Αθήνα, 2018

Professor: Y. Kotidis

Assistant: I. Filippidou

*Authors:*

Polyzogopoulos Pavlos

Katsaris Christos

**Mining Big Datasets**

**Assignment 1**

Εισαγωγή

Κατά τα τελευταία έτη και λόγω της ταχείας ανάπτυξης του Διαδικτύου σε συνδυασμό με το πρόβλημα της συσσώρευσης πληροφοριών, η χρήση των συστημάτων συστάσεων -ΣΣ (recommender systems) έχει αρχίσει να γίνεται απαραίτητη τόσο για τις επιχειρήσεις όσο και για τους τους πελάτες. Τα συστήματα συστάσεων και εξατομίκευσης είναι αλγόριθμοι που χρησιμοποιούνται ευρέως στο ηλεκτρονικό εμπόριο για να προτείνονται προϊόντα ή υπηρεσίες σε χρήστες. Σύσταση θεωρείται μία λίστα κορυφαίων αντικειμένων που εκτιμάται ότι ενδιαφέρει περισσότερο ένα μέλος του δικτύου. Ένας αξιόλογος αριθμός αλγορίθμων έχει αναπτυχθεί τα τελευταία χρόνια, με τους περισσότερους να χρησιμοποιούνται σε εμπορικά περιβάλλοντα. Στόχος των ΣΣ μπορεί να είναι να βοηθήσουν τον χρήστη να αποφασίσει τι να αγοράσει, ποιον να κάνει φίλο σε ένα κοινωνικό δίκτυο ή το τι να διαβάσει (Konstan & Riedl, 2012; Polatidis & Georgiadis, 2013; Prasad & Kumari, 2012).

Ανάμεσα στις πιο δημοφιλείς ιστοσελίδες που χρησιμοποιούν συστήματα συστάσεων είναι η Amazon.com, η οποία παρέχει μια εξατομικευμένη ιστοσελίδα για κάθε μεμονωμένο χρήστη. Το Netflix χρησιμοποιεί εξατομικευμένα συστήματα για να προταθούν ταινίες και τηλεοπτικές εκπομπές. Επίσης το Facebook είναι ένα άλλο παράδειγμα ιστοσελίδας που χρησιμοποιεί εξατομικευμένα συστήματα συστάσεων. Το Facebook είναι μια διαδικτυακή (online) υπηρεσία κοινωνικής δικτύωσης (SNS) που ιδρύθηκε το Φεβρουάριο του 2004. Αν και αρχικά η ιστοσελίδα αποτελούσε ένα δίκτυο για φοιτητές κολλεγίων που ήθελαν να μένουν διασυνδεμένοι ανταλλάζοντας πληροφορίες και βρίσκοντας νέους φίλους, στη συνέχεια επεκτάθηκε σε ένα παγκόσμιο δίκτυο. Από το Σεπτέμβριο του 2012, το Facebook έχει πάνω από ένα δισεκατομμύριο ενεργούς χρήστες. Οι χρήστες μετά την εγγραφή τους στο δικτυακό τόπο δημιουργούν ένα προσωπικό προφίλ με φωτογραφίες, λίστες προσωπικών ενδιαφερόντων, στοιχεία επικοινωνίας, και άλλες προσωπικές πληροφορίες. Μπορούν επίσης να προσθέσουν άλλους χρήστες ως φίλους (friends) στις κοινωνικές επαφές τους. Το Facebook χρησιμοποιεί εξατομικευμένα συστήματα συστάσεων για να προτείνει πιθανές φιλίες μεταξύ των χρηστών του.

Δεδομένου ενός στιγμιότυπου του γράφου θα θέλαμε να συμπεράνουμε τις αλληλεπιδράσεις (φιλίες) που θα προκύψουν στο μέλλον μεταξύ των μελών του. Το πρόβλημα αυτό είναι γνωστό ως link prediction problem και σκοπός αυτής της εργασίας είναι εξετάσει δεδομένα από το Facebook και να φτιάξει ένα ΣΣ το οποίο θα προτείνει πιθανές φιλίες μεταξύ των χρηστών. Τα δεδομένα που θα εξεταστούν αποτελούνται από 4.039 διαφορετικούς χρήστες (nodes) ενώ υπάρχουν συνολικά 88.234 φιλίες (edges) μεταξύ αυτών. Το μόνο γνώρισμα για αυτούς είναι οι φιλίες μεταξύ των χρηστών. Άλλα χαρακτηριστικά, όπως δημογραφικά στοιχεία ή προτιμήσεις των χρηστών δεν είναι γνωστά οπότε το ΣΣ θα εξετάσει μόνο αυτό το γνώρισμα.

