκηνοθέτης της ταινίας καθώς κάποιοι χρήστες μπορεί να έχουν ιδιαίτερη προτίμηση στις ταινίες ενός συγκεκριμένου σκηνοθέτη, (iii) η χρονολογία παραγωγής της ταινίας καθώς μπορεί κάποιος χρήστης να προτιμάει παλαιότερες ή πιο πρόσφατες ταινίες και (iv) το είδος της ταινίας καθώς θεωρείται σχεδόν βέβαιο ότι κάθε χρήστης θα εξέφραζε κάποια ιδιαίτερη προτίμηση σε ταινίες ενός ή περισσότερων συγκεκριμένων ειδών και λιγότερη σε κάποια άλλα είδη.

Ωστόσο, όταν γίνεται αναφορά για λεκτικό περιεχόμενο, όπως η περιγραφή μιας ταινίας (ή ομοίως η κριτική ενός προϊόντος ή το κείμενο μιας είδησης), γίνεται αντιληπτό ότι η δυσκολία είναι να μπορέσουμε να ανιχνεύσουμε από το κείμενο αυτό τα σημαντικότερα χαρακτηριστικά του, γι' αυτό και χρησιμοποιούνται τεχνικές ανάλυσης συναισθήματος και άλλες τεχνικές για αυτό το σκοπό. Βέβαια, εκτός από την περιγραφή του περιεχομένου για τα αντικείμενα χρειάζεται να δημιουργήσουμε και την περιγραφή των χρηστών με αντίστοιχο τρόπο ώστε να περιγράψουμε και τις προτιμήσεις τους. Από τον πίνακα αξιολογήσεων που αναφέρθηκε νωρίτερα μπορούμε να συνοψίσουμε τα περιεχόμενα των αντικειμένων για τα οποία υπάρχει κάποια αξιολόγηση από τον χρήστη. Εναλλακτικά, όπως αναφέρεται σε προηγούμενες μελέτες των Anand et al. (Anand et al., 2012) και Pazzani and Billsus (Pazzani and Billsus, 2007), αντί της δημιουργίας ενός προφίλ για τους χρήστες μπορούμε να δημιουργούμε έναν ταξινομητή για κάθε χρήστη, όπως για παράδειγμα τα δέντρα απόφασης, όπου οι γραμμές του πίνακα αξιολόγησης θα αποτελούν τα δεδομένα εκπαίδευσης και ο ταξινομητής θα πρέπει να είναι σε θέση να προβλέψει τα αποτελέσματα για κάθε αντικείμενο του πίνακα. Κατά συνέπεια, έχοντας δημιουργήσει μια περιγραφή του περιεχομένου των αντικειμένων αλλά και των προτιμήσεων των χρηστών, είναι εφικτό να υπολογίσουμε με διάφορα μέτρα (συνήθως με το cosine distance) την απόσταση-ομοιότητα των δύο αυτών διανυσμάτων και έτσι να εκτιμήσουμε αν ο χρήστης θα προτιμούσε ή όχι ένα αντικείμενο.

Η σύγχρονη τάση ωστόσο είναι τα υβριδικά μοντέλα συστημάτων συστάσεων. Αυτό οφείλεται αφενός στην διαρκή απαίτηση για όλο και ακριβέστερες προβλέψεις-προτάσεις στους χρήστες και αφετέρου στα επιμέρους

μειονεκτήματα της κάθε τεχνικής, είτε είναι collaborative filtering, είτε είναι content-based. Σκοπός λοιπόν των υβριδικών συστημάτων είναι να εκμεταλλευτούν τα πλεονεκτήματα των collaborative filtering και content-based συστημάτων, συνδυάζοντας τεχνικές και αρχιτεκτονικές και από τις δύο αυτές κατηγορίες με σκοπό την βελτίωση της απόδοσης των συστάσεων στους χρήστες (Anand et al., 2012), (Melville and Sindhwani, 2010). Για τους παραπάνω λόγους, υπάρχει αρκετή έρευνα γύρω από αυτό το αντικείμενο και έχουν προταθεί αρκετά υβριδικά μοντέλα συστημάτων συστάσεων. Μια απλοποιημένη προσέγγιση αυτού του προβλήματος την οποία έχουν προτείνει οι Cotter and Smyth (Cotter and Smyth, 2000), είναι η παραγωγή συστάσεων τόσο με collaborative filtering όσο και με content-based μεθόδους και έπειτα ο συνδυασμός των δύο προβλέψεων για να προκύψει η τελική λίστα προβλέψεων για τον χρήστη.

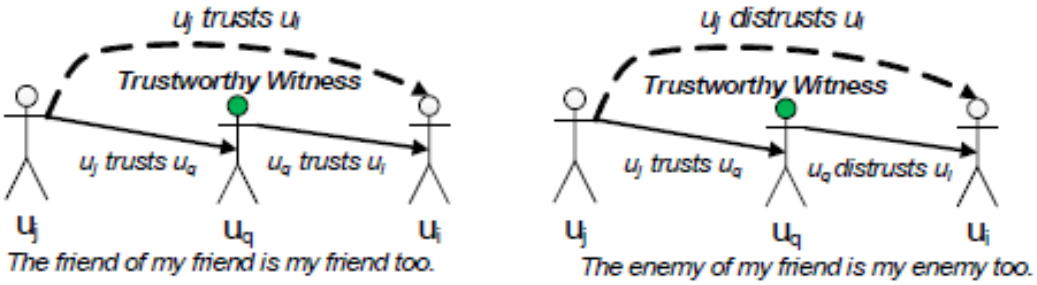
Από τις πιο ενδιαφέρουσες προσεγγίσεις υβριδικών συστημάτων είναι αυτή των Melville et al. (Melville et al., 2002) οι οποίοι υποστηρίζουν ότι σύμφωνα με τα πειράματά τους, το μοντέλο Content-boosted Collaborative Filtering που προτείνουν αποδίδει καλύτερα απ’ ότι αν ακολουθηθούν μεμονωμένα collaborative filtering ή content-based μέθοδοι. Σύμφωνα με τη μελέτη τους, η μέθοδος που προτείνουν αξιοποιεί τα πλεονεκτήματα του collaborative filtering με αποτέλεσμα να μπορεί να εκτελείται σε περιπτώσεις όπου δεν υπάρχει κάποιο ιδιαίτερο περιεχόμενο που μπορεί να αντιστοιχηθεί με τα αντικείμενα ή σε περιπτώσεις όπου το περιεχόμενο είναι δύσκολο να αναλυθεί από έναν υπολογιστή, καθώς και την ικανότητά του να παρέχει προτάσεις για αντικείμενα τα οποία είναι σχετικά με τον χρήστη τα οποία όμως δεν περιέχουν περιγραφές ή άλλο περιεχόμενο που να είναι σχετικό με το προφίλ του χρήστη. Απ’ την άλλη, αξιοποιούν παράλληλα και την ικανότητα των content-based μεθόδων να μπορούν να χαρακτηρίσουν μοναδικά κάθε χρήστη. Για τα πειράματά τους χρησιμοποίησαν ένα σύνολο δεδομένων σχετικό με ταινίες και βαθμολογίες των χρηστών για αυτές. Έτσι, συγκέντρωσαν το περιεχόμενο της κάθε ταινίας από το IMDB καθώς και τον πίνακα αξιολογήσεων των ταινιών από τους χρήστες. Το πρόβλημα όμως στη συγκεκριμένη περίπτωση είναι ότι ο πίνακας αξιολογήσεων των ταινιών περιέχει πολλά κενά, καθώς κάθε χρήστης έχει αξιολογήσει ένα μικρό τμήμα από το σύνολο των ταινιών. Σε αυτό το στάδιο

χρησιμοποιήθηκε το content-based σύστημα το οποίο εκπαιδεύτηκε για κάθε διάνυσμα χρήστη-βαθμολογιών και από το οποίο προέκυψε ένα ψευδο-διάνυσμα χρήστη-βαθμολογιών το οποίο περιέχει τις πραγματικές βαθμολογίες του χρήστη για τις ταινίες που έχει αξιολογήσει και την βαθμολογία που υπολόγισε το content-based σύστημα για τις ταινίες που δεν είχε αξιολογήσει ο χρήστης. Από το σύνολο των ψευδο-διανυσμάτων για κάθε χρήστη προέκυψε ο ψευδο-πίνακας βαθμολογιών, ο οποίος πλέον δεν περιέχει κενές τιμές. Στο τελικό στάδιο του μοντέλου που προτείνεται από τους Melville et al. (Melville et al., 2002), χρησιμοποιείται collaborative-filtering στα δεδομένα του ψευδο-πίνακα για τον υπό εξέταση χρήστη προκειμένου να γίνει η πρόβλεψη για νέα αντικείμενα. Καταλήγοντας στα συμπεράσματά τους οι ερευνητές, αποδεικνύουν ότι το υβριδικό μοντέλο συστάσεων που προτείνουν αποδίδει κατά μέσο όρο έως και 9.2% καλύτερα από το content-based μοντέλο και 4.0-4.6% καλύτερα από το collaborative filtering μοντέλο.

Συχνά όμως, στις σελίδες που χρησιμοποιείται κάποιο σύστημα συστάσεων υπάρχει και κάποιου είδους πληροφορία κοινωνικού δικτύου, η οποία μπορεί για παράδειγμα να εκφράζεται στην πιο απλή της μορφή με το αν κάποιος χρήστης έχει δηλώσει άμεσα ή έμμεσα ότι εμπιστεύεται ή όχι κάποιον άλλο χρήστη. Αυτό μπορεί να γίνει άμεσα όταν ο χρήστης εκδηλώνει την προτίμησή του για κάποιον άλλο χρήστη (προσθέτοντάς τον για παράδειγμα στη λίστα φίλων του ή σε κάποια κατηγορία αξιόπιστων χρηστών). Έμμεσα, μπορεί να εκδηλωθεί με διάφορους τρόπους. Μπορεί για παράδειγμα να εκδηλωθεί κάνοντας απλά Like σε κάποια ανάρτηση ενός άλλου χρήστη, κάνοντας retweet μια δημοσίευση ή δηλώνοντας ότι κάποιο άρθρο ενός χρήστη, όπως για παράδειγμα μία αξιολόγηση για ένα προϊόν, φάνηκε χρήσιμη.

Αυτό λοιπόν που προτείνεται από τους Varlamis et al. (Varlamis et al., 2013), είναι ένα πλαίσιο για την διαχείριση της εμπιστοσύνης στα κοινωνικά δίκτυα, το οποίο βασίζεται σε ένα μηχανισμό που λαμβάνει υπόψη του τις άμεσες ή έμμεσες συνδέσεις μεταξύ των χρηστών, τις αναλύει και παρέχει εξατομικευμένες προτάσεις χρηστών στους χρήστες. Πιο συγκεκριμένα, η λειτουργία του συστήματος διακρίνεται σε τρεις φάσεις. Στην πρώτη φάση, δημιουργούνται οι έμμεσες ή

άμεσες συνδέσεις μεταξύ των χρηστών του δικτύου, στην δεύτερη φάση υπολογίζονται οι αξιολογήσεις της φήμης των χρηστών και στην τρίτη φάση παράγονται οι εξατομικευμένες προτάσεις για κάθε χρήστη. Με αυτόν τον τρόπο, ενισχύεται η αποτελεσματικότητα των συστάσεων του συστήματος καθώς με την χρήση του χαρακτηριστικού της εμπιστοσύνης μεταξύ των χρηστών, θεωρείται πιο πιθανό οι προτιμήσεις ενός χρήστη να σχετίζονται περισσότερο με τον κοινωνικό του κύκλο (στο κοινωνικό του δίκτυο) και κατά συνέπεια το σύστημα μπορεί να συγκλίνει ακόμα περισσότερο σε συστάσεις που ταιριάζουν περισσότερο στο χρήστη. Όσον αφορά την μεταβατικότητα της αξιοπιστίας στο κοινωνικό δίκτυο, γίνεται η παραδοχή ότι “Ο φίλος του φίλου μου είναι και δικός μου φίλος” και ότι “Ο εχθρός του φίλου μου είναι και εχθρός μου” όπως φαίνεται και στην εικόνα 2. Με αυτό τον τρόπο, το σύστημα έχει μια ευρύτερη άποψη για την αξιοπιστία των χρηστών στο κοινωνικό δίκτυο, την οποία συμπεριλαμβάνει κατά την επιλογή των χρηστών στους υπολογισμούς των συστάσεων για τον υπό εξέταση χρήστη.



Εικόνα 1: Μεταβατικότητα της εμπιστοσύνης (πηγή: Varlamis et al., 2013)

2.2 Αλγόριθμοι Collaborative Filtering

2.2.1 User-User

Σύμφωνα με τη βιβλιογραφία, όπως αναφέρουν οι Malone et al. (Malone et al., 1987), οι τεχνικές φιλτραρίσματος της πληροφορίας χωρίζονται σε τρεις κατηγορίες – γνωστική, κοινωνική και οικονομική – ανάλογα με την πηγή πληροφοριών που χρησιμοποιείται για την εξαγωγή συμπερασμάτων για την προτίμηση ή μη ενός άρθρου από έναν χρήστη. Άλλες content-based τεχνικές, βασίζονται στην δημιουργία ενός προφίλ για τον χρήστη, το οποίο «εκπαιδεύεται» με βάση την ανάδραση που δέχονται από τον χρήστη για το αν του άρεσε ή όχι το εκάστοτε άρθρο που του προτάθηκε από το μέχρι πρότινος προφίλ του. Οι τεχνικές για την ανανέωση του προφίλ του χρήστη βασίζονται σε τεχνικές μηχανικής μάθησης ή σε πιθανοτικές τεχνικές (Belkin and Croft, 1992). Η κατηγορία κοινωνικών τεχνικών φιλτραρίσματος της πληροφορίας απ' την άλλη, βασίζεται στην κοινωνική σχέση μεταξύ των ατόμων για τη λήψη αποφάσεων για το ενδιαφέρον ή όχι ενός χρήστη για ένα άρθρο. Οι πολλά υποσχόμενες τεχνικές Collaborative Filtering πάνε ένα βήμα παραπέρα τις κοινωνικές τεχνικές αναζήτησης, αντιμετωπίζοντας όμως προβλήματα όταν μιλάμε για περιεχόμενο κειμένου, όπως ο υπολογισμός του βαθμού σχετικότητας μεταξύ κειμένων που διαφέρουν στην ορολογία αλλά συμφωνούν στο περιεχόμενο (π.χ. λόγω χρήσης διαφορετικών συνώνυμων λέξεων) ή ακόμα πιο σημαντικά χαρακτηριστικά που είναι δύσκολο να καταλάβει μία μηχανή, όπως η ποιότητα ενός κειμένου ή η αξιοπιστία του συγγραφέας του.

Ο User-User αλγόριθμος για Collaborative Filtering που προτάθηκε από τους Resnick et al. (Resnick et al., 1994) βασίζεται στο σύστημα Tapestry των Goldberg et al. (Goldberg et al., 1992) επεκτείνοντάς το με δύο τρόπους. Καταρχήν, σε αντίθεση με το Tapestry, τα δεδομένα βαθμολογιών δεν βασίζονται σε μία μόνο στατική σελίδα, αλλά συλλέγονται από διαφορετικές σελίδες, και δευτερευόντως, ο αλγόριθμός τους σε αντίθεση με το σύστημα Tapestry, περιέχει συγκεντρωτικά ερωτήματα. Επιπλέον, σε αντίθεση με μία άλλη έρευνα του Maltz (Maltz, 1994), ο αλγόριθμος των Resnick et al. προσαρμόζει τα σκορ των προβλέψεων για κάθε

χρήστη ξεχωριστά, αξιοποιώντας τις διαφορετικές προτιμήσεις ανάμεσα στους διάφορους χρήστες.

Το μοντέλο του User-User αλγόριθμου βασίζεται στην υπόθεση ότι αν τα ενδιαφέροντα δύο χρηστών έχουν ταυτιστεί στο παρελθόν, τότε είναι πολύ πιθανό να ξανασυμβεί στο μέλλον, θεωρώντας ότι τα ενδιαφέροντα των χρηστών δεν αλλάζουν ραγδαία. Κατά την υλοποίηση του, ο αλγόριθμος συσχετίζει τις υπάρχουσες βαθμολογίες των χρηστών για τα αντικείμενα, αποδίδοντας έτσι ένα βάρος σε κάθε χρήστη, το οποίο θα χρησιμοποιήσει αργότερα προσπαθώντας να κάνει προβλέψεις για κάποιον άλλο χρήστη. Αν $\rho_{X,Y}$ είναι ο συντελεστής συσχέτισης μεταξύ των μεταβλητών X και Y, στην περίπτωσή μας οι βαθμολογίες του υπό εξέταση χρήστη με κάποιον άλλο, με τυπικές αποκλίσεις σ_X και σ_Y αντίστοιχα, τότε ο **συντελεστής συσχέτισης** ορίζεται ως:

$$\rho_{X,Y} = corr(X,Y) = \frac{cov(X,Y)}{\sigma_X \sigma_Y}, \quad \begin{matrix} corr: correlation\ coefficient \\ cov: covariance \end{matrix}$$

με

$$cov(X,Y) = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{n-1}$$

Αν τώρα έχουμε μια σειρά τιμών x_i και y_i αντίστοιχα, για τις μεταβλητές X και Y, όπου $i \geq 1$, τότε μπορούμε να χρησιμοποιήσουμε τους συντελεστές συσχέτισης δείγματος για τον υπολογισμό της συσχέτισης **Pearson**:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Κατά σειρά, πρώτα υπολογίζονται οι συντελεστές συσχέτισης (Pindyck and Rubinfeld, 1991), μεταξύ του υπό εξέταση χρήστη και των υπόλοιπων χρηστών, με αποτέλεσμα να ληφθούν υπόψη μόνο οι αντίστοιχες βαθμολογίες των κοντινότερα συσχετιζόμενων χρηστών. Με αυτόν τον τρόπο, η βαθμολογία ενός αντικειμένου που προβλέπει το σύστημα για τον υπό εξέταση χρήστη, θα είναι σχετική με την βαθμολογία που έχουν δώσει για αυτό το αντικείμενο οι χρήστες με τους οποίους ο

υπό εξέταση χρήστης έχει συμφωνήσει ξανά στο παρελθόν, ή προκύπτουν να είναι οι πιο κοντινοί γείτονες μετά τον υπολογισμό των ομοιοτήτων.

Συνοπτικά, ο αλγόριθμος επιστρέφει ως συστάσεις τα Ν πρώτα αντικείμενα του πιο όμοιου με τον υπό εξέταση χρήστη, διαφορετικά αν δεν βρεθεί όμοιος με τον υπό εξέταση χρήστη, τότε δεν είναι δυνατή η παραγωγή συστάσεων. Επίσης, η πρόβλεψη για την βαθμολογία ενός αντικειμένου βασίζεται στην βαθμολογία του πιο όμοιου γείτονα, ο οποίος έχει ήδη βρεθεί από τον αλγόριθμο. Σε αντίθετη περίπτωση, αν δεν είναι δυνατό να βρεθεί κάποιος όμοιος γείτονας, χρησιμοποιείται ο μέσος όρος των βαθμολογιών όλων των χρηστών που έχουν βαθμολογήσει αυτό το αντικείμενο ως αποτέλεσμα της πρόβλεψης της βαθμολογίας του υπό εξέταση χρήστη για αυτό το αντικείμενο.

2.2.2 FunkSVD

Ο αλγόριθμος FunkSVD, ο οποίος δημιουργήθηκε από τον Simon Funk[1], είναι ένα είδος SVD αλγορίθμου, διαφοροποιείται όμως στον τρόπο υπολογισμού των προβλέψεων, καθώς χρησιμοποιεί τη μέθοδο gradient descent για τον προσδιορισμό του πίνακα παραγοντοποίησης. Μάλιστα, είναι ο αλγόριθμος που έλαβε την τρίτη θέση στον διαγωνισμό του Netflix για προτάσεις αλγορίθμων πρόβλεψης βαθμολογίας των χρηστών για ταινίες. Την πρώτη θέση και το έπαθλο του 1εκ. δολαρίων στον σχετικό διαγωνισμό έλαβε η ομάδα BellKor's Pragmatic Chaos[2], ο αλγόριθμος της οποίας πέτυχε την καλύτερη απόδοση RMSE με τιμή ίση με 0.8567, βελτιώνοντας την αρχική απόδοση κατά 10%.

Μπορούμε να πούμε ότι ο αλγόριθμος FunkSVD, βασίζεται στην παραδοχή πως αν μπορείς σε ένα πρόβλημα δεδομένων να παραστήσεις τα δεδομένα με λιγότερες διαστάσεις, μειώνοντας έτσι το χώρο στον οποίο θα γίνονται οι υπολογισμοί και άρα την πολυπλοκότητα, μπορείς πιο εύκολα να καταλήξεις σε γενικευμένα συμπεράσματα με εννοιολογικό περιεχόμενο. Πρακτικά αυτό σημαίνει ένα μοντέλο για την αναπαράσταση των αντικειμένων από μία μικρότερη ομάδα χαρακτηριστικών και μία μέθοδο για την εξαγωγή των τιμών αυτών των

[1] http://sifter.org/~simon/journal/20061211.html
[2] http://www.research.att.com/~volinsky/netflix/bpc.html

χαρακτηριστικών από τα πραγματικά δεδομένα. Στην πραγματικότητα λοιπόν, ο FunkSVD αλγόριθμος αξιοποιεί την τεχνική του Singular Value Decomposition. Αν θέλαμε να περιγράψουμε αυτήν την τεχνική με ένα παράδειγμα, θα λέγαμε ότι η βαθμολογία ενός χρήστη για μία ταινία απεικονίζει το σύνολο της προτίμησης του χρήστη για τα διαφορετικά χαρακτηριστικά μιας ταινίας. Αν λοιπόν είχαμε την ταινία (movie) με τα χαρακτηριστικά (f) και τον χρήστη (user) τότε:

$$ratingsMatrix[user][movie] = \sum (userFeature[f][user] * movieFeature[f][movie])$$

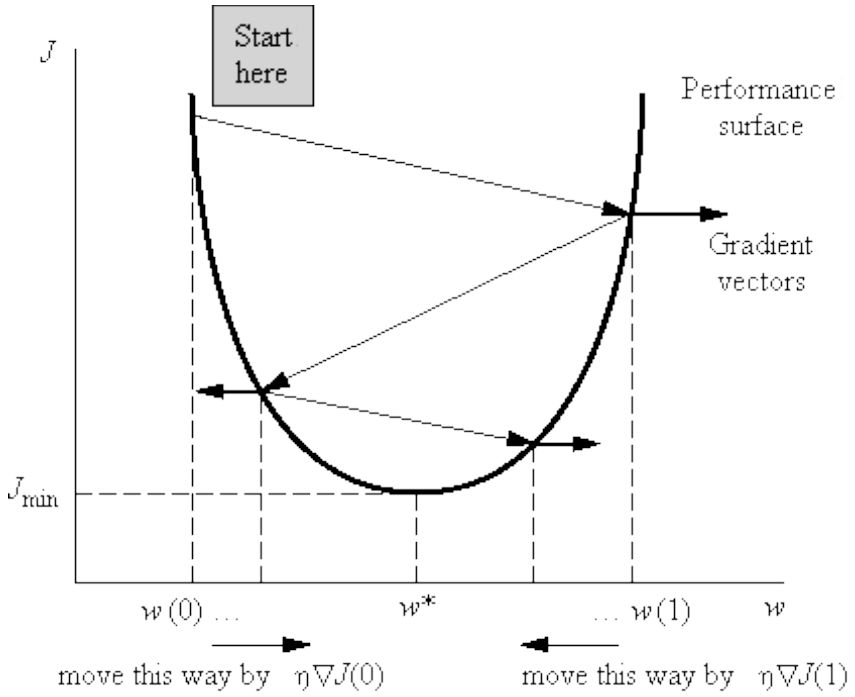
Έτσι, με την μέθοδο του Singular Value Decomposition, ο μεγάλος αρχικός πίνακας $users \times items$ μετατρέπεται σε δύο πίνακες $users \times features$ και $items \times features$, όπου το εσωτερικό γινόμενο των δύο πινάκων δίνει το άθροισμα των γινομένων της προηγούμενης σχέσης. Η χρησιμότητα αυτής της μεθόδου έγκειται στο γεγονός ότι παράγοντας τους δύο νέους, μικρότερους πίνακες αντί του ενός μεγάλου, ελαχιστοποιείται το τελικό σφάλμα υπολογισμού, συγκεκριμένα το μέσο τετραγωνικό σφάλμα. Παραμένει ωστόσο, το πρόβλημα του υπολογισμού του SVD σε μεγάλους πίνακες, ιδιαίτερα σε καθημερινά πραγματικά σύνολα δεδομένων. Το πρόβλημα μετατρέπεται σε πρόβλημα υπολογισμού των παραγόμενων χαρακτηριστικών, από τον αρχικό πίνακα βαθμολογιών και στη μετατροπή του διανύσματος βαθμολογιών για τα αντικείμενα σε βαθμολογίες για τα παραγόμενα χαρακτηριστικά. Η προσέγγιση του FunkSVD σε αυτό, είναι η λήψη του σφάλματος μεταξύ της τιμής που προβλέφθηκε και της πραγματικής τιμής και έπειτα η εύρεση των παραγώγων σε σχέση με τις παραμέτρους που προσπαθούμε να συμπεράνουμε. Όσον αφορά τον ρυθμό μάθησης της καμπύλης (learning rate) της μεθόδου Βηματικής Κατάβασης (Gradient Descent), αυτός αρχικοποιείται σε μια μικρή τιμή (π.χ. lrate=0.001), ώστε να επιτυγχάνει την μικρότερη δυνατή μετατόπιση της καμπύλης σε κάθε επανάληψη για τον υπολογισμό του σφάλματος.

Πριν ο αλγόριθμος ξεκινήσει να μαθαίνει τα χαρακτηριστικά (features) των αντικειμένων, υπολογίζει μία μέση βαθμολογία για κάθε αντικείμενο, ως βάση σύγκρισης για τη συνέχεια. Έτσι, για κάθε αντικείμενο, υπολογίζει το άθροισμα της μέσης βαθμολογίας του από όλους τους χρήστες με την διαφορά της βαθμολογίας του κάθε χρήστη από τον μέσο όρο των βαθμολογιών για αυτό το αντικείμενο, όπως

φαίνεται στον παρακάτω τύπο και ξεκινάει την εκπαίδευση των χαρακτηριστικών, με σκοπό την σταδιακή μείωση του σφάλματος.

$$predictRating_{Baseline} = averageRating[item] + averageOffset[user]$$

Συνεχίζοντας λοιπόν την διαδικασία εκπαίδευσης, στόχος είναι η προσέγγιση του τοπικού βέλτιστου (ελάχιστου) σε μία gradient descent κατανομή όπως η παρακάτω. Μόλις ο αλγόριθμος συγκλίνει όσο γίνεται πιο κοντά στο τοπικό ελάχιστο, έχοντας δηλαδή μειώσει το σφάλμα όσο το δυνατό περισσότερο για ένα χαρακτηριστικό, τότε το προσθέτει στη λίστα με τα χαρακτηριστικά και προχωράει στην εκ νέου εκπαίδευση του επόμενου χαρακτηριστικού.



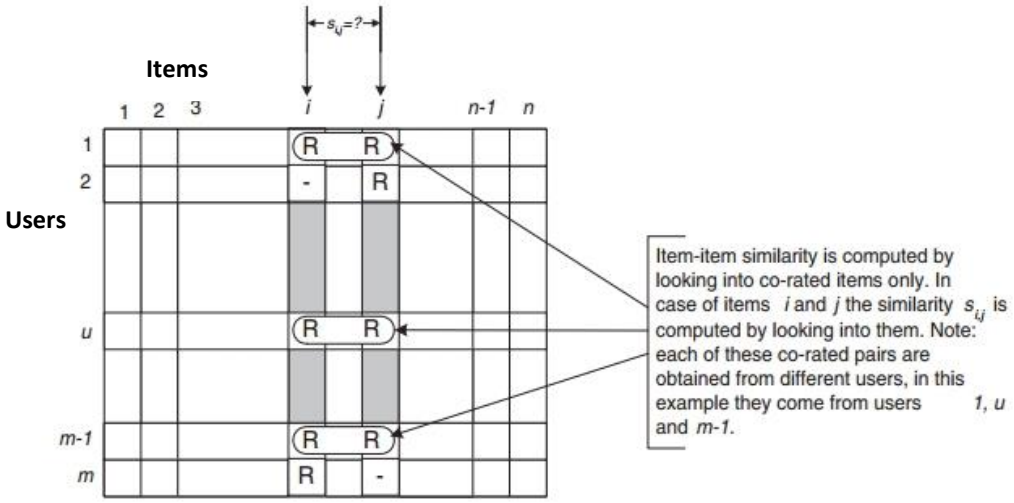
Εικόνα 2: Gradient Descent κατανομή (πηγή: Principe et al., 1999)

2.2.3 Item-Item

Όσον αφορά τον αλγόριθμο Item-Item, οι Sarwar et al. (Sarwar et al., 2001) και (Deshpande and Karypis, 2004), προσέγγισαν τους περιορισμούς των συμβατικών αλγόριθμων collaborative filtering, όπως είχαν ορίσει οι Herlocker et al. (Herlocker et al., 1999), ως την δυσκολία της αναζήτησης αξιόπιστων γειτόνων ανάμεσα σε μία μεγάλη κοινότητα υποψήφιων γειτόνων. Η πρότασή τους λοιπόν, είναι πως η λύση στο παραπάνω πρόβλημα μπορεί να δοθεί από τους Item-Item αλγορίθμους. Σε αντίθεση με τον αλγόριθμο User-User, όπου εξερευνούνται οι

σχέσεις μεταξύ των χρηστών για την εύρεση όμοιων χρηστών, εδώ εξερευνούνται οι σχέσεις μεταξύ των αντικειμένων και κατά συνέπεια, οι συστάσεις για τους χρήστες προκύπτουν βρίσκοντας ομοιότητες ανάμεσα στα αντικείμενα που έχει ήδη βαθμολογήσει ο χρήστης και σε αυτά που δεν έχει αξιολογήσει ακόμα. Οι ερευνητές, προσθέτουν επίσης στα πλεονεκτήματα αυτής της κατηγορίας αλγορίθμων, την δυνατότητα παραγωγής εξίσου αξιόπιστων συστάσεων σε σχέση με τους User-User αλγόριθμους και με τη χρήση λιγότερης υπολογιστικής ισχύος. Για την εύρεση ομοιοτήτων ανάμεσα στα αντικείμενα και την μετέπειτα σύσταση των k μεγαλύτερων αντικειμένων υπάρχουν διάφορες τεχνικές.