Μεθοδολογία

Τα τελευταία χρόνια, ένας αξιόλογος αριθμός αλγορίθμων-μεθόδων έχει αναπτυχθεί για την υλοποίηση ενός ΣΣ με τους περισσότερους να χρησιμοποιούνται σε εμπορικά περιβάλλοντα. Στα πλαίσια αυτής της εργασίας θα εξεταστούν 3 διαφορετικοί αλγόριθμοι-μέθοδοι των συστημάτων συστάσεων, οι οποίοι είναι και οι πιο διαδεδομένοι. Πιο αναλυτικά θα εξεταστούν οι **Common neighbors (friend-of-friend (FoF) method), Jaccard coefficient,** και **Adamic and Adar function**. Όσο αφορά την αρχιτεκτονική με την οποία θα υπολογιστούν οι παραπάνω αλγόριθμοι, θα ακολουθήσουμε 2 διαφορετικούς τρόπους. Αρχικά (για τα ερωτήματα 2 3 και 4) επιλέξαμε να απεικονίσουμε το γράφημα με την μορφή πινάκων γειτνίασης, ενώ στην συνέχεια (ερωτήματα 5 και 6) με λίστες γειτνίασης. Γενικότερα, οι πίνακες γειτνίασης έχουν ως πλεονέκτημα ότι κάθε κόμβος χαρακτηρίζεται από έναν ακέραιο αριθμό και οι ακέραιοι αυτοί χρησιμοποιούνται για διευθυνσιοδότηση του πίνακα γειτνίασης. Στην περίπτωση που δεν χρειάζεται να αποθηκευτούν επιπρόσθετες πληροφορίες για κάθε κόμβο του γράφου(όπως συμβαίνει σε αυτή την περίπτωση), η μέθοδος είναι πολύ ελκυστική και πολλές λειτουργίες υλοποιούνται απλά και αποτελεσματικά. Από την άλλη, συνήθως οι πίνακες γειτνίασης είναι αραιοί ( η τιμή “0” εμφανίζεται πολύ συχνά) πράγμα που τους καθιστά πολλές φορές μη αποτελεσματικούς για την αποθήκευση και επεξεργασία των δεδομένων. Επίσης σε περίπτωση που ο γράφος είναι πολύ μεγάλος (εκατομμύρια χρήστες) υπάρχει ο κίνδυνος να μην χωράει στην κύρια μνήμη του υπολογιστή. Στα πλαίσια αυτής της εργασίας το τελευταίο δεν θα μας απασχολήσει καθώς τα δεδομένα περιέχουν μόνο 4.039 χρήστες. Αντιθέτως η απεικόνιση ενός γραφήματος με την μορφή λίστας γειτνίασης οδηγεί σε καλή χωρική πολυπλοκότητα ιδιαίτερα αν το αυτό αραιό, όμως η χρονική πολυπλοκότητα μίας λειτουργίας όπως «Εύρεση αν δύο κόμβοι είναι γειτονικοί» είναι μεγαλύτερη από όταν χρησιμοποιείται πίνακας γειτνίασης ενώ θεωρείται ακριβή μέθοδος ως προς τη χωρική πολυπλοκότητα για πυκνούς γράφους. Στο τέλος θα σχολιάσουμε και την απόδοση της κάθε προσέγγισης τόσο από άποψη χρόνου όσο και πολυπλοκότητας. Η επεξεργασία των δεδομένων θα γίνει στο περιβάλλον της στατιστικής γλώσσας R, ενώ ο κώδικας για κάθε ένα από τα ερωτήματα βρίσκεται στο αρχείο **Data Mining Techniques - Assignment 1\_R\_Code**

Ανάλυση και επεξεργασία δεδομένων

Αρχικά, τα δεδομένα είναι αποθηκευμένα σε συμπιεσμένο txt αρχείο. Εισάγοντας το αρχείο στην R τα δεδομένα αποτυπώνονται σε έναν πίνακα (dataframe) με δύο στήλες. Κάθε μία από τις στήλες περιέχει τα ids των 4.039 χρηστών (nodes), ενώ κάθε γραμμή, (88.234 συνολικά) υποδηλώνει και την ύπαρξη μίας φιλίας μεταξύ αυτών (edges), όπως φαίνεται και στον παρακάτω πίνακα (Πίνακας 1).