Υποθέτοντας την σύγκριση ομοιότητας ανάμεσα σε δύο αντικείμενα i και j, πρώτο στάδιο είναι η απομόνωση των χρηστών που έχουν αξιολογήσει και τα δύο αυτά αντικείμενα και δεύτερο στάδιο είναι η εφαρμογή τεχνικών υπολογισμού της ομοιότητας $s_{i,j}$. Η Εικόνα 3 απεικονίζει την παραπάνω διαδικασία υπολογισμού ομοιότητας αντικειμένων.



Εικόνα 3: Διαδικασία υπολογισμού ομοιότητας αντικειμένων (πηγή: Sarwar et al., 2001)

Για τον υπολογισμό της ομοιότητας σε αυτή την κατηγορία αλγορίθμων χρησιμοποιείται είτε η συσχέτιση κατά **Pearson**, που περιγράφεται στην ενότητα 2.2.1, είτε η ομοιότητα συνημίτονου (**cosine similarity**), είτε η ομοιότητα προσαρμοσμένου συνημίτονου (**adjusted-cosine similarity**). Η ομοιότητα συνημίτονου για τα αντικείμενα i και j της Εικόνας 3, ορίζεται ως εξής:

$$sim(i,j) = \cos(\vec{i},\vec{j}) = \frac{\vec{i} \cdot \vec{j}}{\|\vec{i}\|_2 * \|\vec{j}\|_2}$$

Μία από τις βασικότερες διαφορές στον τρόπο υπολογισμού της ομοιότητας ανάμεσα στους User-User και τους Item-Item αλγόριθμους, είναι ότι στους User-User η ομοιότητα υπολογίζεται για κάθε γραμμή του πίνακα, δηλαδή για κάθε χρήστη, ενώ στους Item-Item αλγόριθμους υπολογίζεται ανάμεσα στις στήλες του πίνακα, όπως για παράδειγμα προηγουμένως, όπου κάθε ζεύγους αντικειμένων που έχουν βαθμολογηθεί αντιστοιχούν σε διαφορετικό χρήστη. Ωστόσο, ο υπολογισμός της ομοιότητας με το μέτρο του συνημίτονου, έχει ως ελάττωμα την έλλειψη υπολογισμού της διαφοράς στην κλίμακα των βαθμολογιών των διάφορων χρηστών. Αν λοιπόν, από κάθε ζεύγος βαθμολογημένων αντικειμένων αφαιρεθεί ο αντίστοιχος μέσος όρος του χρήστη, τότε καταλήγουμε στο μέτρο του προσαρμοσμένου συνημίτονου, όπως φαίνεται και στον ακόλουθο τύπο:

$$sim(i,j) = \frac{\sum_{u \in U}(R_{u,i} - \bar{R}_u)(R_{u,j} - \bar{R}_u)}{\sqrt{\sum_{u \in U}(R_{u,i} - \bar{R}_u)^2}\sqrt{\sum_{u \in U}(R_{u,j} - \bar{R}_u)^2}}, \bar{R}_u: Μέσος\ όρος\ βαθμολογιών\ του\ χρήστη\ u$$

Το επόμενο βήμα μετά τον υπολογισμό των ομοιοτήτων, είναι η παραγωγή των προβλέψεων. Μετά την δημιουργία λοιπόν του συνόλου των πιο όμοιων αντικειμένων, εξετάζονται οι βαθμολογίες του υπό εξέταση χρήστη, χρησιμοποιώντας είτε τεχνικές παλινδρόμησης (**Regression**), είτε τεχνικές σταθμισμένων αθροισμάτων (**Weighted Sum**) για να υπολογίσει τις προβλέψεις. Κατά την τεχνική σταθμισμένων αθροισμάτων για τον υπολογισμό της πρόβλεψης (**P**) ενός αντικειμένου (**i**) για έναν χρήστη (**u**), υπολογίζεται το άθροισμα των βαθμολογιών του χρήστη για τα όμοια του i αντικείμενα, όπως έχουν προκύψει από την προηγούμενη διαδικασία, συμπεριλαμβανομένου και του βάρους που έχει προκύψει από την ομοιότητα $S_{i,j}$ του αντικειμένου i με κάθε j αντικείμενο, όπως φαίνεται και στον παρακάτω τύπο.

$$P_{u,i} = \frac{\sum_{all\ similar\ items,N}(S_{i,N} * R_{u,N})}{\sum_{all\ similar\ items,N}(|S_{i,N}|)}$$

Κατά την τεχνική της παλινδρόμησης, από την άλλη, αντί να χρησιμοποιούνται οι βαθμολογίες των όμοιων αντικειμένων, χρησιμοποιούνται οι κατά προσέγγιση τιμές τους, βάση ενός μοντέλου παλινδρόμησης. Έτσι, παραμένει ο ίδιος τύπος υπολογισμού της πρόβλεψης όπως και προηγουμένως, αλλά για τις τιμές $R_{u,N}$ των βαθμολογιών, χρησιμοποιούνται οι προσεγγιστικές τιμές $R'_{u,N}$ που προκύπτουν από το παρακάτω μοντέλο γραμμικής παλινδρόμησης.

$$\overline{R'_N} = \alpha \overline{R_\iota} + \beta + \epsilon$$

2.3 Συναφείς Εργασίες

Οι διάφορες τεχνικές Collaborative Filtering καθώς και οι μέθοδοι clustering έχουν υπάρξει πολλές φορές αντικείμενο μελέτης. Σύμφωνα με τους Pham et al. (Pham et al., 2011), η μέθοδος ομαδοποίησης η οποία βασίζεται στις κοινωνικές πληροφορίες των χρηστών προκειμένου να διατυπωθούν συστάσεις μπορεί να αποδώσει καλύτερα από τις κλασικές τεχνικές Collaborative Filtering. Σε προηγούμενη έρευνα των Sarwar et al. (Sarwar et al., 2002), οι οποίοι είχαν εφαρμόσει μία προσέγγιση ομαδοποίησης χρηστών σε subgraphs με βάση την ομοιότητάς τους, είχαν δείξει ότι ήταν δυνατή η παραγωγή συστάσεων με αποτελέσματα συγκρίσιμα με αυτά της παραδοσιακής τεχνικής collaborative filtering. Επίσης, το 2006, σε έρευνα των Backstrom et al. (Backstrom et al., 2006) γίνεται εξέταση της σχέσης του collaborative filtering σε κοινωνικά σχηματισμένες ομάδες μεγάλων κοινωνικών δικτύων, ενώ το 2007, οι Anglade et al. (Anglade et al., 2007) πρότειναν μία σύνθετη προσέγγιση για προτάσεις μουσικής και ραδιοφωνικών καναλιών σε χρήστες P2P δικτύων.

Σε παλαιότερη έρευνά τους, οι O'Connor and Herlocker (O'Connor and Herlocker, 1999), είχαν θεωρήσει ότι το partitioning θα βελτίωνε την απόδοση των τεχνικών collaborative filtering. Για τον λόγο αυτό, είχαν ερευνήσει την απόδοση του collaborative filtering στο σύνολο δεδομένων του MovieLens, εφαρμόζοντας όμως πρώτα τεχνικές clustering στο σύνολο των αντικειμένων, καταλήγοντας σε ακριβέστερες προβλέψεις σε σχέση με τις κλασικές τεχνικές collaborative filtering.

Σε μία ακόμα σχετική έρευνα των De Meo et al. (De Meo et al., 2011), προτείνεται ο συνδυασμός τεχνικών collaborative filtering βασισμένες στους πίνακες παραγοντοποίησης και της πληροφορίας σχέσεων κοινωνικής φιλίας για την παραγωγή ακριβέστερων συστάσεων και την κατάταξη των αντικειμένων ενδιαφέροντός τους. Μάλιστα, η αξιολόγηση της προσέγγισής τους σε ένα πραγματικό κοινωνικό δίκτυο, δείχνει σχετική βελτίωση σε σχέση με τις υπάρχουσες κλασικές τεχνικές.

Επιπλέον, σε πρόσφατη έρευνα των Symeonidis et al. (Symeonidis et al., 2013) γίνεται παραλληλισμός της πρόβλεψης συνδέσμων για προτάσεις φιλίας στα

κοινωνικά δίκτυα με την πρόβλεψη συνδέσμων σε δίκτυα αλληλεπίδρασης πρωτεΐνης – πρωτεΐνης. Η προσέγγισή τους βασίστηκε στην εφαρμογή μίας φασματικής ομαδοποίησης πολλαπλών δρόμων, η οποία χρησιμοποιεί πληροφορίες από τα ελάχιστα υψηλά ιδιοδιανύσματα και ιδιοτιμές του κανονικοποιημένου Laplacian πίνακα, υπολογίζοντας με αυτόν τον τρόπο έναν μικρότερο πίνακα, με λιγότερο θόρυβο και πιο πυκνό σε σχέση με τον αρχικό, προτείνοντας πως με αυτόν τον τρόπο μπορούν να προκύψουν πιο γρήγορες και ακριβείς προτάσεις συνδέσμων.

Λαμβάνοντας, λοιπόν, υπόψη τα παραπάνω, ορίζεται ως προτεινόμενη μεθοδολογική προσέγγιση η δημιουργία ενός νέου μοντέλου, το οποίο ιδανικά θα βελτιώνει τα αποτελέσματα των παραδοσιακών τεχνικών, και μάλιστα με αποδοτικότερο και ταχύτερο χρόνο υπολογισμού. Η πρόταση που θα μελετήσουμε βασίζεται την εφαρμογή τεχνικών collaborative filtering, συνδυάζοντας χαρακτηριστικά από τα μοντέλα των (Melville et al., 2002), (Varlamis et al., 2013), αλλά σε εκ των προτέρων διχοτομημένους γράφους. Στόχος της έρευνας είναι να μελετήσει αν το προτεινόμενο μοντέλο μπορεί να αποδώσει καλύτερα από τα μέχρι τώρα γνωστά μοντέλα και πιθανότατα με συντομότερο χρόνο απόκρισης και να θέσει τις βάσεις για έναν πιο ευέλικτο και πιθανώς κατανεμημένο τρόπο λειτουργίας. Παράλληλα, στόχος της προσέγγισης, είναι η σύγκριση διαφορετικών αλγόριθμων collaborative filtering για να διαπιστωθεί ποιός μπορεί να αποδώσει αποδοτικότερα για το προτεινόμενο μοντέλο, χρησιμοποιώντας μια πληθώρα μετρικών αξιολόγησης.

Η αναγκαιότητα της μελέτης τεκμηριώνεται αν ληφθεί, για παράδειγμα, το γεγονός ότι το Facebook συλλέγει περίπου 4TB δεδομένων την ημέρα, γίνεται αντιληπτό πόσο σημαντικές πληροφορίες μπορούν να εξαχθούν από αυτά τα δεδομένα για κάθε χρήστη του. Το δεδομένο είναι ότι σίγουρα πληροφορίες από το κοινωνικό δίκτυο ενός χρήστη μπορούν, με διάφορους τρόπους, να διαδραματίσουν σημαντικό ρόλο στην παραγωγή εξατομικευμένων συστάσεων για χρήστες ή αντικείμενα. Σημαντικός είναι στην περίπτωση αυτή ο τρόπος της αξιοποίησης αυτής της πληροφορίας καθώς και η βελτιστοποίηση του τρόπου ανάλυσής της.

Για την υλοποίηση της μελέτης θα ληφθούν υπόψη τόσο οι γνώμες που καταθέτουν οι χρήστες για διαφορετικές ιδιότητες ενός αντικειμένου (aspect-based opinion mining), όσο και το κοινωνικό δίκτυο των χρηστών. Τα αποτελέσματα που θα προκύψουν θα συνεκτιμηθούν με αυτά των γνωστότερων τεχνικών συστάσεων και τέλος θα γίνει η κατάλληλη ερμηνεία των αποτελεσμάτων. Δευτερεύον σκοπός αυτής της έρευνας θα είναι η ανάδειξη τυχόν ευρημάτων κατά την διάρκεια εκτέλεσής της τα οποία θα χρήζουν περαιτέρω διερεύνησης και τα οποία θα μπορούσαν να αποτελέσουν νέες προτάσεις για μελλοντικές έρευνες.

2.4 Εργαλεία

Στην παρούσα διπλωματική εργασία για την παραγωγή του επιθυμητού μοντέλου χρησιμοποιήθηκαν συνδυαστικά δύο εργαλεία (frameworks) τα οποία προσαρμόστηκαν κατάλληλα στον υπάρχων java κώδικα για την εκτέλεση τμηματικών λειτουργιών. Αυτά ήταν η οικογένεια προγραμμάτων του **METIS** από το Karypis Lab και το **LensKit Recommender Toolkit**.

Η οικογένεια προγραμμάτων του Metis[3], αποτελεί τμήμα από μία σειρά ερευνητικών προγραμμάτων ανοιχτού κώδικα (open source), γραμμένα σε java, του Karypis Lab που εδρεύει στο Πανεπιστήμιο της Μινεσότα. Είναι ίσως η πιο μακροχρόνια ερευνητική δραστηριότητα του Καθηγητή Γεώργιου Καρύπη που ξεκίνησε ως έργο της διδακτορικής του διατριβής πριν πολλά χρόνια και έρχεται να δώσει λύση σε ένα βασικό πρόβλημα στην επεξεργασία των γράφων και αυτό είναι η διχοτόμηση ενός μεγάλου γράφου σε k τμήματα. Το παραπάνω πρόβλημα βρίσκει εφαρμογή σε πολλούς διαφορετικούς τομείς όπως τα παράλληλα και κατανεμημένα υπολογιστικά συστήματα για καταμερισμό του υπολογιστικού φόρτου (load-balancing), η εξόρυξη δεδομένων με στόχο την συσταδοποίηση (clustering), η ανάλυση των κοινωνικών δικτύων για την ανακάλυψη κοινοτήτων (community discovery), η ανακάλυψη προτύπων και πολλούς άλλους. Η διαίρεση του γράφου πραγματοποιείται με τέτοιο τρόπο ώστε αφενός να ικανοποιεί συγκεκριμένους περιορισμούς και να βελτιστοποιεί συγκεκριμένους σκοπούς. Ο συνηθέστερος περιορισμός είναι η παραγωγή υπογράφων ίσου μεγέθους, ενώ ο συνηθέστερος

[3] http://glaros.dtc.umn.edu/gkhome/

στόχος είναι η ελαχιστοποίηση των ακμών που βρίσκονται στα όρια διαφορετικών υπογράφων. Όπως όμως είναι φυσικό, στην πλειοψηφία των περιπτώσεων, διαφοροποιείται ο σκοπός του πειράματος και έτσι δημιουργούνται και διαφορετικοί περιορισμοί. Οι αλγόριθμοι που υλοποιούνται στο METIS βασίζονται σε πολυεπίπεδα σχήματα αναδρομικής διχοτόμησης, πολυεπίπεδα k-way σχήματα και σχήματα διχοτόμησης πολλών μεταβλητών, τα οποία αναπτύχθηκαν από το Karypis Lab.

Από τα βασικότερα χαρακτηριστικά του METIS που το ξεχωρίζουν από μια σειρά όμοιων εργαλείων είναι η ταχύτητα εκτέλεσής του. Χαρακτηριστικά οι δημιουργοί του αναφέρουν ότι ένας γράφος με πολλά εκατομμύρια ακμών, μπορεί να διαιρεθεί σε 256 τμήματα μέσα σε μερικά δευτερόλεπτα.

Όσον αφορά το LensKit Recommender Toolkit[4], είναι και αυτό ένα εργαλείο ανοιχτού λογισμικού γραμμένο σε java, το οποίο δημιουργήθηκε από την ερευνητική ομάδα GroupLens του Πανεπιστημίου της Μινεσότα. Σκοπός του είναι η κατασκευή, η μελέτη και η έρευνα των συστημάτων συστάσεων. Ο χρήστης μπορεί να χρησιμοποιήσει το εργαλείο είτε για να φτιάξει ένα σύστημα συστάσεων για να το χρησιμοποιήσει σε κάποια άλλη εφαρμογή του, μπορεί να μελετήσει τα συστήματα συστάσεων και την λειτουργία τους κατεβάζοντας την binary έκδοση που περιέχει όλο τον κώδικα του συστήματος καθώς και όλα τα εξαρτώμενα αρχεία, αλλά και scripts για την εκτέλεση του αξιολογητή (evaluator). Μπορεί τέλος να χρησιμοποιηθεί και για ερευνητικούς σκοπούς, είτε για τον πειραματισμό με νέους αλγόριθμους συστάσεων, είτε για την δοκιμή εναλλακτικών προσεγγίσεων εκτίμησης προβλέψεων. Επίσης, παρέχονται δύο Maven archetypes, το lenskit-archetype-simple-analysis το οποίο επιτρέπει την δημιουργία απλών project και το lenskit-archetype-fancy-analysis για την δημιουργία πιο σύνθετων project για την ανάλυση αλγόριθμων συστάσεων.

Το LensKit Recommender Toolkit περιέχει ένα σύνολο πακέτων από java κλάσεις, οι οποίες υλοποιούν την πλειοψηφία των συστατικών ενός συστήματος συστάσεων, ενώ καλύπτουν επίσης τις περισσότερες από τις μετρικές που

[4] http://lenskit.grouplens.org/

χρησιμοποιούνται για την αξιολόγηση της απόδοσης των μοντέλων πρόβλεψης, βαθμολόγησης – κατάταξης και σύστασης. Στις μετρικές περιλαμβάνεται η Ρίζα του Μέσου Τετραγωνικού Σφάλματος (**RMSE, Root Mean Square Error**), η οποία υπολογίζεται τόσο στο σύνολο των διαθέσιμων αξιολογήσεων (**RMSE_ByRating**) και κατά συνέπεια στο σύνολο των χρηστών, όσο και ανά χρήστη (**RMSE_ByUser**), καθώς και το μέσο απόλυτο σφάλμα (**MAE, Mean Absolute Error**), το οποίο και αυτό υπολογίζεται τόσο για το σύνολο των βαθμολογιών (**MAE**), όσο και ανά χρήστη (**MAE_ByUser**). Επίσης, από τα βασικά μέτρα που διατίθενται είναι η μετρική του Κανονικοποιημένου Εκπτωτικού Συγκεντρωτικού Κέρδους (**nDCG, normalized Discounted Cumulative Gain**), καθώς και το μέτρο της Κάλυψης (**Coverage**). Στις παραπάνω μετρικές πρέπει να προστεθεί και το Κανονικοποιημένο Εκπτωτικό Συγκεντρωτικό Κέρδος των top-N προτάσεων (**Top-N nDCG, Top-N normalized Discounted Cumulative Gain**) το οποίο είναι όμοιο με το nDCG, με τη διαφορά όμως ότι υπολογίζεται για τα top-N προϊόντα της κάθε πρόβλεψης, αντί στο σύνολο αυτών.

Όσον αφορά την οικογένεια αλγορίθμων που υλοποιεί το LensKit Recommender framework, αυτή περιλαμβάνει την υλοποίηση του Item-Item Collaborative Filtering (Sarwar et al., 2001; Deshpande and Karypis, 2004) ως τμήμα της μονάδας lenskit-knn, την υλοποίηση του αλγόριθμου για User-User Collaborative Filtering (Resnick et al., 1994), επίσης ως τμήμα της μονάδας lenskit-knn, την υλοποίηση ενός SVD αλγόριθμου για collaborative filtering (FunkSVD) ως τμήμα της μονάδας lenskit-svd και τέλος την υλοποίηση ενός Slope-One αλγόριθμου για την πρόβλεψη βαθμολογιών ως τμήμα της μονάδας lenskit-slopeone.

3 Προτεινόμενη Προσέγγιση

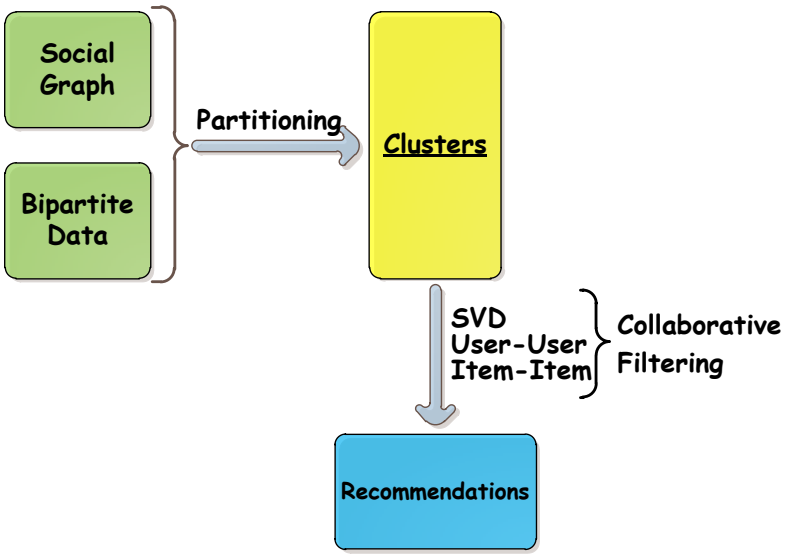
Όπως έχει αναφερθεί, στόχος της παρούσας έρευνας είναι η πρόταση ενός μοντέλου παραγωγής συστάσεων το οποίο όχι μόνο θα λαμβάνει υπόψη του τις γνώμες που καταθέτουν οι χρήστες για τα διάφορα αντικείμενα, αλλά θα αξιοποιεί και την πληροφορία του κοινωνικού δικτύου των χρηστών. Με αυτόν τον τρόπο, σκοπεύουμε να δώσουμε στο σύστημα ακόμα πιο στοχευμένη πληροφόρηση ώστε να είναι σε θέση να παράγει ακόμα πιο ακριβείς και αποτελεσματικές συστάσεις.

Σχετικά με τη σχεδίαση της προσέγγισης πρέπει να χρησιμοποιηθούν δύο σύνολα δεδομένων, ένα το οποίο θα περιέχει όλα τα δεδομένα αξιολόγησης των αντικειμένων από τους χρήστες και ένα το οποίο θα περιέχει τις λεπτομέρειες του κοινωνικού γράφου του δικτύου χρηστών για τον ίδιο ιστότοπο. Το μοντέλο θα πρέπει να είναι σε θέση να βρίσκει και να αξιοποιεί τους κοντινότερους γείτονες του κάθε χρήστη καθώς και τα αντικείμενα που έχουν οι ίδιοι αξιολογήσει και σε συνδυασμό με το «προφίλ» που προκύπτει για το χρήστη από τα αντικείμενα που έχει ήδη αξιολογήσει ο ίδιος να του προτείνει τα πρώτα Ν προϊόντα τα οποία είναι πιο πιθανό να τον ενδιαφέρουν.

Γίνεται λοιπόν η θεώρηση, πως το πρώτο και σημαντικό βήμα προς αυτήν την κατεύθυνση είναι ο διαμερισμός του αρχικού κοινωνικού γράφου των χρηστών σε μικρότερα τμήματα (**partitions** ή **subgraphs**). Η τμηματοποίηση του γράφου σε μικρότερους υπογράφους και ο διαμοιρασμός των χρηστών σε αυτούς στοχεύει αφενός στην βελτιστοποίηση του χρόνου υπολογισμού των συστάσεων από το σύστημα και αφετέρου στην προσπάθεια εξαγωγής προβλέψεων με αποδοτικότερο τρόπο εφόσον το σύστημα πλέον δεν αναζητεί για τις καλύτερες και πιο εξατομικευμένες προτάσεις στον πλήρη κοινωνικό γράφο, αλλά μόνο στο υπογράφο στον οποίο ανήκει ο εκάστοτε χρήστης. Ένας επιμέρους στόχος της παρούσας έρευνας ήταν η αναζήτηση και ο προσδιορισμός του βέλτιστου αριθμού υπογράφων για το συγκεκριμένο σύνολο δεδομένων. Εδώ να σημειωθεί ότι σε αντίθεση με άλλες έρευνες, το μοίρασμα το γράφου σε υπογράφους έγινε με μόνο κριτήριο την ισομερή κατανομή των χρηστών σε αυτούς. Έτσι, αποφεύγονται τυχόν περιορισμοί του κοινωνικού δικτύου ως προς τις σχέσεις φιλίας, δεν προστίθεται

επιπλέον φόρτος στο σύστημα για τον υπολογισμό των βέλτιστων σύμφωνα με τις σχέσεις φιλίας υπογράφους και το σημαντικότερο, εξασφαλίζεται ότι δεν θα υπάρξουν πολύ αραιοί υπογράφοι, με ελάχιστους χρήστες, για τους οποίους θα ήταν ίσως και αδύνατος ο προσδιορισμός κάποιων σχετικών συστάσεων. Το μόνο μειονέκτημα είναι ότι αν οι επιμέρους διμερείς γράφοι βαθμολογιών (Bipartite Ratings Graphs) που προκύπτουν είναι αρκετά αραιοί, δηλαδή έχουμε λίγες βαθμολογίες ή βαθμολογίες για λίγα προϊόντα, ενδέχεται ορισμένοι από τους αλγορίθμους συνεργατικού φιλτραρίσματος (collaborative filtering) να μην αποδίδουν καλά.

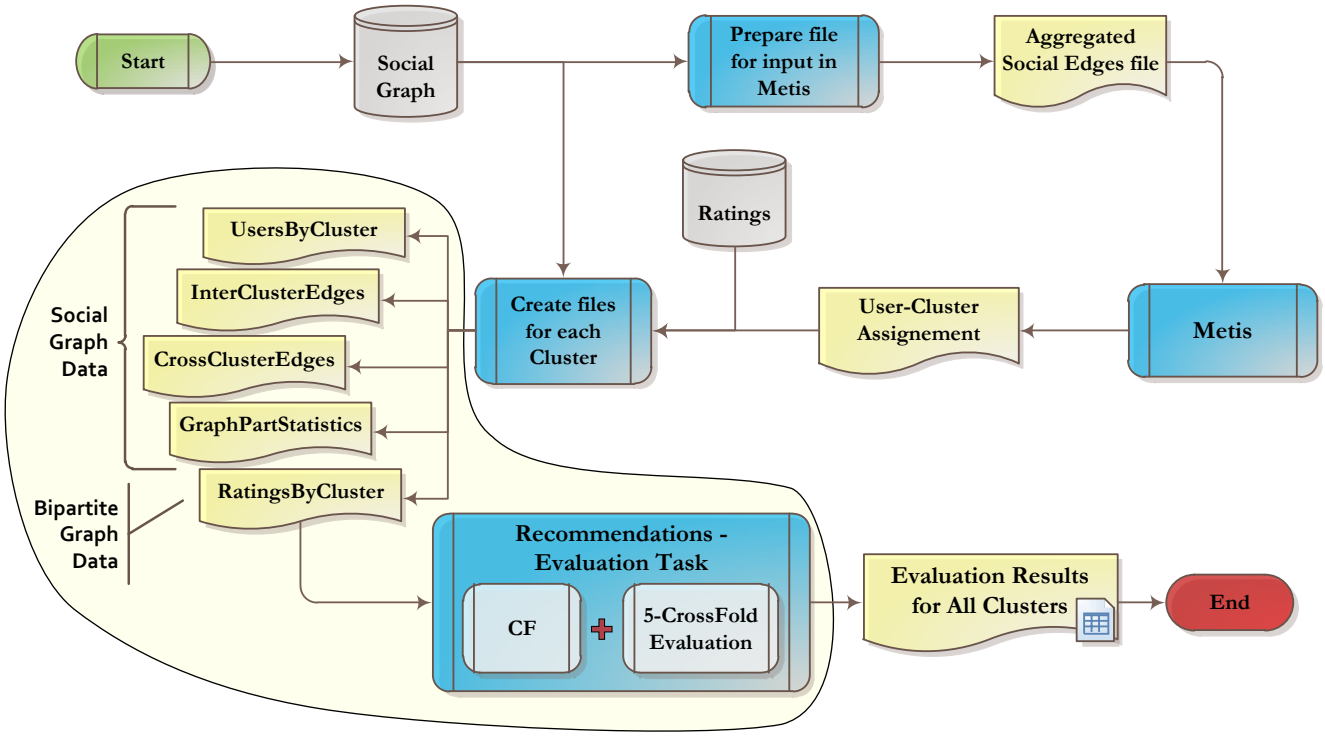
Σύμφωνα με το μοντέλο που έχει υλοποιηθεί, χρησιμοποιήθηκαν τα δεδομένα του κοινωνικού γράφου των χρηστών για την ομαδοποίησή τους σε υπογράφους (subgraphs) και σύμφωνα με την κατάταξή τους σε αυτούς, έγινε και η κατάταξη των βαθμολογιών τους. Έτσι, στη συνέχεια, εκτελέστηκαν οι αλγόριθμοι collaborative filtering πάνω στους υπογράφους για την παραγωγή συστάσεων και την αξιολόγησή τους μέσω του crossfold evaluation. Θα μπορούσαμε αν θέλαμε, να αναπαραστήσουμε αφαιρετικά το διάγραμμα της λειτουργίας του μοντέλου, όπως φαίνεται στο παρακάτω διάγραμμα λειτουργίας.



Εικόνα 4: Αφαιρετικό διάγραμμα λειτουργίας του μοντέλου

3.1 Περιγραφή Συστήματος

Για να επιτευχθούν τα παραπάνω, προτείνεται ένα σύστημα, του οποίου οι κύριες συστατικές μονάδες του συστήματος είναι η μονάδα που είναι υπεύθυνη για την επεξεργασία του κοινωνικού γράφου, η μονάδα που αναλαμβάνει τον διαμοιρασμό του κοινωνικού γράφου σε υπογράφους, η μονάδα επεξεργασίας των υπογράφων και παραγωγής των κατάλληλων αρχείων και τέλος η μονάδα που θα αναλάβει την εκτέλεση των αλγόριθμων Collaborative Filtering και την αξιολόγηση των αποτελεσμάτων από την εκτέλεση των παραπάνω αλγορίθμων. Η περιγραφή των χαρακτηριστικών λειτουργίας του παραπάνω μοντέλου απεικονίζεται στο **SDL** (**S**pecification and **D**escription **L**anguage) διάγραμμα που ακολουθεί. Σε αυτό, απεικονίζονται τα κύρια συστατικά του μοντέλου, η ροή της πληροφορίας κατά την λειτουργία για την παραγωγή συστάσεων καθώς και τα απαραίτητα αρχεία που προκύπτουν ως έξοδος από το σύστημα.



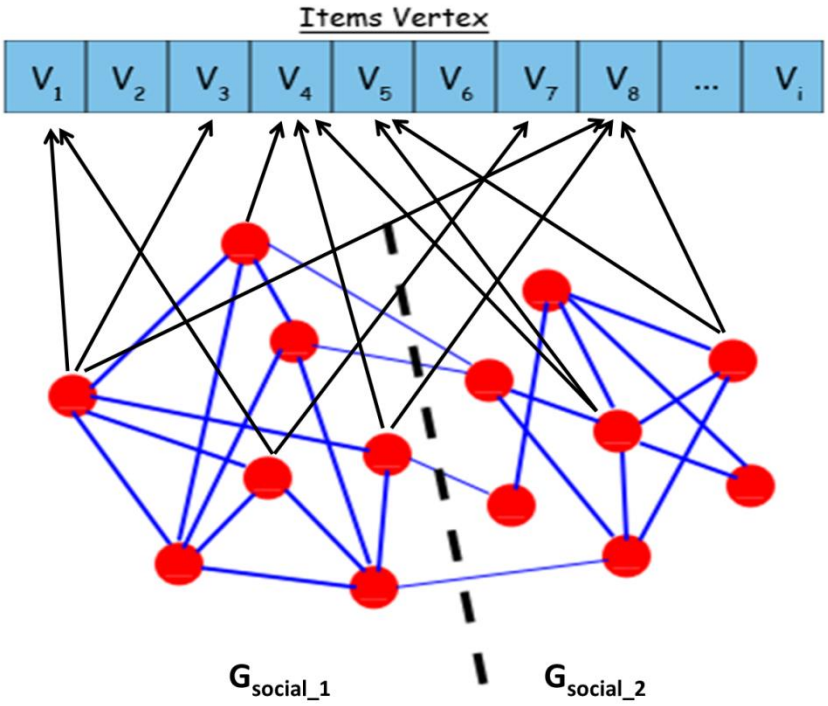
Εικόνα 5: Περιγραφή της λειτουργίας του μοντέλου

Όπως φαίνεται και από το παραπάνω διάγραμμα λειτουργίας, το πρώτο στάδιο αναφέρεται στην επεξεργασία του ενός από τα δύο σύνολα δεδομένων, αυτό του κοινωνικού γράφου (social graph). Έτσι, το αρχείο με τον γράφο που περιλαμβάνει όλες τις ακμές εμπιστοσύνης (φιλίας, δυσπιστίας κλπ) των χρηστών

προς άλλα μέλη του δικτύου διαβάζεται από το σύστημα το οποίο το επεξεργάζεται με σκοπό την δημιουργία ενός αρχείου που περιέχει όλο τον κοινωνικό γράφο, περιέχει όμως ανά γραμμή όλους τους «φίλους» του κάθε χρήστη με τους αντίστοιχους χαρακτηρισμούς εμπιστοσύνης τους. Η μορφή αυτή του αρχείου είναι η κατάλληλη μορφή ώστε να μπορεί να το επεξεργαστεί η επόμενη μονάδα που ακολουθεί και έχει να κάνει με την διχοτόμηση του γράφου σε μικρότερους υπογράφους. Ο αριθμός των υπογράφων που θα προκύψει, ορίζεται από τον χρήστη και επηρεάζει άμεσα το μέγεθος των υπογράφων και κατά συνέπεια το πλήθος των χρηστών που θα περιέχει ο κάθε υπογράφος. Η δεύτερη μονάδα λοιπόν, διαιρεί τον κοινωνικό γράφο σε τόσα τμήματα όσα έχουν οριστεί από τον χρήστη και παράγει ένα αρχείο στο οποίο αντιστοιχεί τον αναγνωριστικό αριθμό (**UserId**) κάθε χρήστη με τον κωδικό του υπογράφου στον οποίο ανήκει (**ClusterId**).

Το αρχείο με τους υπογράφους που έχει παραχθεί από το προηγούμενο στάδιο μαζί με το δεύτερο σύνολο δεδομένων, αυτό με τις βαθμολογίες των χρηστών για τα προϊόντα που έχουν αξιολογήσει (bipartite ratings graph) γίνονται οι είσοδοι της τρίτης φάσης λειτουργίας του μοντέλου. Στο στάδιο αυτό, γίνεται η ομαδοποίηση του συνόλου των χρηστών στους υπογράφους στους οποίους ανήκουν, απ’ όπου προκύπτει το αρχείο **UsersByCluster,** ενώ γίνεται παράλληλα διασταύρωση των ακμών φιλίας των χρηστών ανά subgraph, ώστε να ομαδοποιηθούν και οι ακμές του γράφου σύμφωνα με τα clusterIds στα οποία ανήκουν οι χρήστες τους. Αποτέλεσμα της διαδικασίας αυτής είναι να παραχθούν τα επιμέρους subgraphs στο αρχείο **InterClusterEdges**, ενώ για να θεωρηθεί μία ακμή εσωτερική στο subgraph και να μπει στο subgraph θα πρέπει και οι δύο κόμβοι που ορίζουν την ακμή να ανήκουν στο ίδιο clusterId. Αντίθετα, οι ακμές στις οποίες οι κόμβοι που την ορίζουν δεν ανήκουν στον ίδιο υπογράφο, θεωρούνται εξωτερικές του subgraph και καταχωρούνται στο αρχείο **CrossClusterEdges**. Στην πραγματικότητα, οι κόμβοι που ανήκουν στις crossCluster ακμές του γράφου, είναι «γείτονες» των κόμβων από άλλα subgraphs. Αυτό σημαίνει ότι μπορεί ένας κόμβος να έχει μία ακμή που να δείχνει μια σχέση φιλίας με κάποιον άλλο κόμβο, αλλά ο δεύτερος να μην έχει κανένα ρόλο στην διαμόρφωση σύστασης για τον πρώτο κόμβο, καθώς ανήκουν σε διαφορετικά subgraphs και κατά συνέπεια η ακμή τους

ανήκει στις crossCluster ακμές. Παράλληλα, από το σύστημα δημιουργείται και ένα αρχείο **GraphPartStatistics**, το οποίο περιέχει κάποια στατιστικά σε σχέση με τον νέο διχοτομημένο γράφο, όπως ο μέσος όρος αλλά και ο ελάχιστος και μέγιστος αριθμός κόμβων και ακμών ανά υπογράφο.



Εικόνα 6: Παράδειγμα partitioning του αρχικού κοινωνικού γράφου και των items που έχουν βαθμολογήσει οι χρήστες

Το τελευταίο αρχείο το οποίο προκύπτει από την επεξεργασία του τρίτου σταδίου είναι το αρχείο **RatingsByCluster**, το οποίο δημιουργείται από την διασταύρωση των ακμών αξιολόγησης των χρηστών και των κόμβων που βρίσκονται σε κάθε υπογράφο με σκοπό να ομαδοποιηθούν και οι ακμές των αξιολογήσεων στους αντίστοιχους υπογράφους στους οποίους ανήκουν οι χρήστες αυτών των αξιολογήσεων.

Έτσι, όπως φαίνεται και στο παράδειγμα της Εικόνας 6, ο αρχικός κοινωνικός γράφος G_{social} διαιρείται σε δύο υπογράφους, (G_{social_1} , G_{social_2}). Άρα, στο αρχείο **UsersByCluster** περιέχονται οι χρήστες ομαδοποιημένοι στους δύο υπογράφους, και στο αρχείο **InterClusterEdges** έχουν τοποθετηθεί οι αντίστοιχες ακμές τους που περιέχονται εντός του κάθε υπογράφου. Αντίστοιχα, το αρχείο **CrossClusterEdges** περιλαμβάνει τις ακμές του κοινωνικού γράφου που στην Εικόνα 6 διασχίζονται από

την διαχωριστική γραμμή μεταξύ των δύο υπογράφων. Ενδιαφέρον έχει να παρατηρήσουμε τα περιεχόμενα του αρχείου **RatingsByCluster**.

Παρόλο λοιπόν που όπως φαίνεται πάνω από την Εικόνα 6 έχουν βαθμολογηθεί έξι αντικείμενα, στο αρχείο **RatingsByCluster** τα αντικείμενα V_4 και V_8 θα εμφανίζονται και στους δύο γράφους καθώς όπως παρατηρούμε, έχουν βαθμολογηθεί από χρήστες που ανήκουν σε διαφορετικούς υπογράφους και κατά συνέπεια, μετά την διαίρεση του αρχικού γράφου ο κάθε χρήστης μεταφέρει στον υπογράφο του και το σύνολο των βαθμολογιών που έχει δώσει.

Αν τώρα θεωρήσουμε πιο επίσημα, τον κοινωνικό γράφο G_{social} και τον γράφο αξιολογήσεων $G_{bipartite}$, τέτοιους ώστε:

$$G_{social} = \{V_u, E_s\}, με\ E_s: undirected, unweighted$$

$$G_{bipartite} = \{V_u, V_i, E\}, με\ E: directed, weighted$$

Τότε, μετά το graph partitioning των παραπάνω γράφων, τα αρχεία **UsersByCluster** και **RatingsByCluster** περιέχουν τους i G_{social} και $G_{bipartite}$ υπογράφους αντίστοιχα και οι οποίοι προκύπτουν ως εξής:

$$G_{social_i}: \begin{cases} V_u = \bigcup_{i=1..N} V_{s_i} \\ \forall\, i,j, i \neq j \rightarrow V_{s_i} \cap V_{s_j} = \emptyset \end{cases}$$

$$G_{bipartite_i}: V \in V_u \begin{cases} V \in V_{s_i} \\ e(user, item): e \in E, \rightarrow G_{bipartite_i} = \left\{V_{s_i}, V_{i_{pointed\,V_{si}}}, E\right\} \\ user \in V_{s_i}, item \in V_i \end{cases}$$

Στο αρχείο **RatingsByCluster** λοιπόν περιέχονται ομαδοποιημένες ανά υπογράφο οι ακμές αξιολόγησης, με σκοπό να μπουν πλέον σαν είσοδο, για κάθε subgraph, στο τέταρτο και τελευταίο στάδιο λειτουργίας, στο οποίο βρίσκεται το **Lenskit framework** και το οποίο αναλαμβάνει την εκτέλεση της δημιουργίας και αξιολόγησης των προβλέψεων για κάθε κόμβο (**Recommendation–Evaluation task**). Η εργασία αξιολόγησης εκτελείται για κάθε υπογράφο, για τον οποίο εκτελείται ένα 5-fold cross validation χρησιμοποιώντας έναν αλγόριθμο Collaborative Filtering και

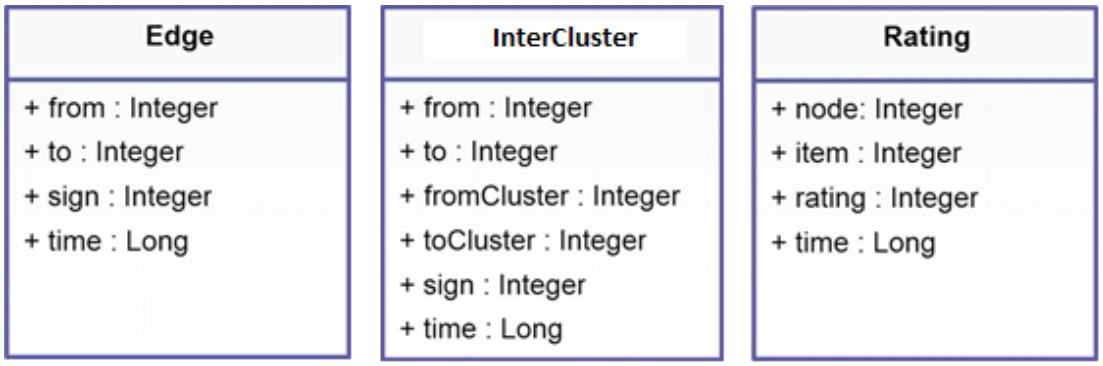
από τα οποία προκύπτουν τα συνολικά αποτελέσματα αξιολόγησης για κάθε υπογράφο. Σχετικά με την εκτέλεση των πειραμάτων για την εργασία αξιολόγησης, έγιναν αρκετές δοκιμές για διάφορα σύνολα παραμέτρων του γράφου, τις οποίες όμως θα περιγράψουμε παρακάτω, μαζί με την περιγραφή της πειραματικής διαδικασίας που ακολουθήθηκε για όλα τα πειράματα που εκτελέστηκαν για το παραπάνω μοντέλο.

3.2 Κλάσεις

Κατά την προσέγγιση του υπό εξέταση μοντέλου, ορίστηκαν τρεις βασικές κλάσεις για το μοντέλο, όπως φαίνονται στο πακέτο gr.hua.dit.basics, οι οποίες και αντιπροσωπεύουν τα τρία κύρια χαρακτηριστικά κάθε συνόλου δεδομένων κοινωνικών δικτύων και δικτύων αξιολόγησης χαρακτηριστικών. Έτσι, η κλάση Edge αντιπροσωπεύει τις ακμές των σχέσεων μεταξύ των χρηστών (κόμβων) του κοινωνικού γράφου, των οποίων όμως οι κόμβοι θα βρίσκονται στον ίδιο υπογράφο, περιλαμβάνοντας κατά συνέπεια τα πεδία **from**, **to** και **sign** τα οποία αντιστοιχούν στα **Από**, **Προς** και **Πρόσημο** αντίστοιχα. Το sign χρησιμοποιείται στην περίπτωση που μιλάμε για γράφο με βάρη τιμών στην ακμή Από→Προς. Στην περίπτωση που οι ακμές του γράφου δεν χαρακτηρίζονται από βάρη, τότε το πεδίο αυτό μπορεί να παραβλεφθεί. Επίσης, όπως φαίνεται και από τον κώδικα της κλάσης, υπάρχει και το πεδίο **time**, το οποίο υπάρχει σαν σχόλιο και αντιστοιχεί στον χρόνο καταγραφής αυτής της σχέσης και το οποίο μπορεί να χρησιμοποιηθεί εφόσον ο κοινωνικός μας γράφος περιέχει αυτήν την πληροφορία. Ωστόσο, στην περίπτωσή μας, αν και αρχικά είχε χρησιμοποιηθεί αυτή η πληροφορία, στη συνέχεια κρίθηκε σκόπιμο για λόγους βελτιστοποίησης της απόδοσης να μην συμπεριληφθεί στην διαδικασία, καθώς λόγου του μεγάλου όγκου δεδομένου του γράφου, το πεδίο του χρόνου αύξανε σημαντικά την πολυπλοκότητα και κατά συνέπεια την υπολογιστική ισχύ που χρειαζόταν για την επεξεργασία του. Τέλος η κλάση Edge κάνει override την μέθοδο toString για να επιστρέφει την ακμή σε μορφή κατάλληλη για το επόμενο στάδιο επεξεργασίας.

Όπως αναφέρθηκε και παραπάνω, η κλάση Edge του πακέτου gr.hua.dit.basics αντιπροσωπεύει τις ακμές των οποίων οι κόμβοι βρίσκονται, μετά την διχοτόμηση του αρχικού γράφου, στον ίδιο υπογράφο. Αντίθετα, η κλάση

InterCluster αντιπροσωπεύει εκείνες τις ακμές του γράφου, των οποίων οι κόμβοι δεν έχουν καταλήξει στον ίδιο υπογράφο μετά το partitioning. Η κλάση αυτή, κληρονομεί την κλάση **Edge**, οπότε περιέχει ομοίως τα πεδία **from**, **to** και **sign** που αντιστοιχούν στον κόμβο αφετηρίας, στον κόμβο προορισμού της σχέσης και το πιθανό πρόσημο της σχέσης, περιέχει ωστόσο και τα πεδία **fromCluster** και **toCluster**. Κατά την επεξεργασία μιας ακμής του γράφου από το σύστημα, γίνεται έλεγχος για το αν ο αριθμός του υπογραφού (**ClusterID**) του κόμβου αφετηρίας είναι ίδιος με τον αριθμό του υπογράφου του κόμβου προορισμού, κατά συνέπεια και αν οι δύο κόμβοι της σχέσης ανήκουν στον ίδιο υπογράφο. Έτσι, το πεδίο fromCluster αντιπροσωπεύει τον αριθμό του υπογράφου του κόμβου αφετηρίας και το πεδίο toCluster τον αριθμό του υπογράφου του κόμβου προορισμού. Και σε αυτήν την περίπτωση έγινε αφαίρεση του πεδίου του χρόνου καθώς δε χρησιμοποιήθηκε στα πειράματα που πραγματοποιήσαμε.



Εικόνα 7: Class Diagrams των βασικών κλάσεων του μοντέλου

Η τελευταία βασική δομική μονάδα που χαρακτηρίζει ένα σύνολο δεδομένων αξιολόγησης είναι οι βαθμολογίες των χρηστών για κάθε ένα από τα διαθέσιμα αντικείμενα. Η τελευταία κλάση λοιπόν από το πακέτο gr.hua.dit.basics είναι η κλάση **Rating** και αντιπροσωπεύει την βαθμολογία ενός συγκεκριμένου χρήστη του δικτύου για ένα συγκεκριμένο προϊόν. Περιλαμβάνει λοιπόν τα πεδία **node**, **item** και **rating** τα οποία αντιστοιχούν στον κόμβο που έχει κάνει μια αξιολόγηση, το αντικείμενο που έχει βαθμολογήσει και τον βαθμό με τον οποίο το έχει αξιολογήσει αντίστοιχα. Για τον λόγο που προαναφέρθηκε, έχει σκοπίμως παραληφθεί το πεδίο του χρόνου **time**.

3.3 Μεθοδολογία

Στη συνέχεια, παρουσιάζεται αναλυτικά ο αλγόριθμος που περιγράφει την μεθοδολογία που ακολουθήθηκε για την διαδικασία του χωρισμού των γράφων σε υπογράφους (partitioning), πάνω στους οποίους στη συνέχεια το μοντέλο εκτελεί την λειτουργία του Collaborative Filtering.

```
PartitionSocialGraph (SG{V,E}, Cross, Subgraphs)
{SsG{V,E}} = Partition(SG)
 Cross = Ø
 For_each e in E
      Vi = e.from
      Vj = e.to
      SsGFrom = GetSubgraphForVertex(Vi , {SsG{V,E}})
      SsGTo = GetSubgraphForVertex(Vj , {SsG{V,E}})
      If (SsGFrom = SsGTo)
        SsGFrom = SsGFrom U {e}
      Else
        InterCluster = CreateInterClusterEdge (e, SsGFrom, SsGTo)
             Cross = Cross U {InterCluster}
 End For_each
 Set Subgraphs = {SsG{V,E}}
End PartitionSocialGraph

PartitionRatingsGraph (BG{Vu,Vi,Er}, RatingSubgraphs)
     {BsG{Vu, Ø, Ø}} = PartitionUsers(Subgraphs)
 For_each e in Er
      Vu'=e.from
      BsG{ Vu',Vi', Er') = GetSubgraphForVertex(Vu', Subgraphs)
      Vi' = Vi' U e.to
      Er' = Er' U {e}
 End For_each
 Set RatingSubgraphs = {BsG{Vu',Vi', Er'}}
End PartitionRatingsGraph
```

Αλγόριθμος Περιγραφής Μεθοδολογικής Προσέγγισης για την Διαδικασία του Partitioning

Η συνάρτηση **Partition** αντιστοιχεί στην κλήση του Metis, απ' όπου προκύπτουν οι υπογράφοι. Η συνάρτηση **GetSubgraphForVertex** επιστρέφει τον υπογράφο στον οποίο ανήκει η κάθε ακμή (χρήστης). Η συνάρτηση **CreateInterClusterEdge** δημιουργεί μία InterCluster ακμή χρησιμοποιώντας ως βάση μια απλή ακμή και τα cluster τα οποία συνδέει. Όπως φαίνεται από τον παραπάνω αλγόριθμο, το πρώτο βήμα είναι να γίνει ο χωρισμός του κοινωνικού γράφου των χρηστών σε υπογράφους (**PartitionSocialGraph**). Στην συνέχεια γίνεται έλεγχος για κάθε ακμή του κοινωνικού γράφου, αν ο υπογράφος του From

της σχέσης είναι ίδιος με αυτόν του To της σχέσης (**GetSubgraphForVertex**). Αν είναι, τότε η σχέση προστίθεται στις InterCluster ακμές, αν δεν είναι τότε η ακμή προστίθεται στο σύνολο των CrossCluster ακμών. Έτσι, μετά από αυτό το βήμα, έχουν χωριστεί πλήρως οι ακμές φιλίας του κοινωνικού γράφου ανάλογα με τους υπογράφους (Subgraphs), έχοντας διαχωρίσει τις ακμές μεταξύ διαφορετικών υπογράφων.

Το επόμενο βήμα, αφορά τον διαχωρισμό των ακμών του γράφου αξιολόγησης (**PartitionRatingsGraph**). Για τον σκοπό αυτό, παίρνουμε τον γράφο αξιολογήσεων και για κάθε ακμή αξιολόγησης ενός αντικειμένου από έναν χρήστη, βρίσκουμε τον υπογράφο στον οποίο ανήκει ο χρήστης (**GetSubgraphForVertex**) και περνάμε την ακμή αξιολόγησης (`e`) στον κάθε υπογράφο. Αφού γίνει η ίδια διαδικασία για όλα τα subgraphs που δημιουργήθηκαν βάση του κοινωνικού γράφου, έχουμε δημιουργήσει τα `RatingSubgraphs`, τα οποία περιέχουν πλέον τις ακμές του bipartite γράφου (`BsG{Vu',Vi', Er'}`) για κάθε subgraph.

3.4 Μετρικές Αξιολόγησης

Όσον αφορά τις μετρικές αξιολόγησης, στη βιβλιογραφία αναφέρονται μια πληθώρα μετρικών που μπορούν να χρησιμοποιηθούν τόσο στατιστικές μετρικές, όσο και μετρικές λήψης αποφάσεων. Στη συνέχεια ακολουθούν οι μετρικές που χρησιμοποιήθηκαν κατά την πειραματική διαδικασία για την αξιολόγηση των αλγόριθμων συστάσεων.

3.4.1 MAE (Mean Absolute Error)

Η μετρική Mean Absolute Error, είναι από τις πιο γνωστές και ευρέως χρησιμοποιούμενη μετρική αξιολόγησης προβλέψεων. Αν υποθέσουμε ότι έχουμε ένα ζεύγος πραγματικής βαθμολογίας – πρόβλεψης $< y_i, \hat{y}_i >$ τότε, το σφάλμα κατά MAE προκύπτει από τον μέσο όρο της απόλυτης διαφοράς μεταξύ της πραγματικής τιμής και της τιμής της πρόβλεψης (δηλαδή του σφάλματος πρόβλεψης) και δίνεται από τον τύπο:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Εδώ πρέπει να επισημανθεί, πως κατά την υλοποίησης του Recommendation-Evaluation task στα πειράματα που έγιναν, η μετρική MAE υπολογίσθηκε πρώτα συνολικά για κάθε βαθμολογία (**MAE**)(**Macro-Average**), ενώ ταυτόχρονα υπολογίσθηκε για κάθε χρήστη (**MAE_ByUser**)(**Micro-Average**).

3.4.2 RMSE (Root Mean Squared Error)

Η μετρική RMSE, προκύπτει από την τετραγωνική ρίζα του αθροίσματος των τετραγώνων των σφαλμάτων πρόβλεψης, όπως δίνεται και από τον ακόλουθο τύπο, ενώ συγκριτικά με την MAE η RMSE τιμωρεί αυστηρά τα μεγάλα σφάλματα στις προβλέψεις, λόγω του υπολογισμού της ρίζας του αθροίσματος τετραγώνων.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Σημειώνεται, ότι κατά την υλοποίηση του evaluation και η μετρική RMSE υπολογίσθηκε τόσο ανά βαθμολογία (**RMSE_ByRating**), όσο και ανά χρήστη (**RMSE_ByUser**).

3.4.3 nDCG (Normalized Discounted Cumulative Gain)

Άλλη μία μετρική η οποία χρησιμοποιήθηκε κατά τα πειράματα είναι η nDCG, η οποία μετράει την κανονικοποιημένη απόδοση του συστήματος συστάσεών μας με βάση την βαθμολογημένη κατά σχετικότητα λίστα των συστάσεων. Το εύρος τιμών της είναι [0.0,1.0], με την ιδανικότερη βαθμολογία ενός στοιχείου που ελαχιστοποιεί το σφάλμα να είναι το 1.0. Πολύ συχνή χρήση της μετρικής γίνεται στις περιπτώσεις ανάκτησης πληροφορίας και στην αξιολόγηση των μηχανών αναζήτησης ιστού. Ο υπολογισμός του nDCG δίνεται από τον παρακάτω τύπο, όπου k είναι το πλήθος των συστάσεων, rel_i είναι η ομοιότητα μεταξύ της πραγματικής και της προβλεπόμενης βαθμολογίας για την πρόταση i και $IDCG_k$ είναι το ιδανικό-μέγιστο DCG.

$$DCG_k = \sum_{i=1}^{k} \frac{2^{rel_i}-1}{log_2(i+1)} \quad , \quad nDCG_k = \frac{DCG_k}{IDCG_k}$$

3.4.4 Top-N nDCG (Normalized Discounted Cumulative Gain for Top-N items)

Όλες οι παραπάνω μετρικές, είναι μετρικές ακρίβειας προβλέψεων, καθώς αξιολογούν την ακρίβεια του συστήματος πρόβλεψης βαθμολογιών, είτε για την κατάταξη των αντικειμένων σε λίστα, είτε για την πρόβλεψη της βαθμολογίας ενός χρήστη για ένα αντικείμενο. Η μετρική top-N nDCG, είναι αντίστοιχη της nDCG με τη διαφορά όμως ότι ανήκει στην κατηγορία των μετρικών που εφαρμόζονται σε λίστες από συστάσεις, οι οποίες θα πρέπει να οριστούν πριν την εκτέλεση των μετρήσεων. Για την ρύθμιση της μετρικής top-N nDCG, έπρεπε να γίνει κάποια παραμετροποίηση για την εκτέλεση του Recommendation-Evaluation. Οι παράμετροι αυτές είναι οι εξής:

- ✓ Ο αριθμός της λίστας των συστάσεων στην οποία θα εφαρμοστεί. Στα πειράματά μας το listsize (αντιστοιχεί στο N του Top-N) είχε οριστεί ίσο με πέντε, καθώς μετά από αρκετές δοκιμές, διαπιστώθηκε ότι πάνω από πέντε αντικείμενα στη λίστα η επιρροή στα αποτελέσματα ήταν σχεδόν μηδενική, οπότε για λόγους βελτιστοποίησης καταλήξαμε στα πέντε αντικείμενα.
- ✓ Η αναλογία στην οποία θα διαιρεθεί το αρχικό σύνολο δεδομένων σε δεδομένα εκπαίδευσης και ελέγχου.

4 Υλοποίηση & Αξιολόγηση του μοντέλου

4.1 Σύνολα Δεδομένων & Πλατφόρμα Πειραμάτων

Για την αξιολόγηση της προτεινόμενης προσέγγισης χρησιμοποιήθηκαν δύο, διαθέσιμα από τα Πανεπιστήμια του Koblenz–Landau[5] (Oct. 2013) και Simon Fraser[6], σύνολα δεδομένων, τα οποία και τα δύο περιείχαν τόσο πληροφορία κοινωνικού δικτύου μεταξύ των χρηστών, όσο και την απαραίτητη πληροφορία αξιολογήσεων αντικειμένων. Το πρώτο σύνολο δεδομένων αφορούσε το Epinions[7], τον μεγαλύτερο ιστότοπο κριτικών καταναλωτών για προϊόντα και το δεύτερο το Flixster[8], έναν από τους μεγαλύτερους ηλεκτρονικούς χώρους κριτικών ταινιών. Ο κύριος όγκος των πειραμάτων εκτελέστηκε στο σύνολο δεδομένων του Epinions, το οποίο περιελάμβανε 131.828 μοναδικούς χρήστες, 841.372 κοινωνικές σχέσεις (εμπιστοσύνης ή μη) μεταξύ τους και 13.668.320 αξιολογήσεις για 876.252 προϊόντα. Συνοπτικά, τα χαρακτηριστικά των συνόλων δεδομένων που χρησιμοποιήθηκαν φαίνονται στον Πίνακα 2.

[5] http://konect.uni-koblenz.de/networks/
[6] http://www.sfu.ca/ , http://www.cs.sfu.ca/~sja25/personal/datasets/
[7] http://www.epinions.com/
[8] http://www.flixster.com/

		Dataset	
Χαρακτηριστικά		**Epinions**	**Flixster**
Κοινωνικός Γράφος	**Πλήθος Μοναδικών Χρηστών (Distinct Users)**	131.828	786.936
	Πλήθος Ακμών Φιλίας (Social Edges)	841.372	7.058.819
	Μέσος Όρος Ακμών Φιλίας ανά Χρήστη (Average Degree)	12,765	17,94
Γράφος Αξιολογήσεων	**Πλήθος Μοναδικών Χρηστών που Βαθμολογούν (Distinct Users)**	120.492	147.612
	Πλήθος Μοναδικών Αντικειμένων (Distinct Items)	755.760	48.794
	Πλήθος Βαθμολογιών (Ratings)	13.668.320	8.196.077
	Μέσος Όρος Πλήθους Βαθμολογιών ανά Χρήστη (Avg. outDegree/User)	113,44	10,42
	Μέσος Όρος Πλήθους Βαθμολογιών ανά Προϊόν (Avg. inDegree/Item)	18,09	167,97
	Κλίμακα αξιολογήσεων	1 – 5	0,5 – 5
	Ακρίβεια/Βήμα	1.0	0.5
Χαρακτηριστικά Κοινωνικού Γράφου			
Τύπος Κορυφών (Vertex type)		User, User	User, User
Τύπος Ακμών (Edge type)		Trust/Distrust	Trust
Τύπος (Format)		Directed	Directed
Βάρος Ακμών (Edge Weights)		Signed	–
Metadata		Timestamp	–
Χαρακτηριστικά Γράφου Αξιολογήσεων			
Τύπος Κορυφών (Vertex type)		User, Product	User, Product
Τύπος Ακμών (Edge type)		Ratings	Ratings
Τύπος (Format)		Bipartite	Bipartite
Βάρος Ακμών (Edge Weights)		Ratings	Ratings
Metadata		Timestamp	–

Πίνακας 2: Χαρακτηριστικά των συνόλων δεδομένων

Όπως φαίνεται και από τον παραπάνω πίνακα, τα δύο σύνολα δεδομένων είναι εκ διαμέτρου αντίθετα σε δομή μεταξύ τους, καθώς το σύνολο του Epinions διαθέτει 131.000 χρήστες σε αντίθεση με αυτό του Flixster που περιλαμβάνει 787.000 και κατά συνέπεια αυτό φαίνεται και στο πλήθος των ακμών φιλίας που περιλαμβάνουν οι δύο γράφοι. Θα πρέπει εδώ να παρατηρήσουμε ότι ένα μεγάλο ποσοστό των χρηστών του Flixster εμφανίζεται στον κοινωνικό γράφο, ωστόσο δεν έχει δώσει καμία αξιολόγηση και κατά συνέπεια δεν συμπεριλαμβάνεται στην συνέχεια κατά την διαδικασία του evaluation. Αυτοί οι χρήστες αντιμετωπίζουν σύμφωνα με την βιβλιογραφία το πρόβλημα του Cold-Start και αλγόριθμοι που βασίζονται στο ιστορικό αξιολογήσεων των χρηστών δεν μπορούν να τους παράγουν προτάσεις. Επίσης, αυτό που παρατηρείται από τα χαρακτηριστικά των δυο γράφων είναι ότι οι γράφοι αξιολόγησης αν και περιλαμβάνουν συγκρίσιμο αριθμό χρηστών που έχουν δώσει αξιολογήσεις, ο γράφος του Epinions περιέχει πολλά περισσότερα προϊόντα σε σχέση με τον γράφο του Flixster. Επακόλουθο αυτού είναι ο γράφος του Flixster να έχει πολύ μεγαλύτερο μέσο πλήθος αξιολογήσεων ανά προϊόν (Avg. inDegree/Item), σε σχέση με τον γράφο του Epinions, παρόλο που ο τελευταίος περιλαμβάνει πολλαπλάσιο συνολικό αριθμό βαθμολογιών.