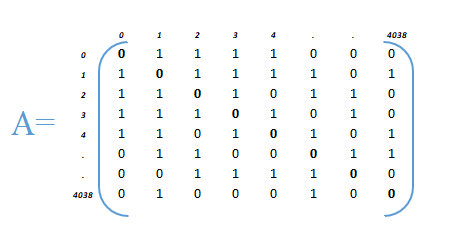
|  |  |  |
| --- | --- | --- |
|  | C1 | C2 |
| R1 | 0 | 1 |
| R2 | 0 | 3 |
| R3 | 1 | 73 |
| R4 | 5 | 213 |
| R5 | 24 | 32 |
| R6 | 755 | 5 |
| R7 | 1002 | 56 |
| . |  |  |
| . |  |  |
| R88.234 | 4039 | 402 |

Πίνακας 1:Το γράφημα αποτελείται από 4039 χρήστες του Facebook ενώ υπάρχουν συνολικά 88.234 φιλίες μεταξύ αυτών. Σύμφωνα με τον πίνακα ο χρήστης με id "0" είναι φίλος με τον χρήστη "1" αλλά και με τον χρήστη "3". Αντίστοιχα ο χρήστης "1" είναι φίλος με τον χρήστη “73” κ.ο.κ.

Το γράφημα είναι κατευθυνόμενο, δηλαδή είναι της μορφής D = (V, A) όπου V, είναι ένα σύνολο από κορυφές ή κόμβοι και Α, είναι μια σειρά από διατεταγμένα ζεύγη κορυφών, τα οποία ονομάζονται τόξα. Ένα τόξο a = (x, y) θεωρείται ότι κατευθύνεται από το x στο y. Το πρώτο βήμα προκειμένου να μπορέσουμε να επεξεργαστούμε τα δεδομένα είναι η μετατροπή του σε μη κατευθυνόμενο. Ένα μη-κατευθυνόμενο γράφημα είναι εκείνο στο οποίο οι ακμές δεν έχουν προσανατολισμό. Η ακμή (α, β) είναι ταυτόσημη με την άκρη (β, α), δηλαδή, δεν υπάρχουν διατεταγμένα ζεύγη, αλλά σύνολα {u, v} των κορυφών.

Η διαδικασία αυτή έγινε με την χρήση της R. Αρχικά δημιουργήθηκε ένας δεύτερος πίνακας ίδιων διαστάσεων, ο οποίος όμως ήταν της μορφής (C2,C1) είχε δηλαδή αντεστραμμένες στήλες σε σχέση με τον αρχικό. Το αποτέλεσμα της συγχώνευσης αυτών των δύο πινάκων, κρατώντας όμως μόνο τα μοναδικά ζεύγη προκειμένου κάθε φιλία να εμφανίζεται μόνο μία φορά, είναι ένα μη κατευθυνόμενο γράφημα με 176.468 φιλίες μεταξύ των χρηστών (edges).

Εν συνεχεία, προκειμένου να γίνει εφικτή η προσέγγιση του προβλήματος μέσω πινάκων γειτνίασης , θα πρέπει και να δημιουργηθεί και η αντίστοιχη δομή. Κάνοντας χρήση του πακέτου “reshape2” της R, το γράφημα θα αναπαρασταθεί μέσα από έναν πίνακα A = [αi,j ] με διαστάσεις n x n. Η θέση του πίνακα i, j υποδηλώνει την ύπαρξη ή όχι της κατευθυνόμενης ακμής (i, j) και περιέχει τις τιμές “0” και “1”, όπου εάν e = (i, j) ∈ E τότε A[i][j] = 1 αλλιώς 0. Ο πίνακας έχει συνολικά n² στοιχεία (όπου n ο αριθμός των χρηστών) και είναι συμμετρικός μιας και το γράφημα είναι πια μη-κατευθυνόμενο, ενώ οι διαγώνιες τιμές του είναι μηδενικές (Πίνακας 2).