4.2 Υλοποίηση

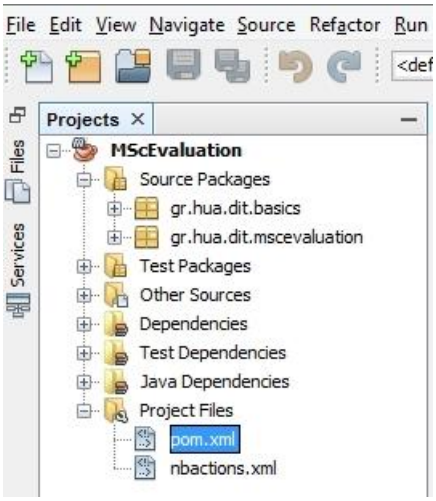
Σε προηγούμενη ενότητα, έγινε περιγραφή των δομικών μονάδων και των λειτουργιών του μοντέλου το οποίο προτείνεται στο πλαίσιο της παρούσας διπλωματικής μελέτης. Αξίζει ωστόσο να αναλύσουμε πιο συγκεκριμένα τον τρόπο λειτουργίας της κάθε οντότητας του συστήματος για να καταλάβουμε την ακριβή λειτουργία του και τα χαρακτηριστικά του.

Η υλοποίηση του συστήματος στο σύνολό του έγινε σε ένα Maven project το οποίο βασίστηκε πάνω στην δομή του LensKit Recommender framework, ενώ υιοθετήθηκε η έννοια του project object model (**POM**) για την διαχείριση της δομής των περιεχομένων του project. Τα κύρια χαρακτηριστικά που κάνουν το Apache Maven Project ιδιαίτερα ευέλικτο και εύχρηστο είναι τα εξής:

- Απλό στη ρύθμισή του και ακολουθεί τις βέλτιστες πρακτικές οργάνωσης και λειτουργίας.
- Ανώτερη διαχείριση των εξαρτήσεων (dependencies), συμπεριλαμβανομένων των αυτόματων ενημερώσεων.
- Ευελιξία στην παράλληλη εργασία με διάφορα projects.
- Μια μεγάλη και διαρκώς αυξανόμενη αποθήκη βιβλιοθηκών και μετα-δεδομένων.
- Είναι επεκτάσιμο, καθώς είναι εύκολο ο χρήστης να δημιουργήσει τα δικά του plugins σε Java ή άλλες scripting γλώσσες.

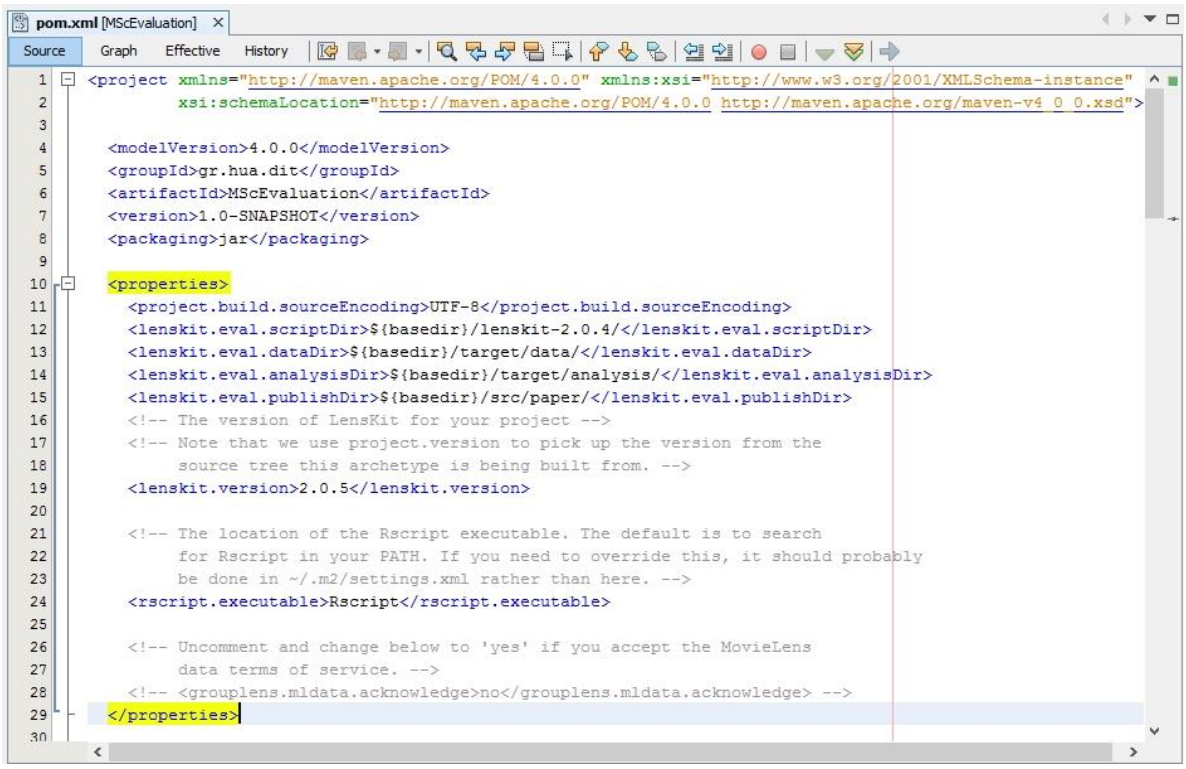
Δημιουργήθηκε λοιπόν ένα Maven Project from Archetype, χρησιμοποιώντας πιο συγκεκριμένα την έκδοση 2.0.5 του lenskit-archetype-fancy-analysis, με GroupID: org.grouplens.lenskit από το Maven repository http://repo.maven.apache.org/maven2/. Προτείνεται η χρήση κάποιου από τα γνωστά ολοκληρωμένα περιβάλλοντα διαχείρισης για την ανάπτυξη κώδικα (**IDE**) που διαθέτουν ενσωματωμένη υποστήριξη για Maven, καθώς δημιουργώντας το project με τον τρόπο που περιγράφεται παραπάνω, το IDE αναλαμβάνει να κατεβάσει και να αρχειοθετήσει στον φάκελο Dependencies, σύμφωνα με την κατά Maven αρχιτεκτονική διαχείρισης φακέλων, όλα τα εξαρτώμενα jar αρχεία που θα χρειαστούν για την ανάπτυξη του κώδικα για την δημιουργία του LensKit Recommender και LensKit Evaluator που θα περιγράψουμε παρακάτω. Εναλλακτικά, θα πρέπει από κονσόλα εντολών να εγκατασταθεί η υποστήριξη για το Maven στο σύστημα, να γίνει αναζήτηση στα Maven αποθετήρια και εγκατάσταση του ανάλογου archetype και να δημιουργηθεί το project με βάση αυτό.

Το project που δημιουργήθηκε, περιλαμβάνει, εκτός των άλλων, δύο πακέτα (**packages**) μέσα στα οποία είναι ομαδοποιημένα τα αρχεία του κώδικα που θα εξηγήσουμε στην συνέχεια και δύο αρχεία xml μέσα στον φάκελο Project Files, τα **pom.xml** και **nbactions.xml**, τα οποία περιέχουν τις ρυθμίσεις σχετικά με τα properties για τα dependencies και plugins τα οποία είναι απαραίτητα για να τρέχει ο LensKit Recommender και Evaluator και τις ρυθμίσεις του maven project στο σύνολό τους αντίστοιχα.



Εικόνα 8: Δομή φακέλων του project

Χαρακτηριστικά, μπορούμε να αναφέρουμε από το pom.xml κάποια properties που χρειάζονται προσοχή για να είναι δυνατή η εκτέλεση του LenKit.



Εικόνα 9: Περιεχόμενα properties στο pom.xml

Τα πιο σημαντικά από τα πεδία που περιλαμβάνει το αρχείο pom.xml είναι τα εξής:

```
<lenskit.eval.scriptDir>${basedir}/lenskit-2.0.4/</lenskit.eval.scriptDir>
<lenskit.eval.dataDir>${basedir}/target/data/</lenskit.eval.dataDir>
<lenskit.eval.analysisDir>${basedir}/target/analysis/</lenskit.eval.analysisDir>
<lenskit.eval.publishDir> ${basedir}/src/paper/</lenskit.eval.publishDir>
```

Για να εκτελεστούν ο LensKit Recommender και ο LensKit Evaluator πρέπει να κληθούν μέσω ενός script σε γλώσσα Groovy[9], στο οποίο περιέχεται ο κώδικας τον οποίο θα πρέπει να εκτελέσουν οι εργασίες για το Recommendation και το Evaluation. Το πεδίο ***<lenskit.eval.scriptDir>*** λοιπόν, ορίζει τον φάκελο μέσα στον οποίο θα γίνει αναζήτηση του **eval.groovy** αρχείου για την εκτέλεση του εργαλείου. Το πεδίο ***<lenskit.eval.dataDir>*** ορίζει τον φάκελο μέσα από τον οποίο έχει οριστεί στο framework να γίνεται η αλληλεπίδραση με τα αρχεία δεδομένων, ωστόσο αυτό δεν είναι απόλυτα δεσμευτικό, καθώς όπως θα δούμε και παρακάτω μπορούμε να παρακάμψουμε αυτή τη ρύθμιση σε περίπτωση που χρειάζεται να αλλάζουν τα αρχεία δεδομένων, κάτι που έγινε και στα πειράματα που εκτελέστηκαν για αυτό το μοντέλο. Τα πεδία ***<lenskit.eval.analysisDir>*** και ***<lenskit.eval.publishDir>*** τέλος, αφορούν τους φακέλους αποθήκευσης των αποτελεσμάτων και των γραφημάτων που μπορούν να προκύψουν από την ανάλυση, κάτι το οποίο και αυτό μπορεί να παρακαμφθεί έμμεσα στην περίπτωση που εκτελούνται τμηματικές αναλύσεις, όπως στα πειράματα που εκτελέστηκαν, και θέλουμε να παραμετροποιήσουμε τα αρχεία εξόδου για να κρατήσουμε όλα τα αποτελέσματα της ανάλυσης.

Στο pom.xml περιέχονται ακόμα οι ρυθμίσεις των εξαρτήσεων (dependencies) τις οποίες χρειάζεται το LensKit Recommender Toolkit για να εκτελεστεί και οι οποίες έχουν κατέβει αυτόματα στο φάκελο Dependencies κατά τη δημιουργία του project. Αυτές περιλαμβάνουν όλα τα πακέτα με τις κλάσεις οι οποίες χρησιμοποιούνται από τον Predictor, καθώς και για τα tests από τον Evaluator.

[9] http://groovy.codehaus.org/

```
<dependencies>
  <dependency>
    <groupId>org.grouplens.lenskit</groupId>
    <artifactId>lenskit-core</artifactId>
    <version>${lenskit.version}</version>
  </dependency>
  <dependency>
    <groupId>org.grouplens.lenskit</groupId>
    <artifactId>lenskit-knn</artifactId>
    <version>${lenskit.version}</version>
  </dependency>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.11</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>org.hamcrest</groupId>
    <artifactId>hamcrest-library</artifactId>
    <version>1.3</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>org.grouplens.lenskit</groupId>
    <artifactId>lenskit-eval</artifactId>
    <version>2.0.5</version>
    <type>jar</type>
  </dependency>
</dependencies>
```

Εικόνα 10: Λίστα εξαρτήσεων στο pom.xml

Επίσης, το αρχείο pom.xml όπως αναφέρθηκε, περιλαμβάνει και τις ρυθμίσεις για τα plugins που χρησιμοποιεί το LensKit Recommender Toolkit και στα οποία συγκαταλέγονται το maven-compiler-plugin από την Apache και με το οποίο γίνεται compile το LensKit Recommender Toolkit,

```
<!-- LensKit requires at least Java 1.6 -->
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-compiler-plugin</artifactId>
  <version>2.0.2</version>
  <configuration>
    <source>1.7</source>
    <target>1.7</target>
  </configuration>
</plugin>
```

Εικόνα 11: Compiler plugin στο pom.xml

αλλά και το lensKit evaluation plugin, το οποίο είναι αυτό που ορίζει την εκτέλεση του recommendations – evaluation task για την σύγκριση των αλγόριθμων πρόβλεψης για τις διάφορες μετρικές. Συγκεκριμένα, το πεδίο ***<script>*** μέσα στο πεδίο ***<configuration>*** ορίζει κατά την εκτέλεση την αξιολόγησης από το σύστημα, θα εκτελεστεί ο κώδικας που βρίσκεται μέσα στο eval.groovy script. Επίσης, το πεδίο **<analysisScript>** ορίζει ένα script για το λογισμικό R[10], το οποίο θα εκτελεστεί

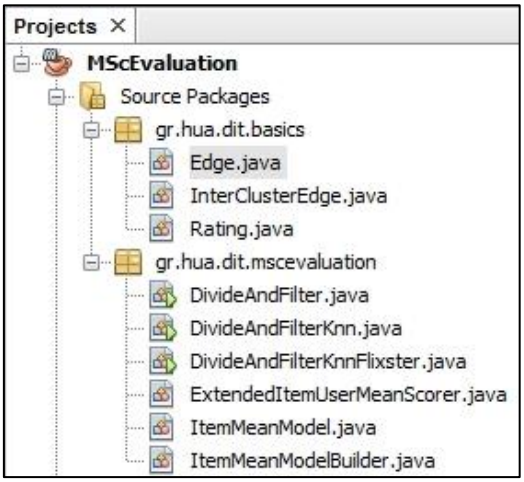
[10] http://www.r-project.org/

σε περίπτωση που θέλουμε να εκμεταλλευτούμε την δυνατότητα του LensKit Recommender Toolkit για παραγωγή και γραφημάτων κατά το στάδιο εκτέλεσης του evaluation.

```
<plugin>
  <artifactId>lenskit-eval-maven-plugin</artifactId>
  <groupId>org.grouplens.lenskit</groupId>
  <version>${lenskit.version}</version>
  <extensions>true</extensions>
  <configuration>
    <script>${lenskit.eval.scriptDir}/eval.groovy</script>
    <analysisScript>${lenskit.eval.scriptDir}/chart.R</analysisScript>
    <targets>
      <target>draw</target>
      <target>analyze</target>
    </targets>
  </configuration>
</plugin>
```

Εικόνα 12: Evaluation plugin στο pom.xml

Όπως φαίνεται στη δομή των φακέλων του project, περιέχονται και δύο πακέτα (**packages**) μέσα στα οποία είναι οργανωμένα όλα τα java αρχεία που περιέχουν των πηγαίο κώδικα (**source code**) που αναπτύχθηκε στα πλαίσια αυτής της έρευνας για την δημιουργία του μοντέλου που περιγράφουν οι ενότητες 3.1 & 3.2. Στο πακέτο **gr.hua.dit.basics** περιλαμβάνονται τα αρχεία των βασικών κλάσεων που χρησιμοποιούνται στην κύρια εκτέλεση του κώδικα και τα οποία περιγράφονται στη συνέχεια, ενώ στο πακέτο **gr.hua.dit.mscevaluation** βρίσκονται τα αρχεία με τον κώδικα που υλοποιεί την λειτουργία όλου του συστήματος, από το στάδιο της προετοιμασίας των αρχείων για είσοδο στο σύστημα, μέχρι την ολοκλήρωση της διαδικασίας αξιολόγησης και την έξοδο των αποτελεσμάτων από το σύστημα στο CSV αρχείο (**Global Evaluation Results**).



Εικόνα 13: Source Code Packages

Στο δεύτερο πακέτο πηγαίου κώδικα java (gr.hua.dit.mscevaluation) περιέχονται καταρχήν τρία αρχεία (ExtendedItemUserMeanScorer, ItemMeanModel, ItemMeanModelBuilder) με java κλάσεις που περιέχονται από προεπιλογή με την δημιουργία του project από το LensKit archetype και το οποίο είναι ένα παράδειγμα μιας πρώτης υλοποίησης ενός ItemScorer σε java. Η υλοποίηση του μοντέλου που προτείνεται στις ενότητες 3.1 & 3.2 ξεκίνησε από το αρχείο **DivideAndFilter**.java, το οποίο περιλαμβάνει όλο τον κώδικα υλοποίησης της διαδικασίας όπως παρουσιάζεται στη συνέχεια.

Έμμεσος στόχος της υλοποίησης ήταν η προσπάθεια δημιουργίας ενός αυτοματοποιημένου τρόπου εκτέλεσης της διαδικασίας λήψης και αξιολόγησης των συστάσεων, εκτελώντας απλά ο χρήστης ένα εκτελέσιμο αρχείο (batch file) και περνώντας σαν ορίσματα τις παραμέτρους εκτέλεσης όπως ο αριθμός των υπογράφων στον οποίο θα γινόταν ο διαχωρισμός του πλήρη γράφου. Μέσα στην κλάση DivideAndFilter όμως, ορίζουμε στατικά τον προεπιλεγμένο αριθμό των υπογράφων, **numOfClusters = 1000**, για την περίπτωση που κατά την εκτέλεση από τη γραμμή εντολών ο χρήστης δεν δώσει ως όρισμα τον αριθμό των υπογράφων, οπότε και από τον έλεγχο που ακολουθεί στην συνέχεια θα οριζόταν η νέα τιμή του numOfClusters.

```
public static void main(String[] args {
      if (args.length > 0 && args[0] != null) {
            numOfClusters = new Integer(args[0]).intValue();
      }
```

Στην συνέχεια, γίνεται η κλήση των τριών μεθόδων οι οποίες αναλαμβάνουν την εκτέλεση των τριών εργασιών που περιγράφονται σε προηγούμενη ενότητα, πριν φτάσει το σύστημα να εκτελέσει την διαδικασία αξιολόγησης.

```
// Δημιουργία του αρχείου με το σωστό format για το gpmetis
createEdgesFile();

// Κλήση του gpmetis
gpmetis();

// Δημιουργία των αρχείων για τους υπογράφους
createClusterFiles();
```

Πιο συγκεκριμένα, η μέθοδος ***createEdgesFile*** αναλαμβάνει την επεξεργασία του γράφου και την προετοιμασία του με μορφή κατάλληλη για να μπορέσει να επεξεργαστεί από το metis. Να διευκρινιστεί ότι καθώς ο Metis υποστηρίζει μη κατευθυνόμενους γράφους με θετικά βάρη, αποφασίσθηκε όλες οι κατευθυνόμενες ακμές του κοινωνικού γράφου (είτε θετικές, είτε αρνητικές) να προστεθούν ως θετικές μη κατευθυνόμενες. Αρχικά, διαβάζει μία –μία τις γραμμές από τα αρχείο με τον γράφο και χωρίζει τα πεδία from και to, κρατώντας ένα αρχείο με την μεταξύ τους σχέση. Στόχος είναι να μαζέψει για κάθε κόμβο όλες τις ακμές του στο γράφο, αλλά παράλληλα, εφόσον ο γράφος είναι μη κατευθυνόμενος, να δημιουργήσει και τις αντίστοιχες ακμές για τους κόμβους που βρήκε να έχουν μία σχέση ως κόμβος προορισμού, αντιστρέφοντας την ίδια σχέση και αναπαριστώντας τον κόμβο ως κόμβο αφετηρίας. Από την παραπάνω διαδικασία προκύπτει το αρχείο AggregatedSocialEdges που περιέχει ανά γραμμή όλες τις ακμές του κάθε κόμβου από τον γράφο, μαζί με το βάρος της κάθε φιλίας.

Στην συνέχεια, πρέπει το αρχείο που παράχθηκε από το παραπάνω βήμα να μπει ως είσοδο στον Metis Graph Partitioner. Για αυτό το σκοπό καλείται η μέθοδος ***gpmetis***, η οποία αρχικά δημιουργεί ένα νέο process και έπειτα εκτελεί ένα bash shell με το οποίο καλείται το metis που βρίσκεται στον φάκελο exe του project, ορίζοντας στην ίδια γραμμή εντολής το όνομα του αρχείου που περιέχει τον προς διχοτόμηση γράφο (αρχείο AggregatedSocialEdges) και στη συνέχεια τον αριθμό των υπογράφων που θέλουμε να δημιουργηθούν. Η έξοδος της διαδικασίας είναι ένα αρχείο με το ClusterId στο οποίο έχει αντιστοιχηθεί ο κάθε κόμβος ανά γραμμή.

```
public static void gpmetis() throws InterruptedException {
   try {
     String directoryPath = "/home/user/NetBeansProjects/MScEvaluation/";
     String fileName = directoryPath + "outputFiles/aggregatedEdges5b.txt";
       ProcessBuilder pb = new ProcessBuilder();
```

```
        pb.directory(new File(directoryPath));
        pb.command("bash", "-c", "exe/gpmetis " +fileName+" "+numOfClusters);
        Process process = pb.start();
        process.waitFor();
    }
```

Έπειτα, η επόμενη κλάση στη σειρά που καλείται είναι η ***createClusterFiles***. Αυτή δέχεται ως είσοδο το αρχείο εξόδου του προηγούμενου σταδίου και το αρχείο με το σύνολο δεδομένων των βαθμολογιών των χρηστών για τα διάφορα αντικείμενα. Αρχικά, γίνεται το διάβασμα των κόμβων και των αντίστοιχων IDs των υπογράφων τους και τους τοποθετεί ως <Key, Values> ζεύγη σε ένα TreeMap. Έτσι, ομαδοποιούνται όλοι οι κόμβοι του γράφου ανά υπογράφο στον οποίο ανήκουν, παίρνοντας την πρώτη έξοδο του σταδίου που έχει να κάνει με τους πλήρεις υπογράφους και τους χρήστες τους οποίους περιλαμβάνουν, δηλαδή το αρχείο **usersByCluster**.

```
// Αποθήκευση του nodesByCluster
for (int clid = 0; clid < nodesByCluster.size(); clid++) {
    if (nodesByCluster.get(clid).size() < minNodes) {
        minNodes = nodesByCluster.get(clid).size();
     }
     if (nodesByCluster.get(clid).size() > maxNodes) {
        maxNodes = nodesByCluster.get(clid).size();
     }
     averageNodes = averageNodes + nodesByCluster.get(clid).size();
     writeNodesByCluster.println(clid+";"+nodesByCluster.get(clid));
}
```

Έχοντας ομαδοποιήσει τους κόμβους ανά υπογράφο στον οποίο ανήκουν, γίνεται ανάγνωση της κάθε γραμμής των σχέσεων φιλίας του αρχικού αρχείου με τον κοινωνικό γράφο και γίνεται έλεγχος αν και τα δύο μέλη της σχέσης ανήκουν στον ίδιο υπογράφο. Αν **Ναι**, τότε η σχέση μεταφέρεται στο edgesByCluster TreeMap, ως τιμή του κλειδιού ClusterId. Αν **Όχι**, τότε δημιουργείται ένα νέο αντικείμενο της βασικής κλάσης InterClusterEdge και η σχέση προστίθεται ως τιμή στο αντίστοιχο κλειδί του crossClusterEdges TreeMap. Από αυτή την διαδικασία προκύπτουν άλλα δύο έγγραφα εξόδου, το **interClusterEdges** με τις εσωτερικές ακμές για κάθε υπογράφο και το **crossClusterEdges** με τις εξωτερικές σχέσεις των υπογράφων. Εδώ να σημειωθεί ότι για λόγους παραμετροποίησης της διαδικασίας, στο όνομα των αρχείων προστίθεται ως επίθεμα και το πλήθος των υπογράφων που έχουμε ορίσει (πχ. nodesByCluster_1000.txt). Επίσης, για τον ίδιο λόγο

δημιουργείται και ο αντίστοιχος φάκελος για κάθε αριθμό πλήθους υπογράφων μέσα στον οποίο αποθηκεύονται όλες οι έξοδοι αυτού του σταδίου.

```
int edgeCounter = 1;
while ((originEdgesLine = originFile.readLine()) != null) {
   fields = originEdgesLine.split(" ");
   from[edgeCounter - 1] = Integer.parseInt(fields[0]);
   to[edgeCounter - 1] = Integer.parseInt(fields[1]);
   sign[edgeCounter - 1] = Integer.parseInt(fields[2]);
   // Βρίσκουμε το clusterId για κάθε From & To κόμβο από το αρχικό αρχείο
   int fromCluster = clusterForNode[from[edgeCounter - 1] - 1];
   int toCluster = clusterForNode[to[edgeCounter - 1] - 1];
   Edge edge = new Edge();
   edge.from = from[edgeCounter - 1];
   edge.to = to[edgeCounter - 1];
   edge.sign = sign[edgeCounter - 1];
   // Έλεγχος αν τα From & To βρίσκονται στο ίδιο cluster
   if (nodesByCluster.get(fromCluster).contains(to[edgeCounter - 1])) {
       if (!edgesByCluster.containsKey(fromCluster)) {
           edgesByCluster.put(fromCluster, new ArrayList<Edge>());
           edgesByCluster.get(fromCluster).add(edge);
           } else {
           edgesByCluster.get(fromCluster).add(edge);
           }
       } // Αν ΟΧΙ τότε προσθέτω την ακμή στο crossClusterEdges Array
       else {
             InterClusterEdge interEdge = new InterClusterEdge();
             interEdge.from = from[edgeCounter - 1];
             interEdge.to = to[edgeCounter - 1];
             interEdge.fromCluster = fromCluster;
             interEdge.toCluster = toCluster;
             interEdge.sign = sign[edgeCounter - 1];
             crossClusterEdges.add(interEdge);
             }
       edgeCounter++;
       }
```

Στη συνέχεια, γίνεται ανάγνωση του αρχείου με τις βαθμολογίες των χρηστών, γίνεται ανάκτηση του αριθμού του υπογράφου στον οποίο ανήκει ο κάθε χρήστης για κάθε βαθμολογία και προστίθεται η βαθμολογία με τη μορφή τριπλέτας (user, item, rating) σε ένα TreeMap από το οποίο προκύπτει στο τέλος αυτής της διαδικασίας το αρχείο **ratingsByCluster**, το οποίο περιλαμβάνει ομαδοποιημένες ανά υπογράφο όλες τις αξιολογήσεις των χρηστών. Τέλος, από αυτό το στάδιο του μοντέλου προκύπτει το αρχείο **graphPartStatistics**, το οποίο περιέχει κάποια χρήσιμα στατιστικά αποτελέσματα σχετικά με το μοίρασμα των χρηστών του γράφου στους υπογράφους τους.

Το επόμενο στάδιο επεξεργασίας αφορά το Recommendations - Evaluation Task και είναι το στάδιο το οποίο θα αξιολογήσει τους προς εκτέλεση αλγόριθμους πρόβλεψης και από το οποίο θα παραχθούν τα τελικά αποτελέσματα αξιολόγησης. Για να γίνει αυτό, το αρχείο με τις βαθμολογίες ανά υπογράφο μπαίνει ως είσοδο

στο σύστημα, το οποίο καλεί την μέθοδο ***evalTempCluster***, η οποία δημιουργεί ένα bash process και εκτελεί το lenskit-eval που βρίσκεται μέσα στον υποφάκελο bin του φακέλου lenskit-2.0.4. Το lenskit-eval με τη σειρά του, καλεί το eval.groovy script, το οποίο όπως έχει αναφερθεί περιέχει τις ρυθμίσεις των αλγόριθμων και του 5-fold Cross Evaluation task που θα εκτελεστούν.

```
public static void evalTempCluster(int numOfClusters, int clID){
   String directoryPath="/.../MScEvaluation/lenskit-2.0.4";
   ProcessBuilder pb = new ProcessBuilder();
   pb.directory(new File(directoryPath));
   pb.command("bash", "-c", "bin/lenskit-eval");
   Process process = pb.start();
   process.waitFor();
```

Στο περιεχόμενο του eval.groovy έχουμε ορίσει αρχικά το crossfold evaluation task, στο οποίο καθορίζουμε το αρχείο στο οποίο θα εκτελεστεί (source CSVfile), το διαχωριστικό των πεδίων που χρησιμοποιείται στο αρχείο, το εύρος της κλίμακας των βαθμολογιών και το βήμα της κλίμακας, καθώς επίσης ως partitions ορίζουμε τον αριθμό των folds που θα εκτελεστούν (στην περίπτωσή μας, πέντε).

```
dataset crossfold("evaluation") {
   source
csvfile("/.../MScEvaluation/outputFiles/clustersTempFile/tempRatings.txt") {
            delimiter ","
            domain {
                minimum 1.0
                maximum 5.0
                precision 1.0
            }
        }
        partitions 5
    }
```

Στην συνέχεια, ορίζονται οι αλγόριθμοι που θα εκτιμηθούν μαζί με τις παραμέτρους του καθενός. Ο αλγόριθμος_1 (algo1) ορίζει τον UserUser αλγόριθμο collaborative Filtering, τον Predictor που θα χρησιμοποιήσει, τους παράγοντες κανονικοποίησης και το μέγεθος της γειτονιάς την οποία θα λάβει υπόψη του για τις προβλέψεις. Ο αλγόριθμος_2 (algo2) απ' την άλλη, χρησιμοποιεί τον αλγόριθμο FunkSVD, ορίζοντάς του τον Scorer που θα υλοποιήσει για τα Predictions, τον αριθμό των χαρακτηριστικών (FeatureCount) που θα δημιουργήσει για κάθε αντικείμενο, καθώς και τον αριθμό των επαναλήψεων που θα εκτελέσει για κάθε πρόβλεψη (IterationCount). Επίσης, ο αλγόριθμος_3 (algo3) ορίζει τον αλγόριθμο

Item-Item με τον Scorer του και ένα Baseline Score για την περίπτωση που δεν βρεθούν ομοιότητες για να παραχθούν πιο ακριβείς συστάσεις.

```
def algo1 = algorithm("UserUser") {
    bind ItemScorer to UserUserItemScorer
    bind (BaselineScorer, ItemScorer) to UserMeanItemScorer
    bind VectorNormalizer to MeanVarianceNormalizer
    set NeighborhoodSize to 1000
}
def algo2 = algorithm("FunkSVD") {
    bind ItemScorer to FunkSVDItemScorer
    bind (BaselineScorer, ItemScorer) to UserMeanItemScorer
    set IterationCount to 5
    set FeatureCount to 100
}
def algo3 = algorithm("PersMean") {
    bind ItemScorer to UserMeanItemScorer
    bind (UserMeanBaseline, ItemScorer) to ItemMeanRatingItemScorer
}
```

Τέλος, ορίζονται οι μετρικές οι οποίες θα υπολογιστούν για την αξιολόγηση, ορίζεται το αρχείο εξόδου με τα αποτελέσματα ανά γράφο πλέον και γίνεται εκτύπωση ενώ διαγράμματος με όλα τα δομικά στοιχεία (components) του LensKit Evaluator καθώς και της ροής των δεδομένων μεταξύ τους.

```
      metric CoveragePredictMetric
      metric RMSEPredictMetric
      metric NDCGPredictMetric
      metric topNnDCG {
            listSize 5
      }
      metric MAEPredictMetric

      output "eval-results.csv"
      dumpGraph {
          output "graph.dot"
          algorithm("UserUser")
      }
```

Ωστόσο, η παραπάνω διαδικασία εκτελείται για κάθε υπογράφο, γεγονός το οποίο σημαίνει ότι το αρχείο αποτελεσμάτων της αξιολόγησης θα πρέπει να επεξεργαστεί πριν την εκτέλεση της αξιολόγησης του επόμενου υπογράφου για να διατηρηθούν τα δεδομένα της κάθε εκτέλεσης και από την επεξεργασία του συνόλου των αποτελεσμάτων να προκύψει το τελικό αποτέλεσμα απόδοσης του αλγορίθμου για τον εκάστοτε αριθμό υπογράφων. Για τον λόγο αυτό, με την ολοκλήρωση την εργασίας αξιολόγησης για κάθε υπογράφο, καλείται η μέθοδος ***processTempCSV***. Η ***processTempCSV***, διαβάζει με τη σειρά της το αρχείο εξόδου με τα αποτελέσματα του evaluation για κάθε υπογράφο και αφού ομαδοποιήσει τα

αποτελέσματα των πέντε folds που έχουν εκτελεστεί για κάθε αλγόριθμο που αξιολογείται, υπολογίζει τον μέσο όρο των πέντε folds για κάθε μετρική που χρησιμοποιείται και επιστρέφει το τελικό αποτέλεσμα αξιολόγησης όλων των μετρικών για κάθε αλγόριθμο του συγκεκριμένου υπογράφου. Τέλος, το αποτέλεσμα που έχει επιστραφεί, αποθηκεύεται στο CSV αρχείο (Evaluation Results for All Clusters) με τα αποτελέσματα όλων των υπογράφων για τον συγκεκριμένο αριθμό υπογράφων που έχει οριστεί, και το οποίο είναι μετά το τέλος της αξιολόγησης διαθέσιμο για περαιτέρω ανάλυση της απόδοσης των αλγόριθμων πρόβλεψης σε σχέση με το μέγεθος των υπογράφων.

4.3 Αξιολόγηση της Πειραματικής Διαδικασίας

Όλα τα πειράματα υλοποιήθηκαν σε Java, με εξαίρεση το script εκτέλεσης του Recommendation-Evaluation, το οποίο ήταν γραμμένο σε Groovy. Η εκτέλεση των πειραμάτων έγινε χρησιμοποιώντας την πλατφόρμα του Okeanos, μια IaaS (Infrastructure as a Service) υπηρεσία της ΕΔΕΤ (Εθνικό Δίκτυο Έρευνας & Τεχνολογίας), και πιο συγκεκριμένα, σε δύο Linux based συστήματα 64-bit με 2-CPUs QEMU Virtual CPU Version 1.7.0 στα 2.1GHz, 512KB cache και 6GB μνήμης RAM το κάθε ένα.

Ο κύριος όγκος των πειραμάτων διεξήχθη στο σύνολο δεδομένων του Epinions, το οποίο για το σύνολο των πειραμάτων χωρίστηκε σε δεδομένα εκπαίδευσης και δεδομένα δοκιμών, ακολουθώντας μία 5-fold cross validation στρατηγική.

Επίσης, σε όλα τα πειράματα, αξιολογήθηκαν και οι τρεις αλγόριθμοι που αναφέρονται στην ενότητα 2.2· ο αλγόριθμος User-User, ο αλγόριθμος FunkSVD και ο αλγόριθμος Item-Item, αλλά να σημειωθεί ότι ο αλγόριθμος Item-Item χρησιμοποιήθηκε μόνο στις δύο πρώτες σειρές πειραμάτων, καθώς κρίθηκε πως οι ιδιότητες και τα χαρακτηριστικά του είναι τέτοια που δεν θα εξυπηρετούσαν τον σκοπό της τρίτης σειράς πειραμάτων. Παράλληλα, να σημειωθεί, ότι η διεξαγωγή 5-fold αξιολογήσεων ορίστηκε στο groovy script.

Επιπροσθέτως, για λόγους πληρότητας, κρίθηκε χρήσιμο να γίνει σύγκριση των πειραμάτων για το σύνολο δεδομένων του Epinions, με το σύνολο δεδομένων του Flixster. Έτσι, εκτελέστηκε εκ νέου η διαδικασία των ενοτήτων 4.3.1 και 4.3.2 για τον γράφο αξιολογήσεων του Flixster. Από αυτή τη διαδικασία θα προκύψουν και χρήσιμα συμπεράσματα για την απόδοση της λειτουργίας του μοντέλου σε γράφους με διαφορετική δομή, μέγεθος και πυκνότητα.

Πιο συγκεκριμένα, τα πειράματα που εκτελέστηκαν κατά την διάρκεια αυτής της μελέτης είναι τα ακόλουθα:

4.3.1 Εκτέλεση Collaborative Filtering στον πλήρη γράφο

Στόχος της εκτέλεσης αυτών των πειραμάτων είναι η εκτέλεση και των τριών αλγορίθμων Collaborative Filtering που αναφέρθηκαν, στο σύνολο δεδομένων του Epinions, χωρίς καμία επεξεργασία των δεδομένων από το σύστημα. Έτσι, τα αποτελέσματα αξιολόγησης των αλγορίθμων στο σύνολο του γράφου πριν υποστεί την διχοτόμηση θα αποτελέσουν ουσιαστικά το μέτρο σύγκρισης της απόδοσης του συστήματος που προτείνεται από την παρούσα εργασία. Η εκτέλεση των αλγορίθμων πραγματοποιήθηκε με το ίδιο configuration script και στα ίδια ακριβώς μηχανήματα με τα υπόλοιπα πειράματα. Αυτό πρακτικά σημαίνει ότι πραγματοποιήθηκε 5-fold cross validation, τα features που ορίστηκαν στον αλγόριθμο SVD ήταν 100, οι μετρικές που αξιολογήθηκαν ήταν όλες οι διαθέσιμες μετρικές του συστήματος και το listsize της μετρικής top-N nDCG ορίστηκε ίσο με 5.

Εδώ έχει μεγάλη σημασία να αναφερθεί ως αποτέλεσμα της εκτέλεσης αυτού του πειράματος ότι κατέστη αδύνατη η εκτέλεση του πειράματος σε ολόκληρο τον γράφο με τις πιο πάνω ρυθμίσεις και στο σύστημα που περιγράφεται, καθώς η μνήμη του συστήματος δεν επαρκούσε για την επεξεργασία του γράφου. Ωστόσο, για λόγους πληρότητας, διαπιστώθηκε ότι η ελάχιστη απαιτούμενη μνήμη συστήματος για την εκτέλεση του πειράματος στον πλήρη γράφο ήταν 8GB, παρόλα αυτά ως μέτρο σύγκρισης για την συνέχεια των πειραμάτων διατηρείται η αδυναμία εκτέλεσης του πειράματος για τα 6GB τα οποία χρησιμοποιήθηκαν στα συστήματα της πειραματικής διαδικασίας.

4.3.2 Εκτέλεση Collaborative Filtering στους πλήρεις υπογράφους

Σε αυτή τη σειρά πειραμάτων, έγινε η τμηματοποίηση του αρχικού γράφου για διαφορετικές τιμές του πλήθους των υπογράφων και για κάθε διαφορετικό αριθμό υπογράφων εκτελέστηκε η Recommendation-Evaluation εργασία για την αξιολόγηση της απόδοσης καθενός από τους τρεις αλγορίθμους. Η αξιολόγηση έγινε με ακριβώς τα ίδια χαρακτηριστικά εκτέλεσης όπως και στον πλήρη γράφο, δηλαδή με 5-fold cross evaluation, 100 features για τον αλγόριθμο SVD, χρήση και των έξι μετρικών αξιολόγησης και top-N nDCG με N=5.

Οι τιμές των υπογράφων που εξετάστηκαν ήταν για $s = \{1, 2, 4, 8, 16, 33, 65, 125, 250, 500, 1000\}$, ενώ πρέπει να σημειωθεί ότι σε αυτό το στάδιο των πειραμάτων, για κάθε αλγόριθμο που έτρεξε πάνω στον κάθε υπογράφο, χρησιμοποιήθηκαν όλοι οι χρήστες του υπογράφου χωρίς να γίνει κάποιος διαχωρισμός στους k-πλησιέστερους γείτονες.

4.3.3 Εκτέλεση Collaborative Filtering στους υπογράφους με K-nn γείτονες

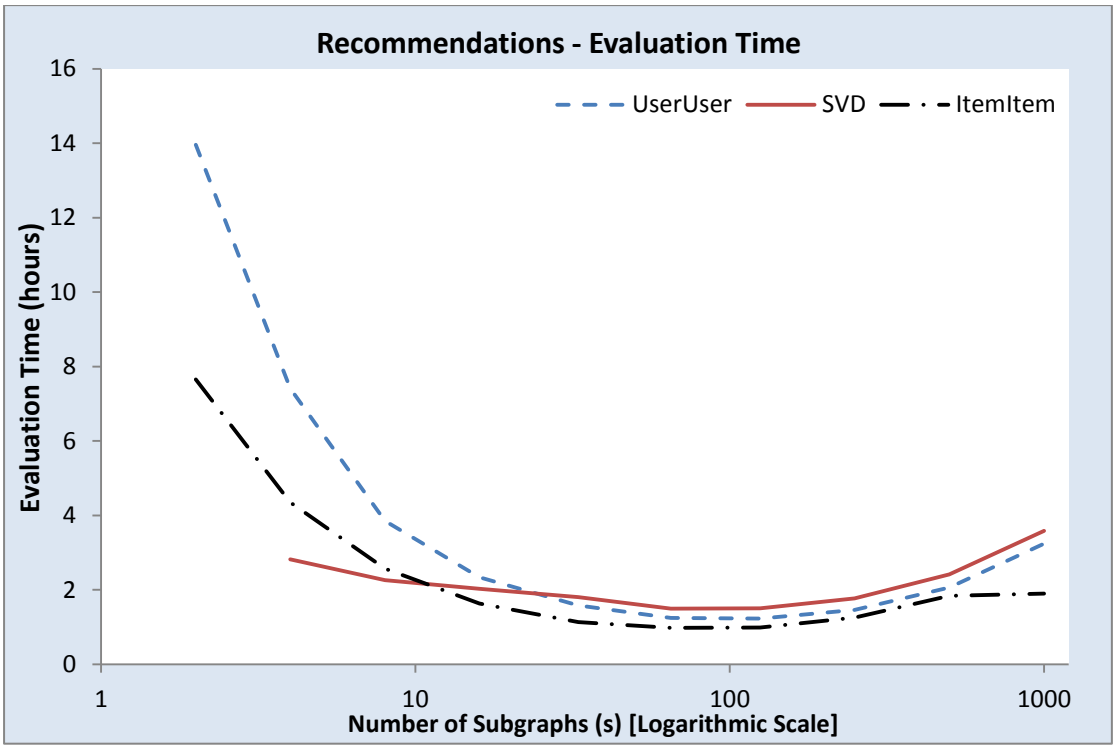
Έχοντας αξιολογήσει την απόδοση του συστήματος για τις διάφορες τιμές πλήθους των υπογράφων, θεωρήθηκε σκόπιμο η εξέταση της συμπεριφοράς του συστήματος και της απόδοσης των αλγορίθμων λαμβάνοντας κάθε φορά ένα μέρος των χρηστών του κάθε υπογράφου ως k-πλησιέστερους γείτονες. Έτσι, σε αυτό το στάδιο, εκτελέστηκαν πειράματα χρησιμοποιώντας και πάλι το ίδιο configuration script με τις ίδιες ακριβώς ρυθμίσεις μετρικών όπως και προηγουμένως, με τη διαφορά όμως ότι για κάθε εκτέλεση πειράματος ορίζεται δυναμικά ο αριθμός του Neighborhood Size που θα χρησιμοποιηθεί για την αξιολόγηση.

Όπως γίνεται κατανοητό, οι αλγόριθμοι Item-Item και FunkSVD που χρησιμοποιήθηκαν έως τώρα δεν χρήζουν επανεκτέλεσης, καθώς δεν επηρεάζονται από το μέγεθος του Neighborhood και κατά συνέπεια δεν διαθέτουν κάποια παράμετρο ρύθμισης του μεγέθους της γειτονιάς κάθε κόμβου. Συνεπώς, τα πειράματα που εκτελέστηκαν, είχαν ως αντικείμενο εξέτασης την συμπεριφορά του αλγόριθμου User-User για τρεις τιμές υπογράφων $s = \{4, 65, 1000\}$ και για Neighborhood Size $k = \{1, 3, 5, 10, 25, 50, 100, 500, 1000\}$.

4.4 Αποτελέσματα

4.4.1 Αποτελέσματα για διαφορετικό πλήθος υπογράφων – s

Μετά από την εκτέλεση των πειραμάτων για τον πλήρη γράφο καθώς και για τους πλήρεις υπογράφους για τις διαφορετικές τιμές του s, το πρώτο βασικό αποτέλεσμα προκύπτει από την μέτρηση του χρόνου που χρειάστηκε το σύστημα για την αξιολόγηση του κάθε subgraph. Στο διάγραμμα που ακολουθεί, απεικονίζεται ο χρόνος εκτέλεσης της διαδικασίας αξιολόγησης (σε ώρες) για κάθε αλγόριθμο, για τις διάφορες τιμές s των υπογράφων. Να σημειωθεί ότι ο οριζόντιος άξονας σε όλα τα γραφήματα που απεικονίζει τον αριθμό των υπογράφων είναι σε λογαριθμική κλίμακα με βάση το δέκα ($\log_{10}$).



Εικόνα 14: Χρόνος εκτέλεσης της αξιολόγησης κάθε αλγόριθμου για τους διαφορετικούς υπογράφους (Epinions)

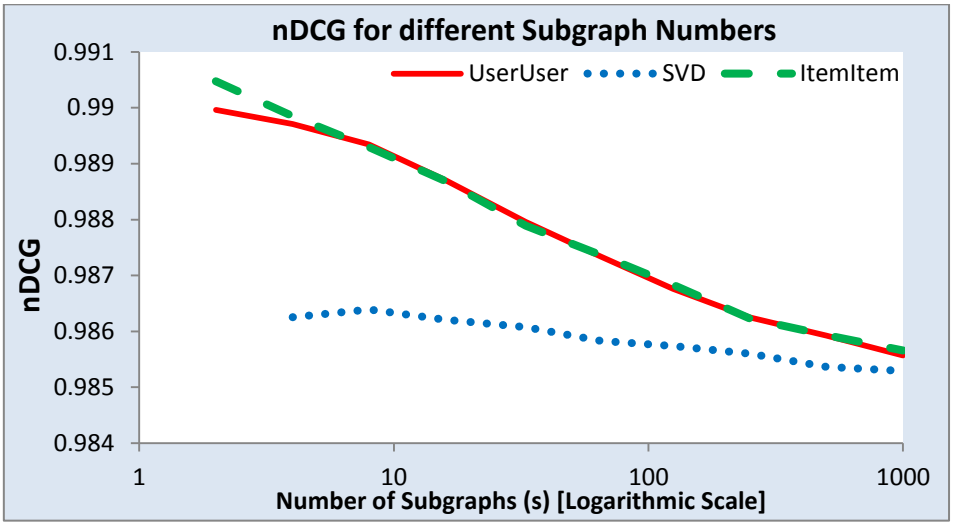
Όπως παρατηρούμε στην Εικόνα 14, όσο αυξάνεται ο αριθμός των υπογράφων στον οποίο χωρίζουμε τον αρχικό γράφο του Epinions, μειώνεται ραγδαία ο χρόνος εκτέλεσης που απαιτείται από το σύστημα για να παραγάγει συστάσεις και να πραγματοποιήσει την αξιολόγηση των αλγορίθμων, έως την τιμή των 65 υπογράφων, όπου ο χρόνος εκτέλεσης αρχίζει μια μικρή αλλά ανοδική τάση. Από άποψη φόρτου και υπολογιστικής ισχύος, κι όπως φαίνεται και από τα

πειράματα στα δύο σύνολα δεδομένων ένας αριθμός 65 υπογράφων έχει την μεγαλύτερη απόδοση.

Επίσης, από το παραπάνω γράφημα για τον γράφο του Epinions, παρατηρούμε ότι από τους 16 υπογράφους (7,530 χρήστες κατά μέσο όρο) και μετά ο αλγόριθμος Item-Item είναι γρηγορότερος σε εκτέλεση από τον SVD και τον User-User, ενώ γίνεται προφανές, ότι για τους αλγόριθμους User-User και Item-Item δεν ήταν δυνατή η εκτέλεσή τους στον πλήρη - μη διχοτομημένο γράφο, ενώ ο αλγόριθμος SVD δεν μπόρεσε να εκτελεστεί για λιγότερα από 4 υπογράφους (~30.123 χρήστες κατά μέσο όρο), το οποίο οφείλεται στον τρόπο λειτουργίας του αλγόριθμου SVD για τον υπολογισμό τον συστάσεων και κατά συνέπεια η μνήμη του συστήματος ήταν ανεπαρκής για την δημιουργία των πινάκων με τα αντίστοιχα διανύσματα των χαρακτηριστικών (features) που προσπαθεί να δημιουργήσει ο αλγόριθμος για λιγότερα από τέσσερις υπογράφους.

Τέλος, από το γράφημα των χρόνων εκτέλεσης, μπορούμε να παρατηρήσουμε ότι σε αντίθεση με τους αλγόριθμους User-User και SVD οι οποίοι μετά τους 1000 υπογράφους έχουν μία αυξητική τάση στον χρόνο εκτέλεσης, ο χρόνος εκτέλεσης του αλγόριθμου Item-Item φαίνεται να σταθεροποιείται περίπου στην 1 ώρα και 54 λεπτά (1h:54m).

Όσον αφορά την απόδοση των αλγορίθμων του μοντέλου κατά nDCG (Εικόνα 15), φαίνεται πως το υπολογιζόμενο κέρδος για τους διαφορετικούς υπογράφους φτάνει στη μέγιστη τιμή, ίση με 0,99. Μάλιστα, συγκρίνοντας την απόδοση των τριών αλγορίθμων ξεχωριστά, παρατηρούμε πως οι μεταξύ τους διαφορές είναι της τάξης τους τρίτου δεκαδικού (0,005).

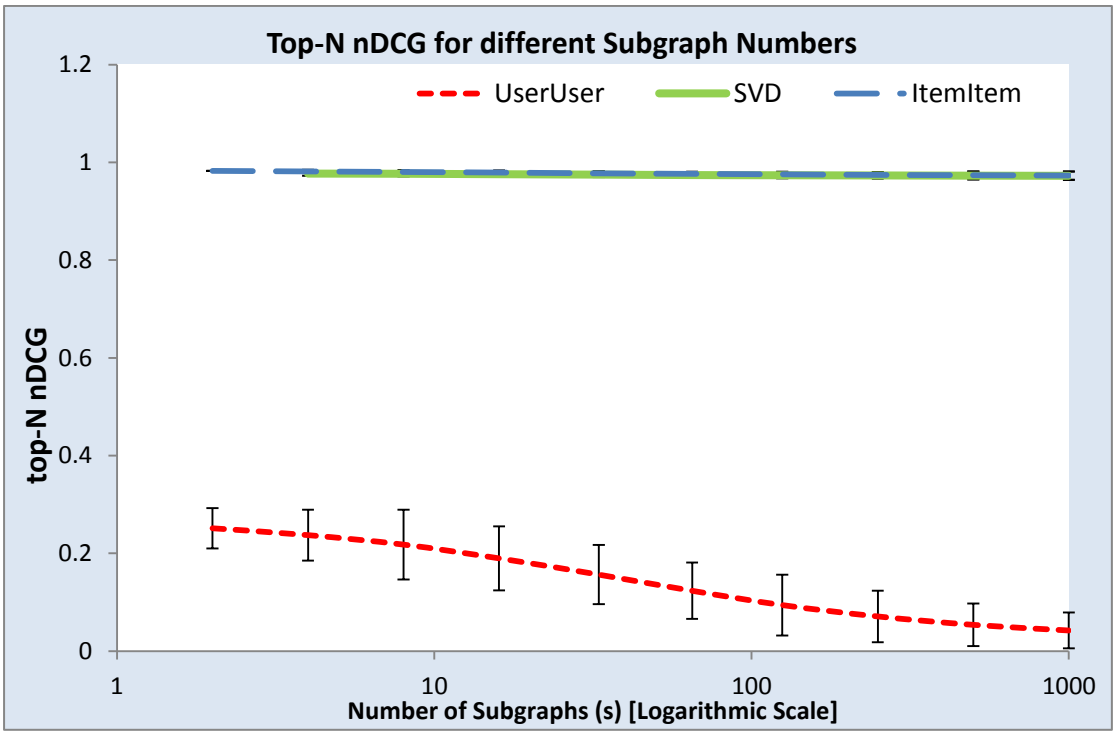


Εικόνα 15: Απόδοση κατά nDCG όλων των αλγορίθμων στο Epinions για τους διαφορετικούς υπογράφους (s)

Πρακτικά δηλαδή, η απόδοση και των τριών αλγορίθμων κατά nDCG (normalized Discounted Cumulative Gain) προσεγγίζει την ιδανική τιμή κέρδους nDCG=1. Αυτό σημαίνει, ότι μπορούμε ανάλογα με την δομή και την πυκνότητα του γράφου ως προς τους χρήστες και τα αντικείμενα, να χρησιμοποιήσουμε τον αλγόριθμο που ταιριάζει καλύτερα στα χαρακτηριστικά του γράφου και εφαρμόζοντας το μοντέλο αυτό στους s υπογράφους να πετύχουμε συστάσεις ανάλογες με αυτές που θα είχαμε για τον πλήρη υπογράφο και μάλιστα ιδιαίτερα ταχύτερα.

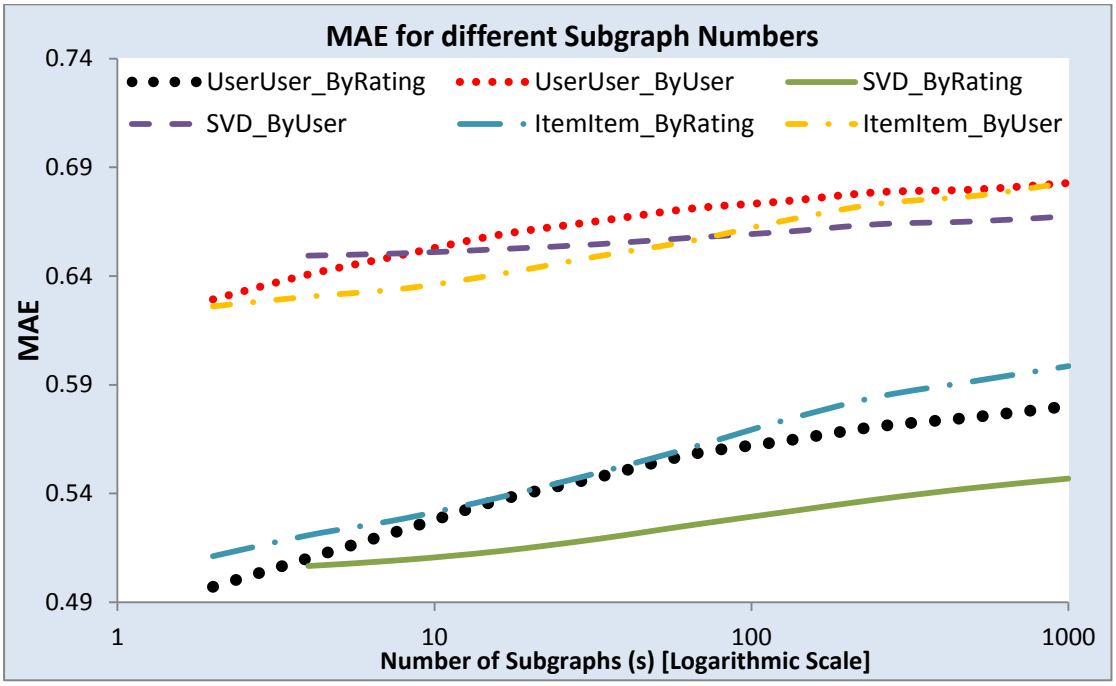
Το γεγονός αυτό, επιβεβαιώνεται και από την μέτρηση της απόδοσης με την μετρική Top-N nDCG, που αξιολογεί μόνο τις 5 πρώτες προτάσεις (N=5), όπως απεικονίζεται στην Εικόνα 16, όπου οι αλγόριθμοι SVD και Item-Item έχουν κέρδος κοντά στην μονάδα. Εξαίρεση όμως αποτελεί ο αλγόριθμος User-User ο οποίος έπιασε πολύ χαμηλές τιμές κέρδους και μάλιστα αντιστρόφως ανάλογες της αύξησης των αριθμών των υπογράφων s. Αυτό δικαιολογείται από το γεγονός ότι το πλήθος των αντικειμένων είναι ιδιαίτερα μεγάλο, με άμεσο επακόλουθο οι αξιολογήσεις που περιέχει κάθε υπογράφος να είναι αρκετά διαφορετικές μεταξύ τους και κατά συνέπεια ο αλγόριθμος User-User να μην είναι σε θέση να εντοπίσει όμοιους με βάση τις βαθμολογίες τους χρήστες, άρα και οι συστάσεις που κάνει για κάθε χρήστη να μην είναι ιδιαίτερα επιτυχείς. Ο παραπάνω λόγος, επιβεβαιώνει και την τυπική απόκλιση των τιμών κέρδους κοιτάζοντας την Εικόνα 16, η οποία για τους αλγόριθμους SVD και Item-Item είναι σχεδόν μηδενική, σε αντίθεση με την

τυπική απόκλιση των τιμών του User-User, η οποία παρουσιάζει μικρές διακυμάνσεις. Η καλύτερη απόδοσή τους σε σχέση με τον User-User οφείλεται αφενός στο ότι ο SVD κάνει ένα είδος αντιστοίχησης του χώρου των αντικειμένων σε έναν μικρότερο χώρο διανυσμάτων, το οποίο είναι και λιγότερο αραιό (sparse). Αφετέρου, οφείλεται στο ότι ο αλγόριθμος Item-Item αναζητεί αντικείμενα με όμοιες βαθμολογίες, κατά συνέπεια, μπορεί να βρει ομοιότητες μέσα στους υπογράφους, ανεξάρτητα αν οι βαθμολογίες των χρηστών είναι αρκετά αραιές ώστε να μπορεί ο User-User να εντοπίσει όμοιους χρήστες.



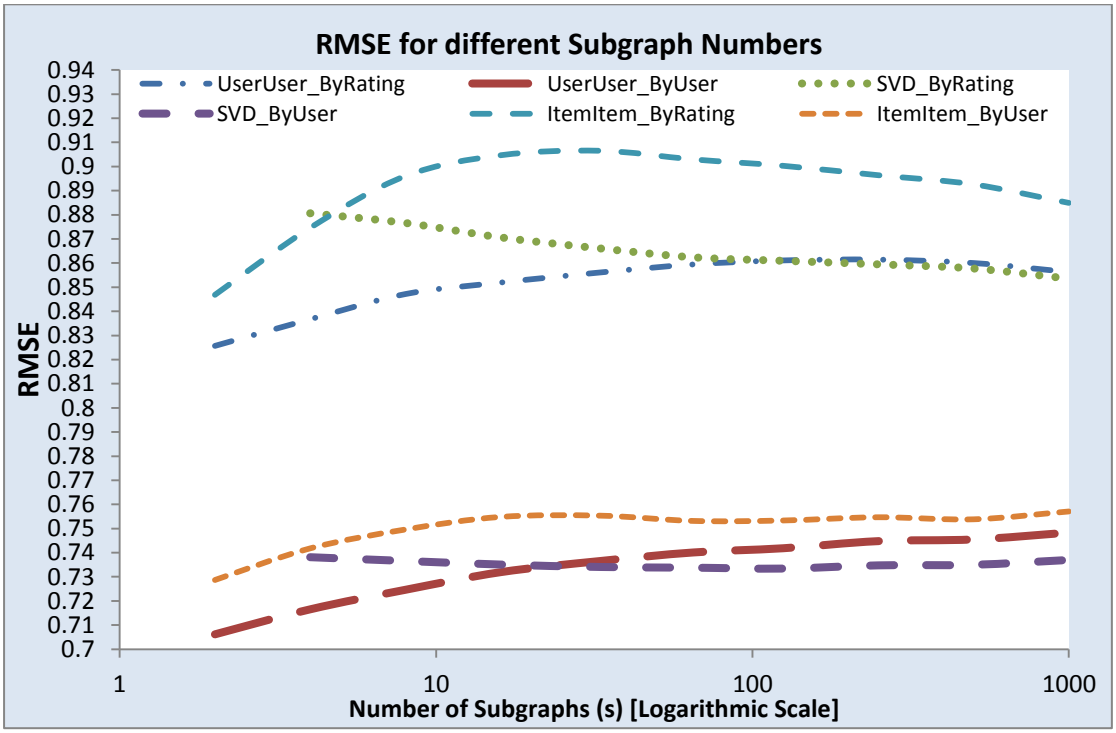
Εικόνα 16: Απόδοση κατά Top-N nDCG όλων των αλγορίθμων στο Epinions για τους διαφορετικούς υπογράφους (s)

Σε συνέχεια των παραπάνω αποτελεσμάτων, στην Εικόνα 17, απεικονίζονται τα αποτελέσματα της απόδοσης των αλγορίθμων κατά MAE, η οποία είναι και μία από τις πιο συχνά χρησιμοποιούμενες μετρικές αποτελεσμάτων αλγόριθμων συστάσεων.