Πίνακας 2: Αναπαράσταση του γράφου με την μορφή ενός πίνακα γειτνίασης. Οι διαστάσεις του πίνακα είναι 4039x4039 ενώ οι διαγώνιες τιμές του είναι μηδενικές. Η τιμή “1” δηλώνει ποιοι χρήστες είναι φίλοι μεταξύ τους ενώ η τιμή “0” το αντίθετο. Από τον πίνακα προκύπτει ότι ο χρήστης 0 είναι φίλος με τους 1 2 3 και 4 κ.ο.κ.

Προσεγγίσεις σύστασης

Σε αυτό το κεφάλαιο θα εξεταστούν κάποιες από τις μεθόδους που είναι διαθέσιμες για το link prediction problem. Όλοι οι αλγόριθμοι υπολογίζουν ένα score(X,Y) για τα ζεύγη των μελών (X,Y) του γραφήματος και παράγουν μία ταξινομημένη λίστα συστάσεων (σε φθίνουσα μορφή). Μπορεί να θεωρηθεί ότι υπολογίζουν το βαθμό ομοιότητας μεταξύ των μελών Χ και Υ του γραφήματος λαμβάνοντας υπόψιν την τοπολογία του δικτύου. Οι αλγόριθμοι που θα εξεταστούν παρακάτω υπολογίζονται βάση της γειτονιάς του κάθε χρήστη και βασίζονται στην ιδέα ότι εάν οι γειτονιές δύο χρηστών X και Y επικαλύπτονται τότε υπάρχει μεγάλη πιθανότητα αυτοί οι δύο χρήστες να αναπτύξουν μία φιλία στο μέλλον.

***Common Neighbors (friend-of-friend (FoF) method)*** Αυτός ο αλγόριθμος απορρέει από την ιδέα ότι εάν δύο μη φίλοι στο γράφημα έχουν πολλούς κοινούς φίλους μεταξύ τους, τότε υπάρχει μεγάλη πιθανότητα να γίνουν και αυτοί φίλοι στο μέλλον. Με αυτή την λογική λοιπόν εάν ο χρήστης Χ, ο οποίος δεν είναι φίλος με τον χρήστη Υ, μοιράζονται έναν κοινό φίλο τότε ίσως αυτοί θα έπρεπε να γίνουν φίλοι μεταξύ τους. Καλύτερη πρόταση θεωρείται ο χρήστης με τον οποίο ο Χ έχει τους περισσότερους κοινούς φίλους.

Η συνάρτηση με την οποία υπολογίζεται η κάθε πρόταση φίλιας μεταξύ όλων των μη-φίλων είναι η παρακάτω:

𝒔𝒄𝒐𝒓𝒆 (𝑨, 𝑩) = |𝑵𝑨 ∩ 𝑵𝑩| (1)

Όπου ΝΑ είναι η γειτονιά του χρήστη Α.

Το αποτέλεσμα του πολλαπλασιασμού μεταξύ του αρχικού πίνακα Α με κάθε έναν από τους χρήστες για τους οποίους ενδιαφερόμαστε (στα πλαίσια αυτής της εργασίας είναι οι χρήστες 107 1126 14 και 35), θα μας δώσει τον συνολικό αριθμό των κοινών φίλων μεταξύ κάθε έναν από αυτούς με τους υπόλοιπους χρήστες. Ο κάθε χρήστης αναπαρίσταται και αυτός ως ένας πίνακας ίδιας μορφής με αυτής του Α με διαστάσεις όμως 4039x1. Ουσιαστικά αυτό που θα υπολογίσουμε είναι το πόσοι είναι οι κοινοί φίλοι των παραπάνω χρηστών με όλους τους υπόλοιπους χρήστες. Στο τέλος θα κρατήσουμε τις 10 καλύτερες προτάσεις, δηλαδή τους 10 χρήστες με το μεγαλύτερο σκορ (περισσότεροι κοινοί φίλοι), έχοντας εξαιρέσει προφανώς όλους τους αρχικούς φίλους του κάθε χρήστη.