Εικόνα 17: Απόδοση κατά MAE όλων των αλγορίθμων στο Epinions για τους διαφορετικούς υπογράφους (s)

Όπως προκύπτει από την Εικόνα 17, η απόδοση του μοντέλου σύμφωνα με την μετρική MAE κρίνεται αρκετά ικανοποιητική και ιδιαίτερα για τον αλγόριθμο SVD και τους τέσσερεις υπογράφους, καθώς το σφάλμα πρόβλεψης που προκύπτει για την MAE_ByRating είναι 0,5 μονάδες και φτάνει ως μέγιστη τιμή σφάλματος τις 0,55 μονάδες για τους 1000 υπογράφους. Θα πρέπει να αναφερθεί ότι λόγω της μη κανονικοποίησης στις μετρικές MAE και RMSE, οι μέγιστες τιμές που μπορούν να πάρουν τα σφάλματα είναι μεγαλύτερα της μονάδας, καθώς το εύρος τιμών των βαθμολογιών είναι [1..5]. Η επιτυχία του συγκεκριμένου αλγόριθμου εξηγείται από το γεγονός ότι ακόμα και στους 1000 υπογράφους, το πλήθος των διαφορετικών αντικειμένων τα οποία έχουν αξιολογηθεί έστω και από ένα χρήστη, είναι ακόμη μεγάλο. Το ίδιο ισχύει και όσο μικραίνει ο αριθμός των υπογράφων και κατά συνέπεια αυξάνει το μέγεθός τους, ακόμη κι αν δεν υπάρχουν επικαλύψεις μεταξύ των χρηστών. Επακόλουθο αυτού είναι ο αλγόριθμος SVD να έχει μεγαλύτερο εύρος αντικειμένων στα οποία ψάχνει για να αντιστοιχήσει τα ζητούμενα features και άρα μπορεί να προβλέψει εγκυρότερες συστάσεις για τον χρήστη. Το γεγονός αυτό έχει ιδιαίτερο ενδιαφέρον και για έναν άλλο λόγο, καθώς αποδεικνύει την δυνατότητα πρόβλεψης και σύστασης αντικειμένων ακόμα και σε μεγάλους γράφους και ανάμεσα από χιλιάδες αντικείμενα με σχετικά μεγάλη εγκυρότητα και επιτυχία.

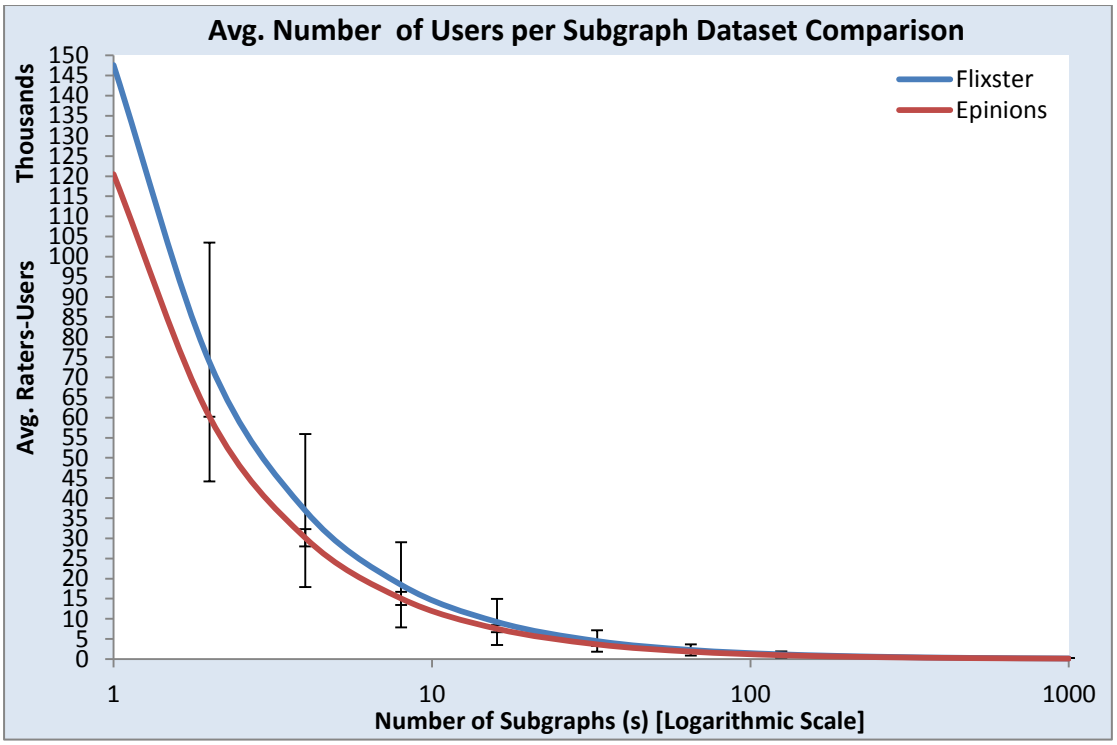


Εικόνα 18: Απόδοση κατά RMSE όλων των αλγορίθμων στο Epinions για τους διαφορετικούς υπογράφους (s)

Αντίστοιχα με την MAE, ήταν και τα αποτελέσματα της Εικόνας 18 για την μετρική RMSE, όπου η καλύτερη επίδοση του μοντέλου προκύπτει από τον αλγόριθμο User-User για $n = 2$ υπογράφους με τιμή 0,7. Λογικό συμπέρασμα καθώς στους δύο υπογράφους υπάρχει αρκετή πληροφορία ώστε ο αλγόριθμος να βρει αρκετούς όμοιους χρήστες για να προτείνει πιθανά αντικείμενα. Ωστόσο, ακόμα και στους 65 ή 125 υπογράφους, ο αλγόριθμος SVD μπορεί να δώσει λύση, προβλέποντας πιθανά αντικείμενα με σφάλμα $\sim 0{,}73$.

Για λόγους πληρότητας, τα ίδια ακριβώς πειράματα, με το ίδιο configuration script, εκτελέστηκαν και στο σύνολο δεδομένων του Flixster, να σημειωθεί όμως ότι λόγω περιορισμού στη μνήμη του συστήματος, για s=16 οι αλγόριθμοι User-User και SVD δεν μπόρεσαν να τρέξουν σε έναν υπογράφο και κατά συνέπεια, για τους υπολογισμούς αυτών των αλγορίθμων για s=16 χρησιμοποιήθηκαν 15 από τους 16 υπογράφους. Επίσης, για τους 1000 υπογράφους, λόγω του μοιράσματος των χρηστών στους υπογράφων υπήρξαν 24 υπογράφοι στα οποία είτε δεν υπήρξαν βαθμολογίες, είτε όλες οι βαθμολογίες ήταν από έναν χρήστη και συνεπώς χρησιμοποιήθηκαν οι υπόλοιποι 976 για το evaluation.

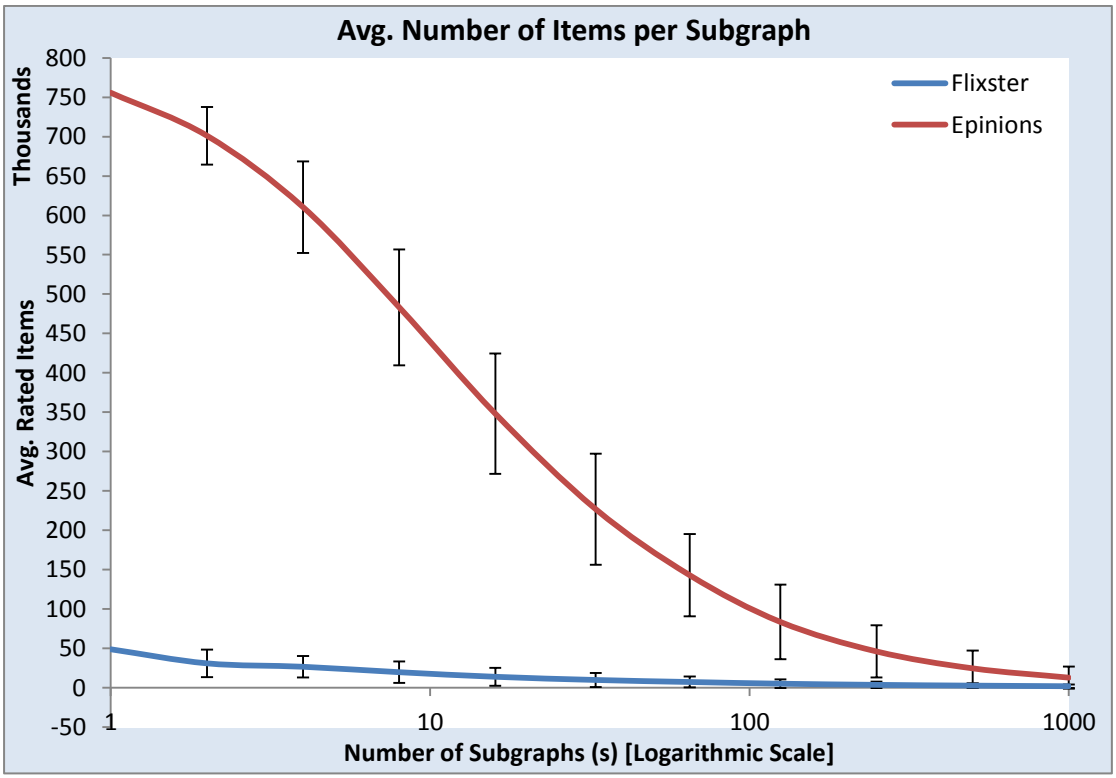
Σε αυτό το σημείο, αξίζει να κάνουμε μία σύγκριση μεταξύ των δύο γράφων αξιολογήσεων στα δύο σύνολα δεδομένων. Στην Εικόνα 19 απεικονίζεται η κατανομή των χρηστών που έχουν κάνει έστω μία αξιολόγηση ανά subgraph για κάθε ένα από τα σύνολα δεδομένων, μαζί με την τυπική απόκλιση για κάθε subgraph. Έτσι, μπορούμε να παρατηρήσουμε ότι αν και αρχικά τα δύο dataset έχουν μεγάλη διαφορά στο πλήθος των χρηστών, από τους 16 περίπου υπογράφους και πάνω αρχίζουν να έχουν αρκετά μικρές διαφορές τόσο στο πλήθος χρηστών όσο και στην απόκλιση για κάθε υπογράφο, σε βαθμό ώστε να μπορούμε να θεωρήσουμε τα δύο σύνολα συγκρίσιμα για κάθε υπογράφο. Η μεγάλη τυπική απόκλιση στον αριθμό των βαθμολογητών ανά υπογράφο δεν οφείλεται σε αδυναμία του Metis να ισομοιράσει τον κοινωνικό γράφο, αλλά στο γεγονός ότι το Flixster έχει μεγάλο αριθμό χρηστών που δεν δίνουν βαθμολογίες, οι οποίοι όμως συμμετέχουν στον κοινωνικό γράφο και μοιράζονται στους υπογράφους.



Εικόνα 19: Σύγκριση του μέσου πλήθους μοναδικών χρηστών ανά υπογράφο για τα δύο σύνολα δεδομένων

Παράλληλα με την σύγκριση των συνόλων ως προς το πλήθος των αξιολογητών ανά υπογράφο, πρέπει να συγκρίνουμε και την κατανομή των μοναδικών αντικειμένων τα οποία έχουν αξιολογηθεί για κάθε πλήθος υπογράφων ανάμεσα στα δύο σύνολα δεδομένων, το οποίο φαίνεται στην Εικόνα 20. Όπως

είναι εμφανές και στην Εικόνα 20, για οποιονδήποτε αριθμό υπογράφων, το πλήθος των αντικειμένων που έχουν αξιολογηθεί από τους χρήστες στο σύνολο δεδομένων του Epinions, είναι πολλαπλάσια αυτών που έχουν βαθμολογηθεί στο σύνολο του Flixster. Από τη μία, το μεγάλο πλήθος αντικειμένων του Epinions δικαιολογεί, όπως προαναφέρθηκε, κάποια από τα συμπεράσματα που προηγήθηκαν για την συμπεριφορά ορισμένων αλγορίθμων, ενώ από την άλλη δείχνει μία τάση για την συμπεριφορά που αναμένεται να έχουν κάποιοι από τους αλγόριθμους στο σύνολο του Flixster.

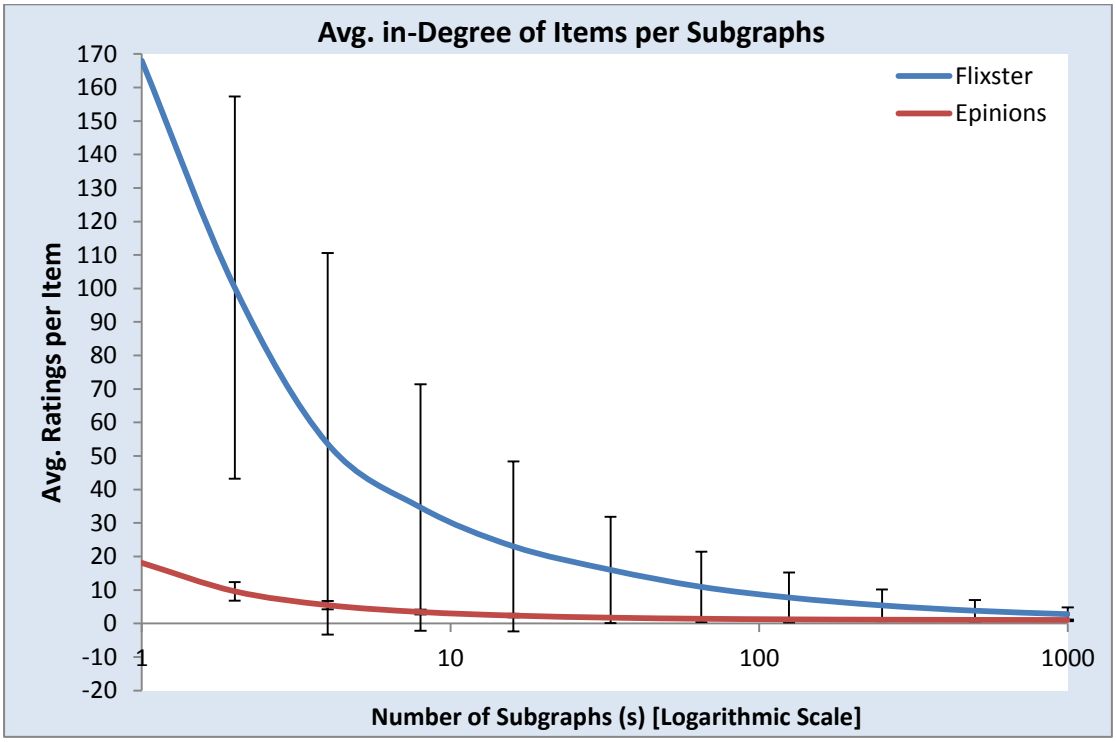


Εικόνα 20: Σύγκριση του μέσου πλήθους μοναδικών αντικειμένων ανά υπογράφο για τα δύο σύνολα δεδομένων

Όπως λοιπόν προκύπτει από την παραπάνω σύγκριση των χρηστών που έχουν βαθμολογήσει κάποιο αντικείμενο για τους διάφορους υπογράφους και του πλήθους των αντικειμένων που έχουν βαθμολογηθεί μέσα σε αυτά, στην Εικόνα 21 μπορούμε να δούμε την κατανομή του in-Degree των αντικειμένων για κάθε υπογράφο. Έτσι, όπως είναι αναμενόμενο, εφόσον ο μέσος όρος αξιολογητών ανά υπογράφο είναι προσεγγιστικά όμοιος ανάμεσα στα δύο σύνολα και το σύνολο του Epinions έχει πολύ περισσότερα αντικείμενα που αξιολογούνται ανά υπογράφο, ο μέσος όρος του πλήθους των αξιολογήσεων που λαμβάνει κάθε αντικείμενο μέσα σε κάθε υπογράφο για το σύνολο δεδομένων του Epinions είναι έως και οκτώ φορές

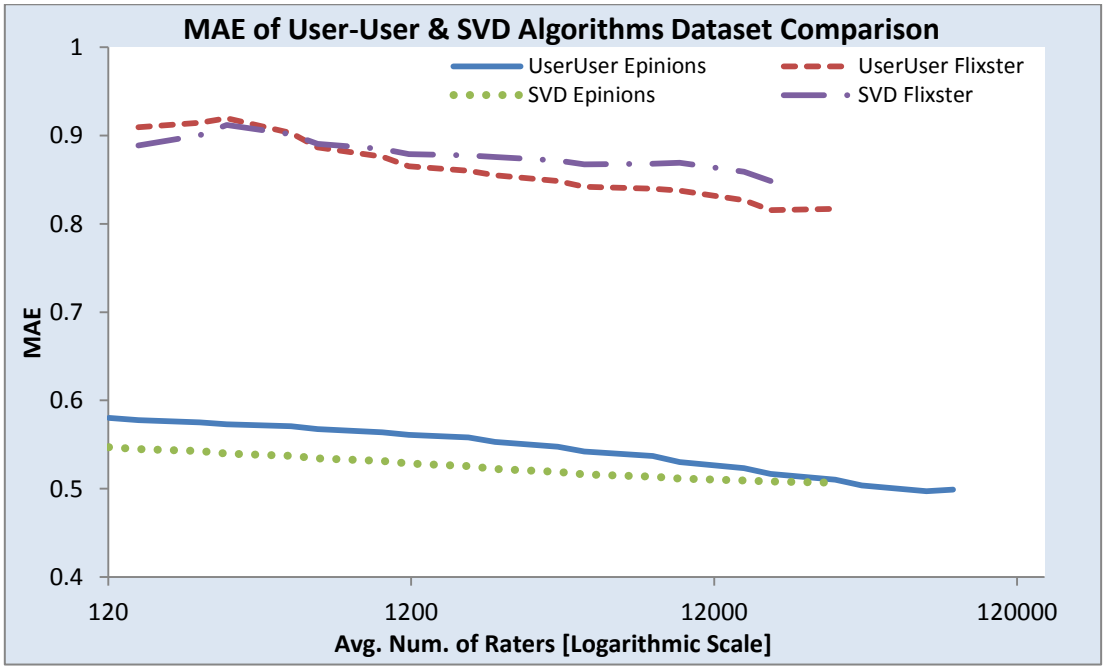
μικρότερος από αυτόν στο σύνολο δεδομένων του Flixster, γεγονός που επιβεβαιώνει την διαφορά στην απόδοση αλγορίθμων όπως ο User-User, που αναμένουμε να υπάρξει ανάμεσα στα δύο σύνολα λόγω της δυσκολίας του να ανιχνεύσει όμοιους χρήστες σε αραιούς γράφους αξιολογήσεων.

Φυσικά, η διαφορά αυτή ανάμεσα στα δύο σύνολα, τείνει να μειώνεται όσο αυξάνεται το πλήθος των υπογράφων, κάτι το οποίο είναι λογικό εφόσον όπως είπαμε το πλήθος των χρηστών από τους 16 υπογράφους και πάνω τείνει να είναι όμοιος και κατά συνέπεια, όσο μειώνεται το μέγεθος των υπογράφων μειώνεται κατ' αντιστοιχία και το πλήθος των βαθμολογιών το οποίο μεταφέρεται σε κάθε υπογράφο, άρα και το πλήθος των βαθμολογιών ανά προϊόν τείνει να είναι μικρό και να προσεγγίζει όμοιες τιμές και στα δύο σύνολα. Από την Εικόνα 21 επίσης, πρέπει να σημειώσουμε την μεγάλη απόκλιση που υπάρχει στο πλήθος των βαθμολογιών ανά αντικείμενο για κάθε υπογράφο στο σύνολο δεδομένων του Flixster και ιδιαίτερα κάτω από τους 65 υπογράφους, το οποίο οφείλεται στον κοινωνικό γράφο του Flixster, λόγω του οποίου από το μοίρασμα των χρηστών σε υπογράφους που έγινε από το Metis προέκυψαν αρκετά ανόμοιοι σε μέγεθος υπογράφοι, με αποτέλεσμα για κάθε πλήθος υπογράφων να υπάρχουν κάποιοι υπογράφοι με πολλές βαθμολογίες και άλλοι με ελάχιστες.

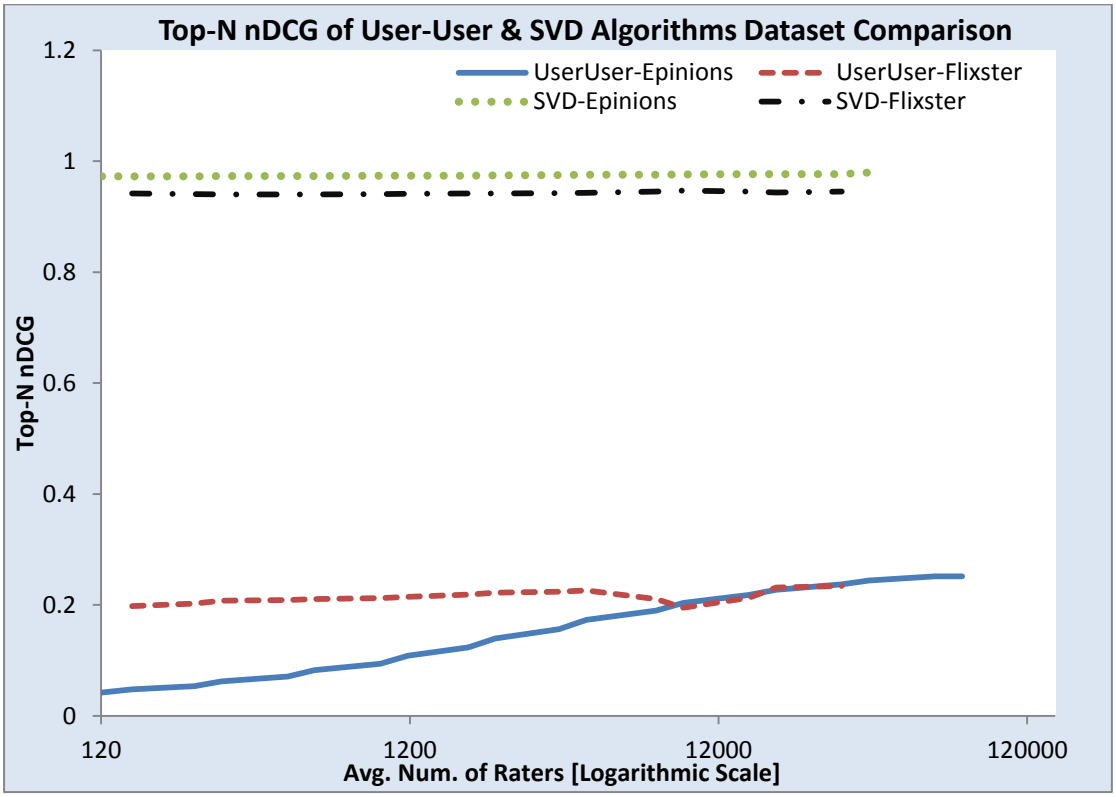


Εικόνα 21: Σύγκριση του μέσου πλήθους βαθμολογιών ανά αντικείμενο (in-Degree), ανά υπογράφο για τα δύο σύνολα δεδομένων

Στην Εικόνα 22, φαίνεται η σύγκριση των αποτελεσμάτων κατά την μετρική MAE, και των δύο συνόλων δεδομένων για τους αλγόριθμους User-User και SVD ως προς τον μέσο αριθμό χρηστών, ενώ στην Εικόνα 23, έχουμε την σύγκριση της απόδοσης των δύο συνόλων κατά Top-N nDCG για τους ίδιους αλγόριθμους.



Εικόνα 22: Σύγκριση του μέσου σφάλματος των αλγορίθμων User-User_ByRating & SVD ως προς τον μέσο αριθμό χρηστών σε κάθε υπογράφο για τα δύο datasets



Εικόνα 23: Σύγκριση του κανονικοποιημένου κέρδους των αλγορίθμων User-User_ByRating & SVD ως προς τον μέσο αριθμό χρηστών σε κάθε υπογράφο για τα δύο datasets

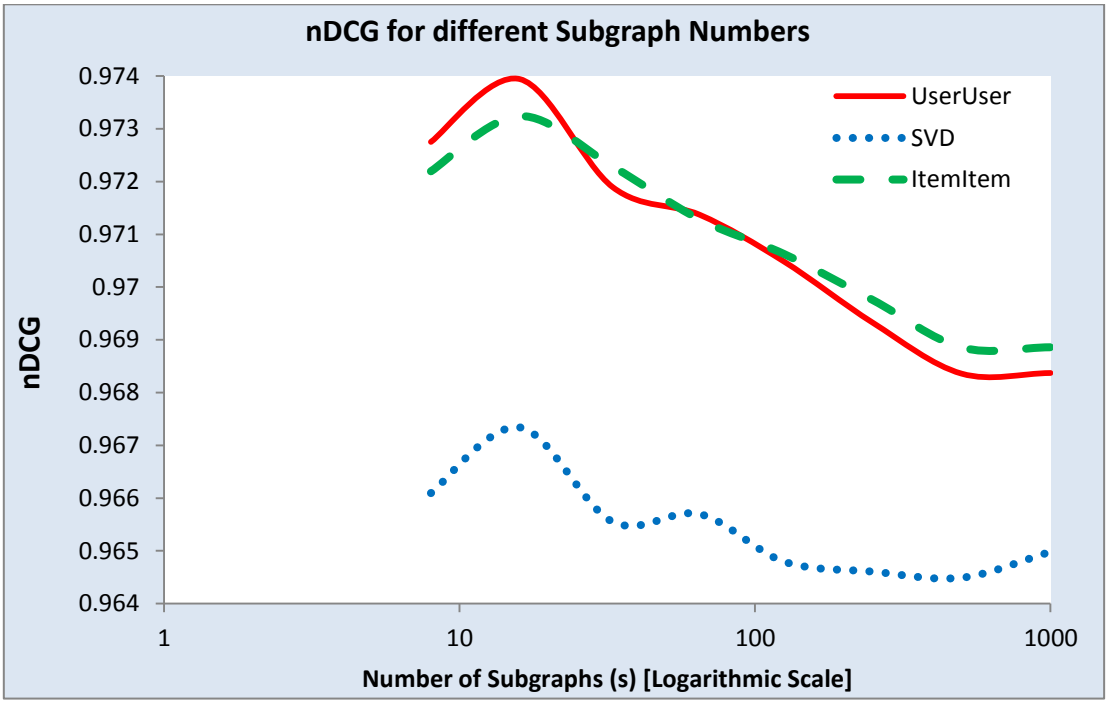
Όπως μπορούμε να παρατηρήσουμε από την Εικόνα 23, ο αλγόριθμος User-User εκ των πραγμάτων δεν αποδίδει καλά, αποδίδει όμως καλύτερα στο Flixster και τους μικρούς υπογράφους που περιέχουν λίγους χρήστες σε σχέση με το Epinions κι αυτό γιατί στο Epinions υπάρχουν πολύ περισσότερα προϊόντα, πολύ λιγότερες βαθμολογίες για το καθένα από αυτά, με αποτέλεσμα να είναι πολύ δύσκολο να βρεθούν όμοιοι χρήστες. Αυτό επηρεάζει και το Top-N nDCG καθώς η επικάλυψη μεταξύ προτάσεων και πραγματικών προτιμήσεων είναι πάντοτε πολύ μικρή. Όπως μπορούμε όμως να δούμε στην Εικόνα 22, το μέσο απόλυτο σφάλμα για τον αλγόριθμο User-User στο Flixster είναι μεγαλύτερο από αυτό του Epinions. Αυτό δικαιολογείται από το γεγονός ότι το πλήθος των βαθμολογιών ανά χρήστη στο Epinions είναι πολύ μεγαλύτερο από αυτό του Flixster με αποτέλεσμα ο αλγόριθμος να βρίσκει πιο εύκολα όμοιους χρήστες για να μπορεί να προβλέψει καλύτερα αποτελέσματα. Το ίδιο συμπέρασμα προκύπτει και για τον αλγόριθμο SVD από την Εικόνα 22, μόνο που λόγω της λειτουργίας του αλγόριθμου επαληθεύεται ότι μπορεί να προτείνει αντικείμενα με μεγαλύτερη ακρίβεια από τον User-User.

Όπως προκύπτει από τα παραπάνω, η τεχνική που προτείνεται από την παρούσα διπλωματική μελέτη, έχει το μειονέκτημα της χαμηλής απόδοσης όταν καταλήγουμε να έχουμε sparse bipartite γράφους αξιολογήσεων, μπορεί όμως ακόμα και σε αυτήν την περίπτωση να δώσει λύσεις με άλλους αλγόριθμους όπως ο SVD. Το γεγονός αυτό υποστηρίζεται και από τα αποτελέσματα της Εικόνας 23, καθώς όπως φαίνεται ο αλγόριθμος SVD είναι ανεξάρτητος από το πλήθος των βαθμολογιών και από το πλήθος των βαθμολογούμενων αντικειμένων, γιατί κάνει όπως έχουμε αναφέρει, το aggregation σε features και συνεπώς λύνει το πρόβλημα των αραιών βαθμολογιών μέσα στους υπογράφους.

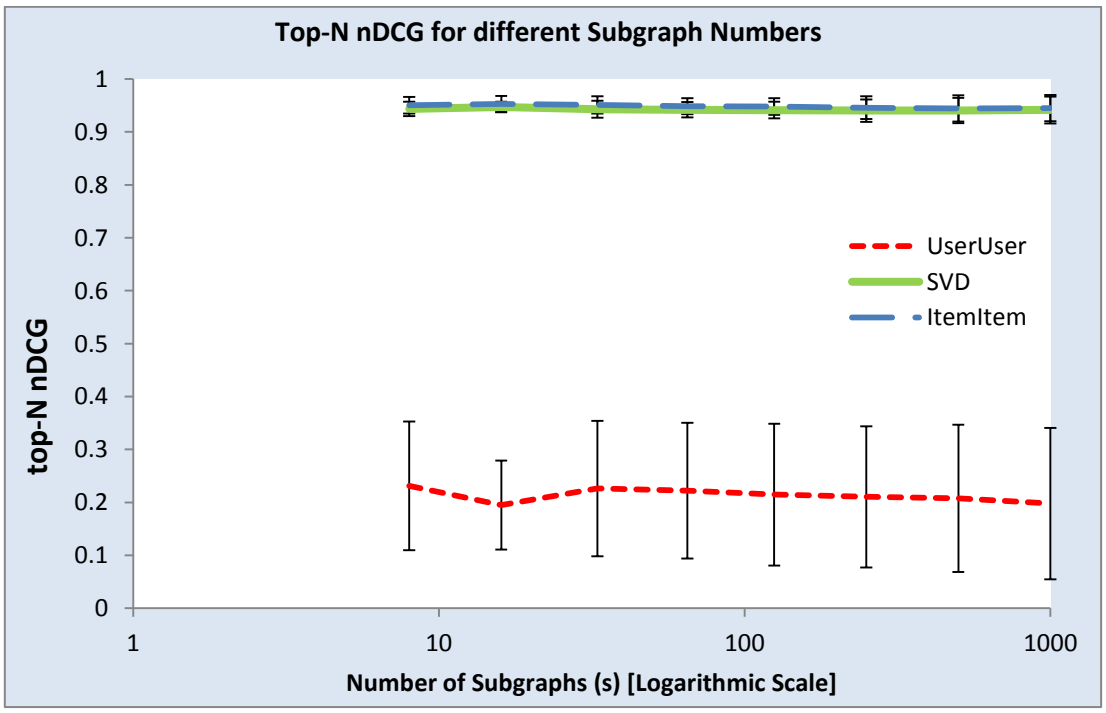
Ένα σημαντικό αποτέλεσμα που προκύπτει, είναι ότι με αυτό τον τρόπο, το μοντέλο μας μπορεί να δώσει λύση σε ένα σημαντικό πρόβλημα των συστημάτων συστάσεων, γνωστό και ως Cold-Start problem. Ακόμα λοιπόν και στην περίπτωση νέων χρηστών, για τους οποίους το σύστημα δεν έχει ακόμα επαρκείς πληροφορίες για να προτείνει αντικείμενα, το σύστημα το οποίο προτείνει αυτή η έρευνα μπορεί κατατάσσοντας τον χρήστη σε έναν υπογράφο και με τη χρήση του αλγόριθμου SVD να του προτείνει πιθανά αντικείμενα με αρκετά καλή ακρίβεια πρόβλεψης.

Σε συνέχεια της επανάληψης των πειραμάτων που έγιναν για τους διάφορους υπογράφους στο σύνολο δεδομένων του Flixster, στα Γραφήματα που ακολουθούν, παραθέτονται τα αποτελέσματα των διάφορων μετρικών για τα διάφορα μεγέθη των υπογράφων.

Έτσι, από τις Εικόνες 24 & 25 για την απόδοση κατά nDCG και Top-N nDCG, επαληθεύεται ότι η συμπεριφορά του μοντέλου είναι ανεξάρτητη από το σύνολο δεδομένων, καθώς τα αποτελέσματα είναι συγκρίσιμα με αυτά για το Epinions που παρουσιάστηκαν νωρίτερα.



Εικόνα 24: Απόδοση κατά nDCG όλων των αλγορίθμων στο Flixster για τους διαφορετικούς υπογράφους (s)

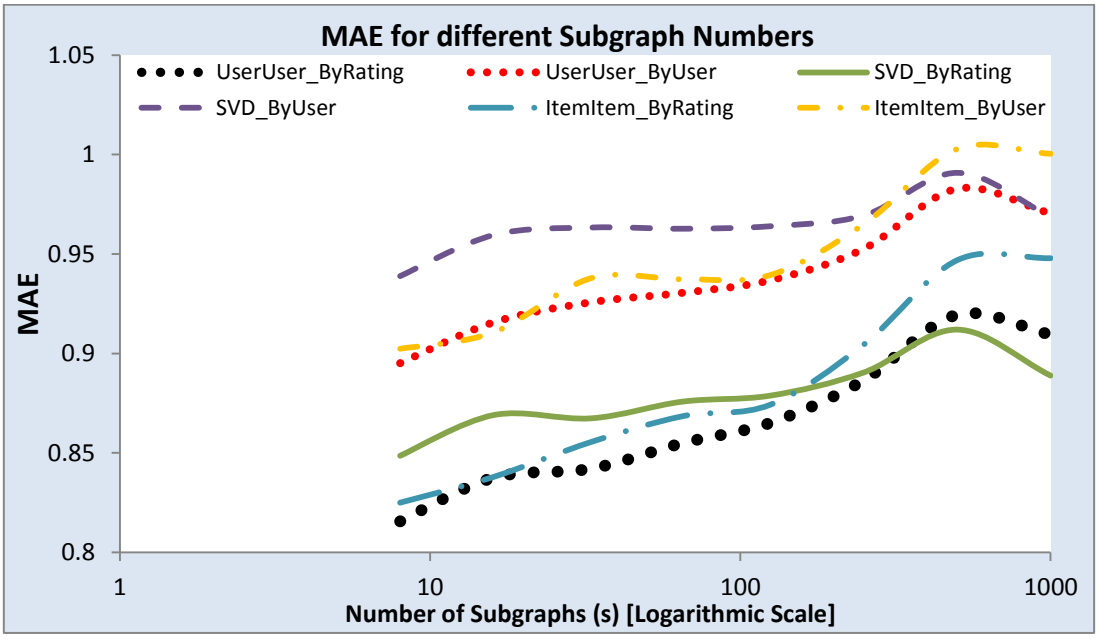


Εικόνα 25: Απόδοση κατά Top-N nDCG όλων των αλγορίθμων στο Flixster για τους διαφορετικούς υπογράφους (s)

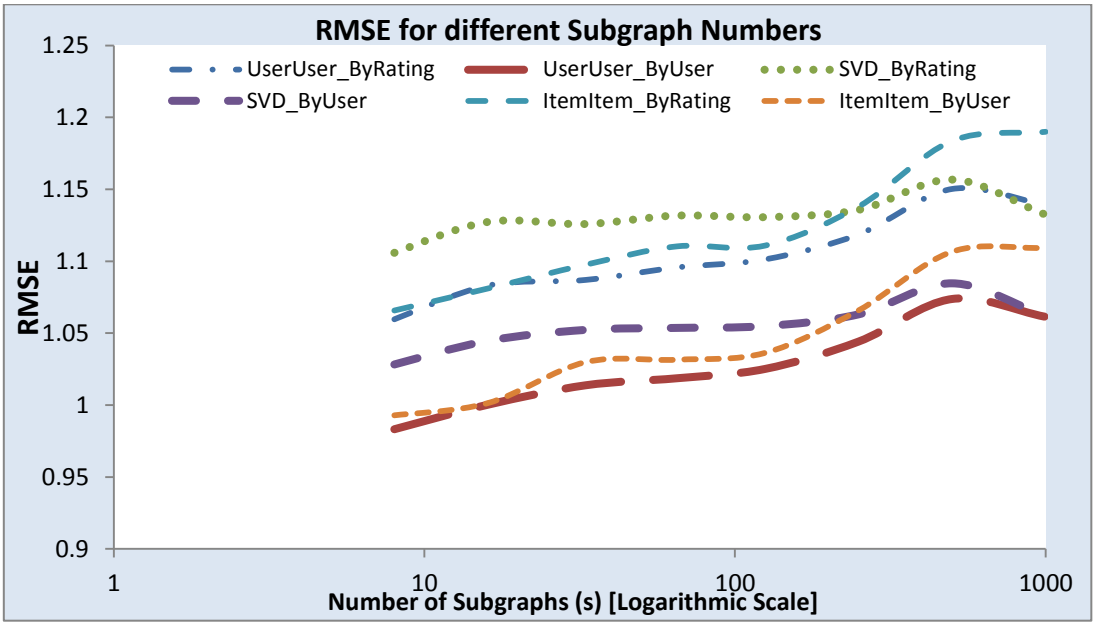
Και πάλι μπορούμε να παρατηρήσουμε, ότι ο αλγόριθμος User-User δεν αποδίδει καλά για τα top-N αντικείμενα (Εικόνα 25), παρόλο που σαν κέρδος στο σύνολο των βαθμολογιών (Εικόνα 24) προσεγγίζει σε απόδοση τους Item-Item και SVD (με μέγιστη απόκλιση 0,006).

Επίσης, από τις Εικόνες 26 και 27 που ακολουθούν για την απόδοση κατά RMSE και MAE αντίστοιχα, γίνεται εμφανής ο υπολογισμός προβλέψεων για τα διαφορετικά sub-graphs με σφάλμα πρόβλεψης $\sim 0{,}84 - 1{,}1$ μονάδες. Να θυμίσουμε ότι λόγω της μη κανονικοποίησης στις μετρικές MAE και RMSE, είναι πιθανή η εμφάνιση σφαλμάτων ίσως και λίγο μεγαλύτερα της μονάδας, κάτι το οποίο είναι αποδεκτό αν σκεφτούμε ότι το εύρος τιμών των βαθμολογιών είναι [1..5].

Επίσης, όπως φαίνεται σε όλα τα διαγράμματα απόδοσης του συστήματος για το Flixster, να σημειωθεί, πως λόγω της δομής του γράφου αξιολογήσεων, δεν ήταν δυνατή η εκτέλεση των αλγορίθμων για λιγότερους από οκτώ υπογράφους και φυσικά ούτε σε ολόκληρο τον γράφο. Το γεγονός αυτό, επιβεβαιώνει την χρησιμότητα και την αποδοτικότητα της μεθόδου πρόβλεψης συστάσεων σε διχοτομημένους γράφους, ακόμα και για σύνολα δεδομένων με ιδιαίτερη δομή.



Εικόνα 26: Απόδοση κατά MAE όλων των αλγορίθμων στο Flixster για τους διαφορετικούς υπογράφους (s)



Εικόνα 27: Απόδοση κατά Top-N nDCG όλων των αλγορίθμων στο Flixster για τους διαφορετικούς υπογράφους (s)

4.4.2 Συγκριτικά Αποτελέσματα

Λαμβάνοντας υπόψη τα παραπάνω αποτελέσματα εκτέλεσης της διαδικασίας για τους διαφορετικούς αριθμούς υπογράφων (s), ας δούμε τους παρακάτω χρήσιμους συγκριτικούς πίνακες κυριαρχίας (Dominance Matrices) των αλγορίθμων, οι οποίοι συνοψίζουν την απόδοση τους για κάθε μία από τις πέντε μετρικές που χρησιμοποιήθηκαν.

Epinions Dominance Matrices

RMSE_ByRating — Dominance Matrix (Epinions)

	UserUser	ItemItem	FunkSVD
UserUser	0	65	41
ItemItem	0	0	0
FunkSVD	24	65	0

RMSE_ByUser — Dominance Matrix (Epinions)

	UserUser	ItemItem	FunkSVD
UserUser	0	53	6
ItemItem	12	0	7
FunkSVD	59	58	0

MAE_ByRating — Dominance Matrix (Epinions)

	UserUser	ItemItem	FunkSVD
UserUser	0	42	0
ItemItem	23	0	0
FunkSVD	65	65	0

MAE_ByUser — Dominance Matrix (Epinions)

	UserUser	ItemItem	FunkSVD
UserUser	0	5	0
ItemItem	60	0	34
FunkSVD	65	31	0

Top-N nDCG — Dominance Matrix (Epinions)

	UserUser	ItemItem	FunkSVD
UserUser	0	0	0
ItemItem	65	0	63
FunkSVD	65	2	0

: Dominant Algorithm

Πίνακας 3: Dominance Matrices for all three algorithms in Epinions Dataset

Οι παραπάνω πίνακες απεικονίζουν, για την περίπτωση όπου s=65, την σύγκριση της απόδοσης του καθενός από τους τρείς αλγορίθμους για κάθε μία από τις αντίστοιχες μετρικές στο σύνολο δεδομένων του Epinions και για κάθε έναν από τους 65 υπογράφους. Έτσι, όπως προκύπτει από τα συγκριτικά, ο αλγόριθμος Funk_SVD βρέθηκε καλύτερος από τους Item-Item & User-User, κερδίζοντάς τους σε 59 και 58 από τους 65 υπογράφους αντίστοιχα για την μετρική **RMSE_ByUser** και σε όλους τους υπογράφους για την μετρική **MAE_ByRating**. Ομοίως, ο αλγόριθμος Item-Item απέδωσε καλύτερα από τους Funk_SVD & User-User για τις μετρικές **MAE_ByUser** και **Top-N nDCG**. Τέλος, ο αλγόριθμος User-User απέδωσε καλύτερα από τους άλλους δύο κατά τη μετρική **RMSE_ByRating**, επικρατώντας σε όλους τους υπογράφους του Item-Item και σε 41 από τους 65 του SVD, ενώ ο Funk_SVD ήταν σε όλους τους υπογράφους καλύτερος από τον Item-Item.

Ομοίως, μπορούμε να δούμε τους πίνακες σύγκρισης επικράτησης των αλγορίθμων για το σύνολο δεδομένων του Flixster (Πίνακας 4).

Flixster Dominance Matrices

RMSE_ByRating — Dominance Matrix (Flixster)

	UserUser	ItemItem	FunkSVD
UserUser	0	43	60
ItemItem	22	0	57
FunkSVD	15	8	0

RMSE_ByUser — Dominance Matrix (Flixster)

	UserUser	ItemItem	FunkSVD
UserUser	0	51	60
ItemItem	14	0	59
FunkSVD	5	6	0

MAE_ByRating — Dominance Matrix (Flixster)

	UserUser	ItemItem	FunkSVD
UserUser	0	53	60
ItemItem	12	0	55
FunkSVD	5	10	0

MAE_ByUser — Dominance Matrix (Flixster)

	UserUser	ItemItem	FunkSVD
UserUser	0	39	60
ItemItem	26	0	59
FunkSVD	5	6	0

Top-N nDCG — Dominance Matrix (Flixster)

	UserUser	ItemItem	FunkSVD
UserUser	0	0	0
ItemItem	65	0	59
FunkSVD	65	6	0

: Dominant Algorithm

Πίνακας 4: Dominance Matrices for all three algorithms in Flixster Dataset

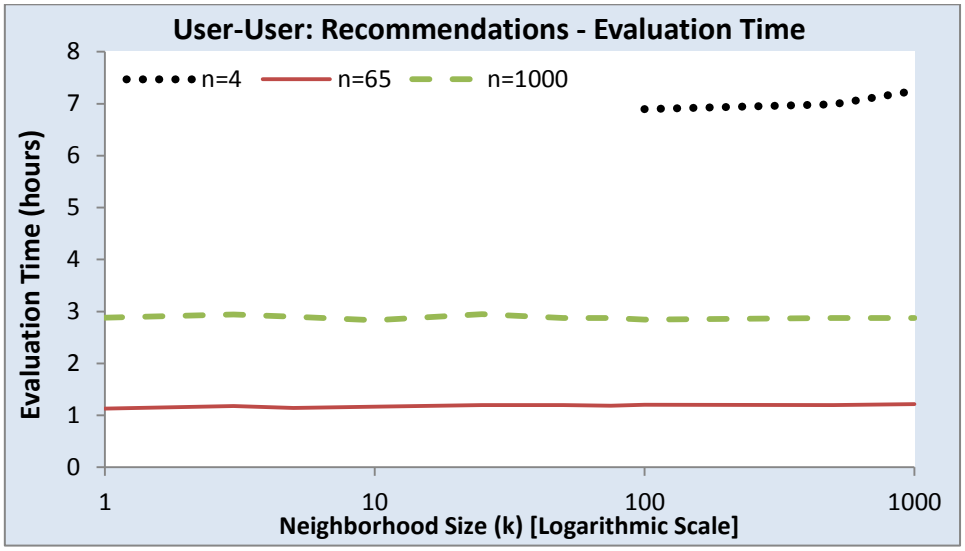
Αν και οι διαφορές μεταξύ των αποτελεσμάτων των 65 υπογράφων ήταν της τάξης του δεύτερου δεκαδικού, όπως παρατηρούμε από τον Πίνακα 4, ο

αλγόριθμος User-User έχει επικρατήσει των αλγορίθμων Item-Item και Funk_SVD σε τέσσερις από τις πέντε μετρικές, ενώ στην μετρική **Top-N nDCG** έχει επικρατήσει ο αλγόριθμος Item-Item, επικρατώντας σε όλους τους υπογράφους του User-User και σε 59 από τους 65 του Funk_SVD.

4.4.3 Αποτελέσματα για διαφορετικό αριθμό πλησιέστερων γειτόνων – knn

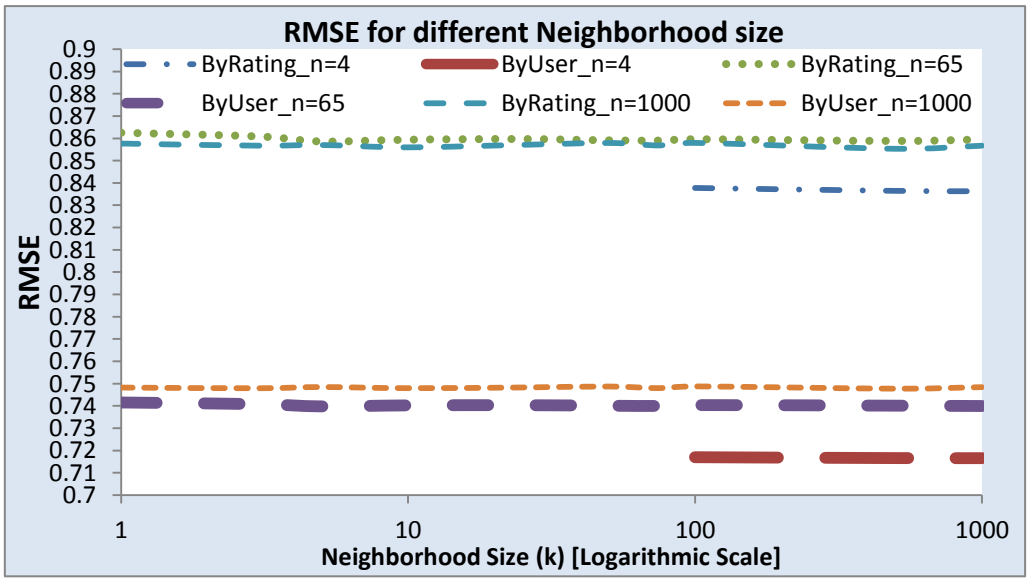
Έχοντας διαπιστώσει την καλύτερη απόδοση του συστήματος στους 65 υπογράφους, ξεκινήσαμε την ανάλυση της απόδοσης για τις διάφορες τιμές του μεγέθους της γειτονιάς (knn) του κάθε κόμβου για τους 65 υπογράφους στο σύνολο δεδομένων του Epinions και στην συνέχεια εξετάσαμε την απόδοση για $s = 4$ και $s = 1000$ για να διαπιστώσουμε και την συμπεριφορά του συστήματος γύρω από την τιμή των 65 υπογράφων.

Παρατηρώντας την Εικόνα 28 για τον χρόνο εκτέλεσης της αξιολόγησης για τις διάφορες τιμές του k, διακρίνουμε καταρχήν ότι η εκτέλεση για οποιοδήποτε μέγεθος γειτονιάς είναι γρηγορότερη στους 65 υπογράφους σε σχέση με τους 4 και 1000 υπογράφους, γεγονός το οποίο επιβεβαιώνει και το αποτέλεσμα σχετικά με το χρόνο εκτέλεσης από την προηγούμενη σειρά πειραμάτων. Δευτερευόντως, μπορούμε να συμπεράνουμε ότι για κάθε μία από τις τρεις τιμές του s ο χρόνος εκτέλεσης παρουσιάζει ελάχιστες διακυμάνσεις, με μικρή εξαίρεση στους 4 υπογράφους και για πάνω από 500 κοντινότερους γείτονες, όπου παρουσιάζεται μία μικρή αύξηση στο χρόνο εκτέλεσης. Από το παραπάνω γεγονός, μπορούμε να συμπεράνουμε ότι ο χρόνος εκτέλεσης του αλγόριθμου User-User είναι ανεξάρτητος από το μέγεθος των k-κοντινότερων γειτόνων. Να σημειωθεί, ότι λόγω του αυξημένου χρόνου εκτέλεσης της διαδικασίας για τους 4 υπογράφους, κρίθηκε σκόπιμο η εξέταση μόνο των 100, 500 και 1000 k-κοντινότερων γειτόνων.



Εικόνα 28: Χρόνος εκτέλεσης της αξιολόγησης για διαφορετικό Neighborhood Size και s={4, 65,1000} στο Epinions

Όσον αφορά την απόδοση του συστήματος για τις διαφορετικές τιμές του k, μετρώντας την με την μετρική RMSE η οποία απεικονίζεται στην Εικόνα 29, παρατηρούμε ότι η καλύτερη απόδοση επιτυγχάνεται στην μετρική RMSE_ByUser, στους 4 υπογράφους με σφάλμα πρόβλεψης της τάξης του 0,71. Ωστόσο, πολύ κοντά είναι και οι μετρήσεις της απόδοσης για τους 65 και 1000 υπογράφους κατά την RMSE_ByUser, με διαφορά κατά $\sim 0{,}03 - 0{,}04$ σε σχέση με τα 4 sub-graphs.



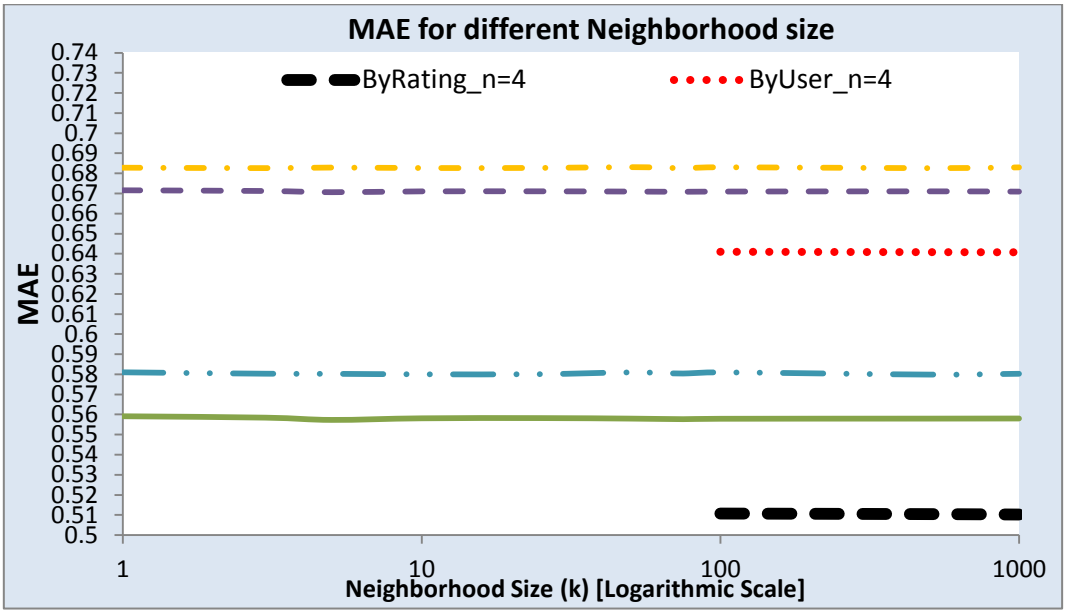
Εικόνα 29: Απόδοση κατά RMSE για διαφορετικό Neighborhood Size (k)

Σε κάθε περίπτωση, μπορούμε να συμπεράνουμε από την Εικόνα 29, ότι η διακύμανση της απόδοσης του συστήματος σύμφωνα με τον αλγόριθμο User-User για τις διάφορες τιμές του k είναι ελάχιστη. Θα μπορούσαμε λοιπόν να πούμε ότι

μετρώντας την απόδοση του μοντέλου κατά RMSE, η αλλαγή του πλήθους των πλησιέστερων γειτόνων στον αλγόριθμο User-User δεν επηρεάζει ούτε αρνητικά ούτε θετικά την απόδοση του. Εν μέρει, αυτό οφείλεται στην προσπάθεια του αλγορίθμου να εντοπίσει τους k ομοιότερους χρήστες μέσα σε έναν ήδη διχοτομημένο γράφο και μάλιστα όχι τυχαία, αλλά με βάση τον κοινωνικό γράφο των χρηστών.

Αυτό πρακτικά σημαίνει δύο πράγματα. Καταρχήν, εφόσον οι τιμές του RMSE είναι σχετικά ίδιες για τα διάφορα k, τότε οι χρήστες που έχουν προκύψει από το αρχικό μοίρασμα του γράφου στους υπογράφους είναι αρκετά όμοιοι και κατά συνέπεια το αποτέλεσμα δεν διαφέρει σημαντικά είτε επιλέξουμε τους 10 είτε τους 100 κοντινότερους γείτονες για την πρόταση συστάσεων. Δευτερευόντως, οφείλεται στον τρόπο δημιουργίας των συστάσεων από τον αλγόριθμο σε συνδυασμό με την κατανομή των αξιολογήσεων στους υπογράφους. Ουσιαστικά δηλαδή, ο αλγόριθμος User_User βρίσκει τον πιο όμοιο χρήστη από τον οποίο θα προκύψουν και τα προτεινόμενα αντικείμενα, αναζητώντας μέσα στον υπογράφο και για k κοντινότερους γείτονες. Στην πράξη όμως για όλα τα k που εξετάστηκαν, ο πιο κοντινός με τον υπό εξέταση χρήστη γείτονας βρίσκεται ήδη μέσα στο neighborhood του χρήστη και κατά συνέπεια δεν επηρεάζεται η ποιότητα των προβλέψεων ανεξάρτητα με το μέγεθος της γειτονιάς στην οποία θα αναζητήσουμε.

Επίσης, σχετικά με την διαδικασία αξιολόγησης του μοντέλου μπορούμε να παρατηρήσουμε ότι ο υπολογισμός του RMSE ανά χρήστη και η λήψη στη συνέχεια του συνολικού RMSE είναι αποδοτικότερη σε σχέση με τον υπολογισμό του για κάθε αξιολόγηση ξεχωριστά και τον υπολογισμό στη συνέχεια του συνολικού αποτελέσματος της μετρικής, το οποίο και προσθέτει μια διαφορά στο συνολικό σφάλμα της τάξης του 0,15 σε σχέση με το RMSE_ByUser.

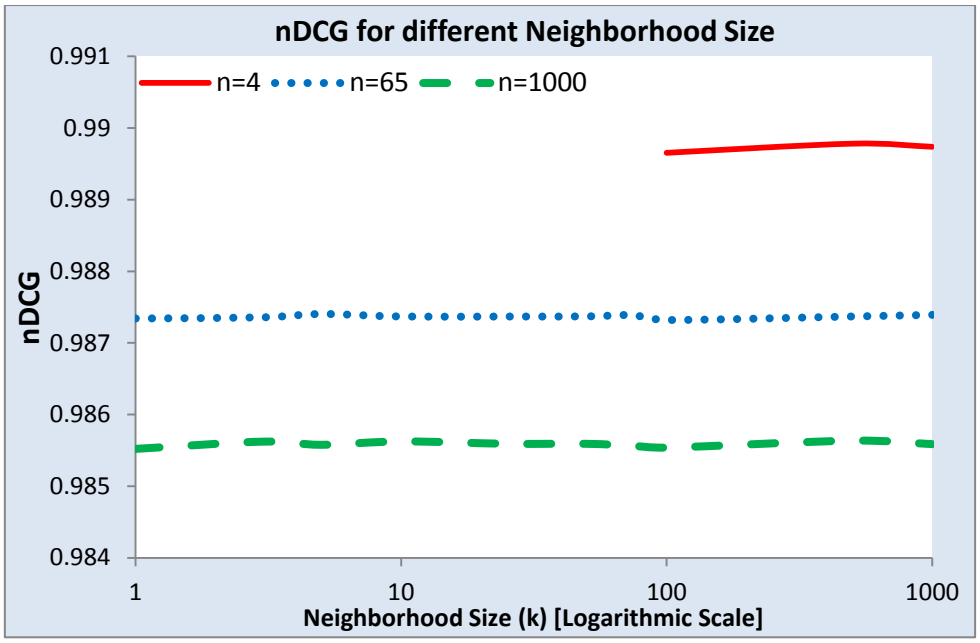


Εικόνα 30: Απόδοση κατά MAE για διαφορετικό Neighborhood Size (k)

Η Εικόνα 30, στην οποία φαίνεται ο υπολογισμός της απόδοσης του μοντέλου κατά MAE, επιβεβαιώνει τα συμπεράσματα που προέκυψαν από τον υπολογισμό της απόδοσης κατά RMSE, καθώς η συμπεριφορά του αλγορίθμου και για τις δύο μετρικές είναι ανάλογη. Ωστόσο, όπως προκύπτει από την σύγκριση των δύο μετρικών, η αξιολόγηση της εκτέλεσης του αλγόριθμου κατά MAE δίνει χαμηλότερες τιμές σφάλματος έως και κατά 0,35 μονάδες, το οποίο κρίνεται σημαντικό δεδομένου της απόκλισης των αποτελεσμάτων και του εύρους της κλίμακας αξιολογήσεων [1-5]. Πιο συγκεκριμένα, μπορούμε να δούμε ότι για του 4 υπογράφους και χρησιμοποιώντας τους 100 πλησιέστερους γείτονες, η μετρική RMSE_ByRating προβλέπει ένα σφάλμα της τάξης των 0,84 μονάδων, ενώ στον αντίστοιχο υπολογισμό κατά MAE η πρόβλεψη είναι με σφάλμα 0,51 μονάδων. Αντίστοιχα είναι τα σφάλματα και για τις υπόλοιπες τιμές του k, το οποίο επιβεβαιώνει το γεγονός ότι λόγω του αρχικού partitioning των χρηστών με βάση τον κοινωνικό τους γράφο, η απόδοση του αλγορίθμου είναι ανεξάρτητη από το πλήθος των κοντινότερων γειτόνων και για τις τρεις τιμές του s.

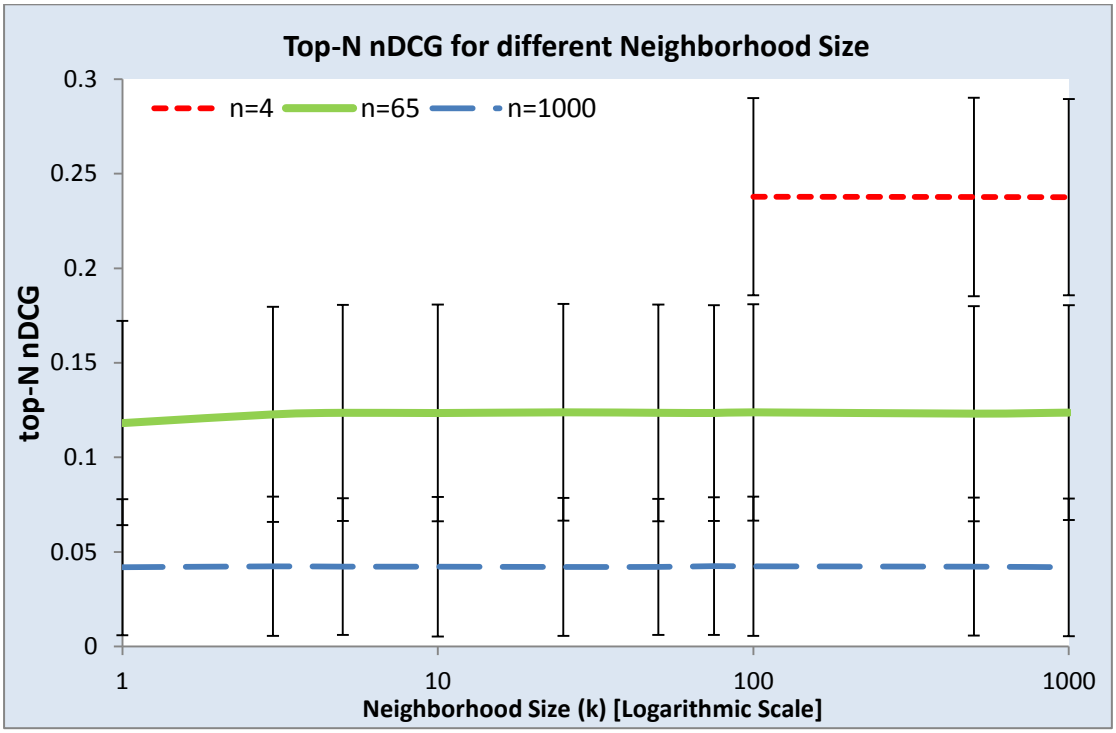
Παρ’ όλα αυτά, υπολογίζοντας την απόδοση του συστήματος σύμφωνα με την μετρική nDCG, όπως φαίνεται και στην Εικόνα 31, εξακολουθούμε να παρατηρούμε την ανεξαρτησία των προβλέψεων από την αλλαγή των k κοντινότερων γειτόνων, θα μπορούσαμε να πούμε ωστόσο ότι η απόδοση των προβλέψεων του αλγόριθμου και στις τρεις περιπτώσεις των υπογράφων ήταν

ιδανική, καθώς το Gain, το οποίο να θυμίσουμε πως είναι της κλίμακας [0,1] με την ιδανική περίπτωση να είναι για nDCG=1, στην χειρότερη των περιπτώσεων ήταν για $n = 1000$ υπογράφους ίσο με 0,985. Φυσικά οι διαφορές μεταξύ των s διαφορετικών υπογράφων για όλες τις περιπτώσεις των k γειτόνων είναι εξαιρετικά μικρές και της τάξης του 0,02 – 0,04.



Εικόνα 31: Απόδοση κατά nDCG για διαφορετικό Neighborhood Size (k)

Σε αντίθεση όμως με την nDCG, η εκτέλεση της αξιολόγησης ως προς τη μετρική Top-N nDCG, που φαίνεται στην Εικόνα 32, είχε εκ διαμέτρου αντίθετα αποτελέσματα. Συγκεκριμένα, η μέγιστη απόδοση που υπολογίστηκε από την μετρική ήταν για τους τέσσερεις υπογράφους και ίση με 0,24. Βέβαια, με βάση τον τρόπο υπολογισμού του Top-N nDCG, η πρόβλεψη με τέτοιο σφάλμα είναι λογική, κι αυτό γιατί η μετρική υπολογίζεται πάνω σε μία λίστα με τα TopN αντικείμενα. Κατά συνέπεια, αφού όπως είπαμε προηγουμένως λόγω του αρχικού διαμοιρασμού των χρηστών σε υπογράφους με βάση τον κοινωνικό τους γράφο είναι πιθανό η λίστα των TopN αντικειμένων να μην περιλαμβάνει όμοια με τον υπό εξέταση χρήστη προϊόντα, και συνεπώς το υπολογιζόμενο κέρδος για αυτή τη λίστα προϊόντων είναι πολύ μικρό.



Εικόνα 32: Απόδοση κατά Top-N nDCG για διαφορετικό Neighborhood Size (k)

5 Συμπεράσματα

Συνοψίζοντας τα αποτελέσματα της παρούσας μελέτης, συμπερασματικά προκύπτει πως η προσέγγιση και το μοντέλο το οποίο προτείνεται από αυτήν την έρευνα, μπορεί να αποδώσει ακόμα και καλύτερα από τις κλασικές τεχνικές παραγωγής συστάσεων. Αυτό εξαρτάται κυρίως από την δομή του γράφου αξιολογήσεων (κατά πόσο δηλαδή είναι sparse ο bipartite γράφος) για τον οποίο θα γίνει η παραγωγή συστάσεων και μπορούμε να πούμε πως είναι το μόνο μειονέκτημα αυτής της μεθόδου. Παρόλα αυτά όμως με την χρήση αλγορίθμων όπως ο SVD, μπορούμε να δώσουμε προτάσεις ακόμα και σε sparse γράφους όπως αυτός του Flixster.

Αξιοσημείωτο συμπέρασμα είναι η μείωση του χρόνου υπολογισμού των συστάσεων με την εκτέλεση της διαδικασίας Collaborative Filtering σε τμηματοποιημένους με βάση τον κοινωνικό τους γράφο υπογράφους. Με αυτόν τον τρόπο, θα μπορούσαμε να κατανείμουμε την εργασία παραγωγής συστάσεων σε επιμέρους, σε ένα κατανεμημένο υπολογιστικό σύστημα και να ελαχιστοποιηθεί ο χρόνος υπολογισμού, διατηρώντας στο ίδιο επίπεδο τα αποτελέσματα των προβλέψεων.

Ένα ακόμα σημαντικό συμπέρασμα της ανάλυσης είναι η δυνατότητα παροχής συστάσεων ακόμα και σε χρήστες που δεν έχουν κάνει καμία αξιολόγηση, γνωστό και ως Cold-Start πρόβλημα, αξιοποιώντας τις πληροφορίες του κοινωνικού γράφου των χρηστών. Αυτό το συμπέρασμα αποδεικνύει πλήρως την χρησιμότητα της εκμετάλλευσης και των κοινωνικών γράφων μαζί με τους γράφους αξιολογήσεων για την παραγωγή εξατομικευμένων συστάσεων στους χρήστες, το οποίο επαληθεύει και τον απώτερο ερευνητικό στόχο αυτής της διπλωματικής εργασίας.

6 Μελλοντικές Εργασίες

Από τα παραπάνω πειράματα, προκύπτουν αρκετά ζητήματα προς εξέταση και τα οποία αποτελούν κίνητρο για περαιτέρω επέκταση της παρούσας έρευνας. Αρχικά, όπως αναφέρθηκε, ο τύπος των κοινωνικών γράφων εμπιστοσύνης/δυσπιστίας ήταν κατευθυνόμενος αλλά για λόγους partitioning θεωρήθηκαν ως μη κατευθυνόμενοι. Στη συνέχεια λοιπόν της έρευνας, πρέπει να ελεγχθεί η συμπεριφορά του μοντέλου αν χρησιμοποιηθεί και η πληροφορία της δυσπιστίας στις σχέσεις φιλίας και κατά πόσον αυτό μπορεί να επηρεάσει θετικά ή αρνητικά την απόδοση του μοντέλου.

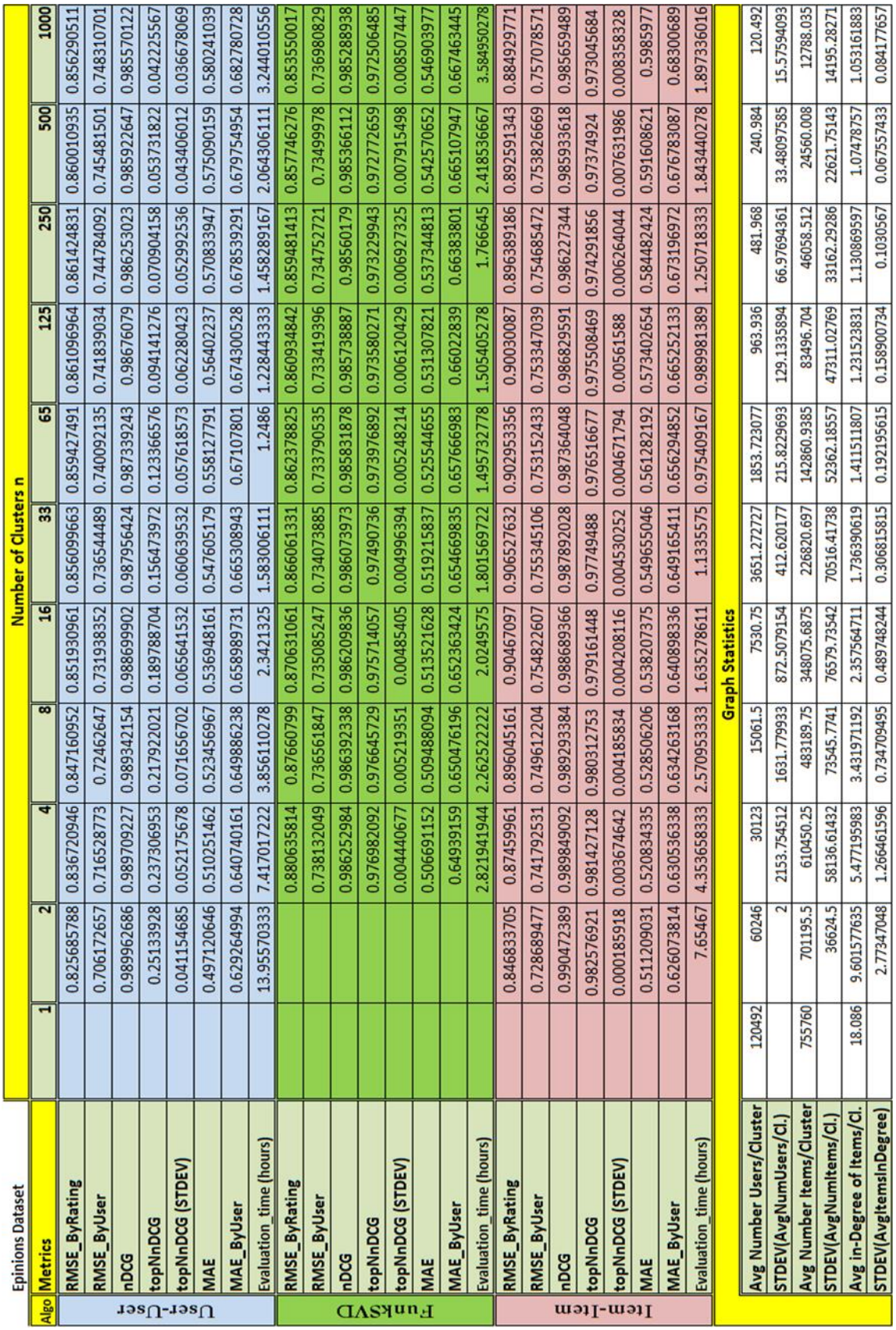
Επίσης, κρίνεται σκόπιμο, να γίνει περαιτέρω έλεγχος της επιρροής των CrossCluster Edges, δηλαδή των γειτονικών ακμών από τους γύρω υπογράφους, στην απόδοση του συστήματος, καθώς συμπεριλαμβάνοντας και κοντινές ακμές από τους γύρω υπογράφους, προσθέτουμε και επιπλέον αξιολογήσεις προς εξέταση.

Επιπροσθέτως, θα πρέπει να ερευνήσουμε τί θα συμβεί, κυρίως σε γράφους όπως το Flixster που έχουν περισσότερες διαφορετικές βαθμολογίες ανά υπογράφο, αν αυξήσουμε τον αριθμό των top-N αντικειμένων στα οποία κάνει evaluate η μετρική Top-N nDCG. Βεβαίως, αυτό μπορεί να μην επηρεάσει ιδιαίτερα την απόδοση στο Epinions, θα μας δώσει όμως μία καλή εικόνα για το τί μπορεί να συμβεί σε ιδιαίτερους γράφους με την μορφή του Flixster.

Τέλος, άλλο ένα σενάριο που περιλαμβάνεται στα μελλοντικά σχέδια της παρούσας μελέτης, είναι η ανάλυση αν και πώς θα μπορούσε να αλλάξει την συμπεριφορά του συστήματος η εφαρμογή αλγορίθμων όπως ο PageRank πάνω στις εσωτερικές ακμές κάθε υπογράφου.

7 Παραρτήματα

7.1 Παράρτημα I (Πίνακες Αποτελεσμάτων Εκτέλεσης των δύο datasets)



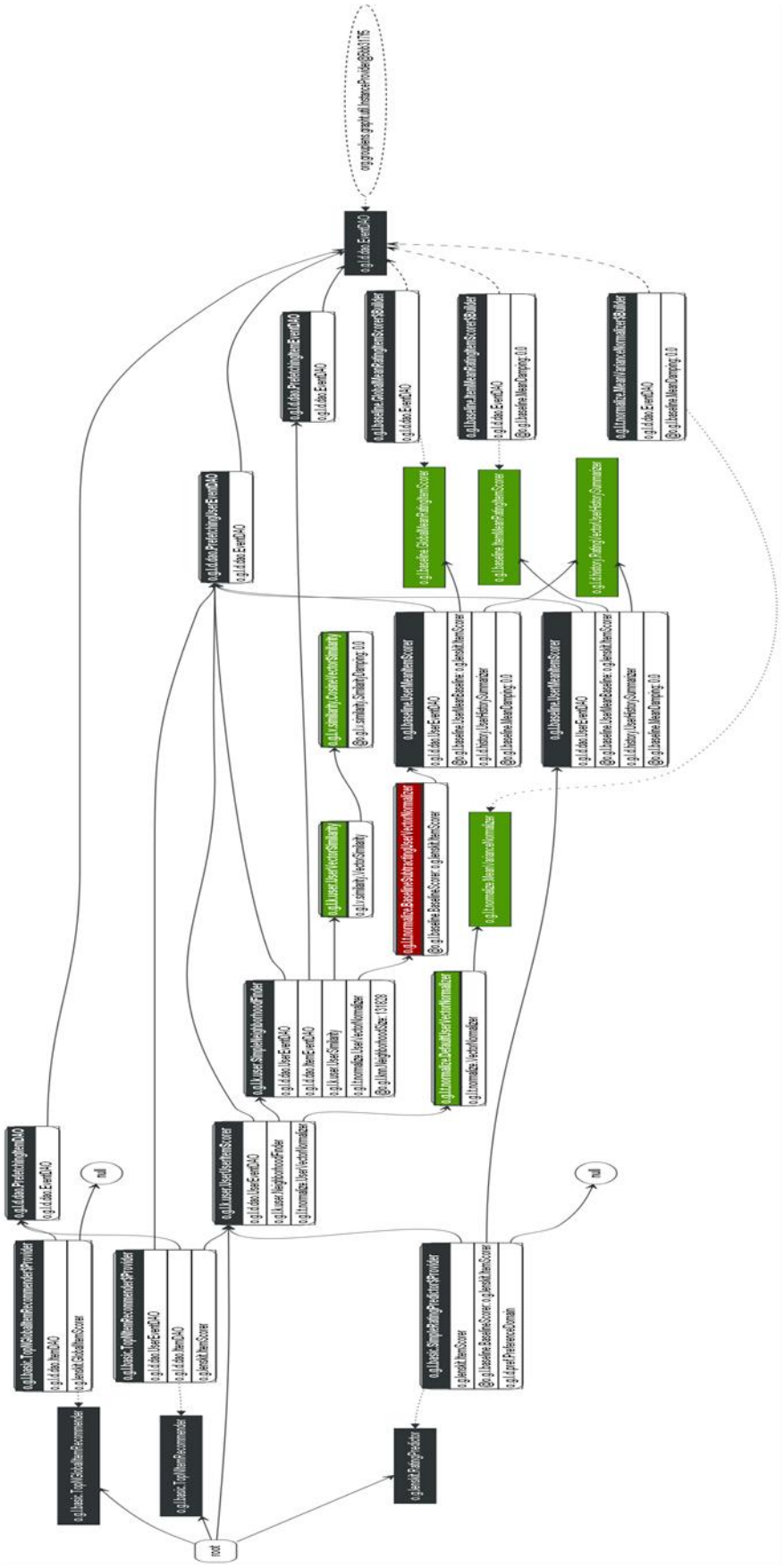
Epinions Dataset

Algo	Metrics	Number of Clusters n											
		1	2	4	8	16	33	65	125	250	500	1000
User-User	RMSE_ByRating		0.825685788	0.836720946	0.847160952	0.851930961	0.856099663	0.859427491	0.861096964	0.861424831	0.860010935	0.856290511
	RMSE_ByUser		0.706172657	0.716528773	0.72462647	0.731938352	0.736544489	0.740092135	0.741839034	0.744784092	0.745481501	0.748310701
	nDCG		0.989962686	0.989709227	0.989342154	0.988699902	0.987956424	0.987339243	0.98676079	0.986253023	0.985922647	0.985570122
	topNnDCG		0.25133928	0.237306953	0.217922021	0.189788704	0.156473972	0.123366576	0.094141276	0.070904158	0.053731822	0.042225567
	topNnDCG (STDEV)		0.041154685	0.052175678	0.071656702	0.065641532	0.060639532	0.057618573	0.062280423	0.052992536	0.043406012	0.036678069
	MAE		0.497120646	0.510251462	0.523456967	0.536948161	0.547605179	0.558127791	0.56402237	0.570833947	0.575090159	0.580241039
	MAE_ByUser		0.629264994	0.640740161	0.649886238	0.658989731	0.665308943	0.67107801	0.674300528	0.678539291	0.679754954	0.682780728
	Evaluation_time (hours)		13.95570333	7.417017222	3.856110278	2.3421325	1.583006111	1.2486	1.228443333	1.458289167	2.064306111	3.244010556
FunkSVD	RMSE_ByRating			0.880635814	0.87660799	0.870631061	0.866061331	0.862378825	0.860934842	0.859481413	0.857746276	0.853550017
	RMSE_ByUser			0.738132049	0.736561847	0.735085247	0.734073885	0.733790535	0.733419396	0.734752721	0.73499978	0.736980829
	nDCG			0.986252984	0.986392338	0.986209836	0.986073973	0.985831878	0.985738887	0.98560179	0.985366112	0.985288938
	topNnDCG			0.976982092	0.976645729	0.975714057	0.97490736	0.973976892	0.973580271	0.973229943	0.972772659	0.972506485
	topNnDCG (STDEV)			0.004440677	0.005219351	0.00485405	0.004996394	0.005248214	0.006120429	0.006927325	0.007915498	0.008507447
	MAE			0.506691152	0.509488094	0.513521628	0.519215837	0.525544655	0.531307821	0.537344813	0.542570652	0.546903977
	MAE_ByUser			0.64939159	0.650476196	0.652363424	0.654669835	0.657666983	0.66022839	0.66383801	0.665107947	0.667463445
	Evaluation_time (hours)			2.821941944	2.262522222	2.0249575	1.801569722	1.495732778	1.505405278	1.766645	2.418536667	3.584950278
Item-Item	RMSE_ByRating		0.846833705	0.87459961	0.896045161	0.90467097	0.906527632	0.902953356	0.90030087	0.896389186	0.892591343	0.884929771
	RMSE_ByUser		0.728689477	0.741792531	0.749612204	0.754822607	0.755345106	0.753152433	0.753347039	0.754685472	0.753826669	0.757078571
	nDCG		0.990472389	0.989849092	0.989293384	0.988689366	0.987892028	0.987364048	0.986829591	0.986227344	0.985933618	0.985659489
	topNnDCG		0.982576921	0.981427128	0.980312753	0.979161448	0.97749488	0.976516677	0.975508469	0.974291856	0.97374924	0.973045684
	topNnDCG (STDEV)		0.000185918	0.003674642	0.004185834	0.004208116	0.004530252	0.004671794	0.00561588	0.006264044	0.007631986	0.008358328
	MAE		0.511209031	0.520834335	0.528506206	0.538207375	0.549655046	0.561282192	0.573402654	0.584482424	0.591608621	0.5985977
	MAE_ByUser		0.626073814	0.630536338	0.634263168	0.640898336	0.649165411	0.656294852	0.665252133	0.673196972	0.676783087	0.68300689
	Evaluation_time (hours)		7.65467	4.353658333	2.570953333	1.635278611	1.1335575	0.975409167	0.989981389	1.250718333	1.843440278	1.897336016
Graph Statistics												
	Avg Number Users/Cluster	120492	60246	30123	15061.5	7530.75	3651.272727	1853.723077	963.936	481.968	240.984	120.492
	STDEV(AvgNumUsers/Cl.)		2	2153.754512	1631.779933	872.5079154	412.620177	215.8229693	129.1335894	66.97694361	33.48097585	15.57594093
	Avg Number Items/Cluster	755760	701195.5	610450.25	483189.75	348075.6875	226820.697	142860.9385	83496.704	46058.512	24560.008	12788.035
	STDEV(AvgNumItems/Cl.)		36624.5	58136.61432	73545.7741	76579.73542	70516.41738	52362.18557	47311.02769	33162.29286	22621.75143	14195.28271
	Avg in-Degree of Items/Cl.	18.086	9.601577635	5.477195983	3.431971192	2.357564711	1.736390619	1.411511807	1.231523831	1.130869597	1.07478757	1.053161883
	STDEV(AvgItemsInDegree)		2.77347048	1.266461596	0.734709495	0.489748244	0.306815815	0.192195615	0.158900734	0.1030567	0.067557433	0.084177657

Flixster Dataset

Algo	Metrics	Number of Clusters n: 1	2	4	8	16	33	65	125	250	500	1000
User-User	RMSE_ByRating				1.059632172	1.083080904	1.08702377	1.095730356	1.10130236	1.118654369	1.150266052	1.139200282
	RMSE_ByUser				0.98320768	1.000247698	1.013753145	1.018656673	1.025153451	1.04416715	1.073768282	1.06135127
	nDCG				0.972750628	0.973941022	0.971892495	0.971369451	0.970487857	0.969326096	0.968359223	0.968370701
	topNnDCG				0.231388099	0.194809996	0.226119594	0.222190217	0.214680202	0.210401171	0.207558744	0.197847879
	topNnDCG (STDEV)				0.121622543	0.084124991	0.127829969	0.128347827	0.134110891	0.133414367	0.138918392	0.143061732
	MAE				0.815613349	0.837547038	0.842105147	0.854924447	0.865141489	0.886564489	0.919494257	0.90955679
	MAE_ByUser				0.895119015	0.915655041	0.925740984	0.93049425	0.936820453	0.952634075	0.982825706	0.970457255
FunkSVD	RMSE_ByRating				1.10585687	1.127334262	1.12588701	1.131775549	1.130656274	1.135743062	1.15675144	1.132386272
	RMSE_ByUser				1.028245498	1.044676598	1.052276158	1.053670728	1.054936983	1.063090696	1.084652056	1.061845264
	nDCG				0.966094931	0.967340581	0.965549711	0.9656973	0.964798716	0.964607041	0.964491799	0.964981374
	topNnDCG				0.943359874	0.947099009	0.942914894	0.941902138	0.941217818	0.940156558	0.940269981	0.941526973
	topNnDCG (STDEV)				0.013504527	0.010108823	0.015880611	0.014475093	0.015863779	0.021306315	0.023775329	0.025440631
	MAE				0.848598242	0.86903698	0.867403795	0.875806232	0.878836039	0.890482654	0.912006184	0.88888539
	MAE_ByUser				0.938900521	0.95954094	0.963327538	0.96271694	0.963977806	0.969411357	0.990813814	0.96880341
Item-Item	RMSE_ByRating				1.065769296	1.081055842	1.097563292	1.110347699	1.110896756	1.138039007	1.183650868	1.190007743
	RMSE_ByUser				0.992892041	1.0013589	1.029828794	1.031539683	1.036317213	1.06582795	1.106640484	1.109290064
	nDCG				0.972194205	0.973240581	0.972283881	0.971280568	0.97062942	0.969755725	0.968854709	0.968858809
	topNnDCG				0.950474529	0.952815032	0.950891239	0.948513392	0.948023639	0.945835	0.944433893	0.944870812
	topNnDCG (STDEV)				0.015779693	0.01498417	0.016188277	0.015047648	0.015677244	0.021232618	0.024955363	0.024744058
	MAE				0.825012541	0.837960494	0.855513413	0.868459749	0.874282601	0.904510999	0.946918769	0.947902017
	MAE_ByUser				0.902392148	0.910625581	0.937858814	0.937361978	0.939462441	0.965005637	1.002792233	1.000397208
	Evaluation_time (hours)				21.61850917	11.69180556	18.21101139	10.24634222	6.882604444	5.138458889	4.963068611	6.320891393
Graph Statistics	Avg Number Users/Cluster	147612	73806	36903	18451	9225	4473	2270	1180	590	295	151
	STDEV(AvgNumUsers/Cl.)		29693	19018.75802	10570.25499	5695.781356	2651.103332	1404.336482	765.1677881	388.6119607	198.3684854	97.46509463
	Avg Number Items/Cluster	48794	30931	26593	19714	13885	9820	7254	5103	3645	2635	1889
	STDEV(AvgNumItems/Cl.)		17468	13716.72177	13520.5334	11375.79436	8874.460448	6909.587132	5368.755442	4013.39388	3063.173696	2343.61968
	Avg in-Degree of Items/Cl.	167.97305	100.2549666	53.6387825	34.64267257	23.01646544	16.00947676	10.92773317	7.752322367	5.397197495	3.8039413	2.783455059
	STDEV(AvgItemsInDegree)		57.07877999	56.95780748	36.78517258	25.34400665	15.86029649	10.4869764	7.453183215	4.741587081	3.171399566	2.064762752

7.2 Παράρτημα ΙΙ (Διάγραμμα Επικοινωνίας των Components)



8 Βιβλιογραφία

- Anand, R., Ullman J.D., Melville P. and V. Sindhwani (2012). "Recommendation Systems." Mining of massive datasets. New York, N.Y.: Cambridge University Press.
- Anglade, A., Tiemann, M., & Vignoli, F. (2007, October). Complex-network theoretic clustering for identifying groups of similar listeners in p2p systems. InProceedings of the 2007 ACM conference on Recommender systems (pp. 41-48). ACM.
- Backstrom, L., Huttenlocher, D., Kleinberg, J., & Lan, X. (2006, August). Group formation in large social networks: membership, growth, and evolution. InProceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 44-54). ACM.
- Belkin, N. J., & Croft, W. B. (1992). Information filtering and information retrieval: two sides of the same coin?. Communications of the ACM, 35(12), 29-38.
- Cotter, P., & Smyth, B. (2000, July). Ptv: Intelligent personalised tv guides. InAAAI/IAAI (pp. 957-964).
- De Meo, P., Ferrara, E., Fiumara, G., & Provetti, A. (2011, November). Improving recommendation quality by merging collaborative filtering and social relationships. In Intelligent Systems Design and Applications (ISDA), 2011 11th International Conference on (pp. 587-592). IEEE.
- Deshpande, M., & Karypis, G. (2004). Item-based top-n recommendation algorithms. ACM Transactions on Information Systems (TOIS), 22(1), 143-177.
- Eirinaki, M., Louta, M. D., & Varlamis, I. A Trust-Aware System for Personalized User Recommendations in Social Networks, in IEEE Transactions on Systems, Man, and Cybernetics: Systems (Part A) (TSMCA), (Volume:PP , Issue: 99), 2013.
- Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. Communications of the ACM, 35(12), 61-70.

° Herlocker, J. L., Konstan, J. A., Borchers, A., & Riedl, J. (1999, August). An algorithmic framework for performing collaborative filtering. In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval (pp. 230-237). ACM.

° Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems (TOIS), 22(1), 5-53.

° Malone, T. W., Grant, K. R., Turbak, F. A., Brobst, S. A., & Cohen, M. D. (1987). Intelligent information-sharing systems. Communications of the ACM,30(5), 390-402.

° Maltz, D. A. (1994). Distributing information for collaborative filtering on Usenet net news.

° Melville, P., & Sindhwani, V. (2010). Recommender systems. In Encyclopedia of machine learning (pp. 829-838). Springer US.

° Melville, P., Mooney, R. J., & Nagarajan, R. (2002). Content-boosted collaborative filtering for improved recommendations. AAAI/IAAI, 187-192.

° O'Connor, M., & Herlocker, J. (1999, August). Clustering items for collaborative filtering. In Proceedings of the ACM SIGIR workshop on recommender systems (Vol. 128). UC Berkeley.

° Pazzani, M. J., & Billsus, D. (2007). Content-based recommendation systems. In The adaptive web (pp. 325-341). Springer Berlin Heidelberg.

° Pham, M. C., Cao, Y., Klamma, R., & Jarke, M. (2011). A Clustering Approach for Collaborative Filtering Recommendation Using Social Network Analysis. J. UCS, 17(4), 583-604.

° Pindyck, R. S., & Rubinfeld, D. L. Econometric models and economic forecasts, 1981.

° Principe, J. C., Euliano, N. R., & Lefebvre, W. C. (1999). Neural and adaptive systems: fundamentals through simulations with CD-ROM. John Wiley & Sons, Inc..

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

° Sarwar, B. M., Karypis, G., Konstan, J., & Riedl, J. (2002, December). Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering. In Proceedings of the fifth international conference on computer and information technology (Vol. 1).

° Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001, April). Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th international conference on World Wide Web (pp. 285-295). ACM.

° Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. Advances in artificial intelligence, 2009, 4.

° Symeonidis, P., Iakovidou, N., Mantas, N., & Manolopoulos, Y. (2013). From biological to social networks: Link prediction based on multi-way spectral clustering. Data & Knowledge Engineering, 87, 226-242.