Σύμφωνα με τα παραπάνω, ο τελικές προτάσεις για κάθε έναν από τους 4 χρήστες είναι οι εξής:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***id*** | 513 | 400 | 559 | 373 | 492 | 500 | 378 | 436 | 431 | 514 |
| ***score*** | 19 | 18 | 18 | 17 | 17 | 17 | 16 | 16 | 15 | 15 |

Πίνακας 3: Final Friends Recommendation list using CN algorithm for **Node107**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***id*** | 916 | 1238 | 1750 | 1230 | 1004 | 1791 | 1530 | 1172 | 1570 | 1597 |
| ***score*** | 113 | 103 | 99 | 98 | 94 | 94 | 90 | 89 | 85 | 84 |

Πίνακας 3: Final Friends Recommendation list using CN algorithm for **Node1126**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***id*** | 2 | 17 | 140 | 111 | 137 | 162 | 19 | 333 | 44 | 243 |
| ***score*** | 9 | 9 | 9 | 8 | 7 | 7 | 6 | 6 | 5 | 4 |

Πίνακας 4: Final Friends Recommendation list using CN algorithm for **Node14**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***id*** | 46 | 68 | 99 | 131 | 175 | 177 | 225 | 227 | 278 | 321 |
| ***score*** | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |

Πίνακας 5: Final Friends Recommendation list using CN algorithm for **Node35**

Σύμφωνα με τα αποτελέσματα, ο χρήστης 107 θα πρέπει να γίνει φίλος με τους χρήστες 513 400 559 373 492 500 378 436 431 και 514 καθώς με αυτούς έχει τους περισσότερους κοινούς φίλους. Αντίστοιχα ερμηνεύονται και τα αποτελέσματα για τους χρήστες 1126 14 και 35.

***Jaccard’s Coefficient*** Ο συντελεστής Jaccard υπολογίζει την ομοιότητα μεταξύ δύο χρηστών ανάλογα με ένα χαρακτηριστικό τους γνώρισμα. Στην συγκεκριμένη περίπτωση, εάν θεωρήσουμε ως χαρακτηριστικό την γειτονιά του κάθε χρήστη, τότε ο συντελεστής Jaccard θα υπολογίσει το ποσοστό των κοινών φίλων μεταξύ δύο χρηστών (ένωση) προς το συνολικό αριθμό των φίλων τους (τομή). Ο συντελεστής Jaccard παίρνει τιμές από 0 έως 1 (ή 0% έως 100%) και για δύο χρήστες Α και Β υπολογίζεται από την παρακάτω εξίσωση:

𝒔𝒄𝒐𝒓𝒆 (𝑨, 𝑩) = |𝑵𝑨 ∩ 𝑵𝑩| / |𝑵𝑨 ∪ 𝑵𝑩| (2)

Η τομή (κοινοί φίλοι) μεταξύ των χρηστών έχει ήδη υπολογιστεί στο παραπάνω ερώτημα, ενώ η ένωση (συνολικοί φίλοι) μπορεί να υπολογιστεί πολύ εύκολα καθώς γνωρίζουμε ήδη τους αρχικούς φίλους του κάθε χρήστη. Η ένωση δύο χρηστών μπορεί να υπολογιστεί και ως το άθροισμα των φίλων κάθε χρήστη μείον την τομή τους. Οπότε η εξίσωση (2) μπορεί να γραφτεί και ως εξής:

𝒔𝒄𝒐𝒓𝒆 (𝑨, 𝑩) = |𝑵𝑨 ∩ 𝑵𝑩| /|𝑵𝑨| + |𝑵𝑩| - |𝑵𝑨 ∩ 𝑵𝑩| (3)

Το μόνο που μένει να υπολογιστεί είναι οι αρχικοί φίλοι του κάθε χρήστη που δεν είναι άλλο από το άθροισμα κάθε στήλης του αρχικού πίνακα γειτνίασης Α. Έχοντας λοιπόν όλα στοιχεία που απαιτούνται και τοποθετώντας τα στην εξίσωση παίρνουμε τις παρακάτω προτάσεις φιλίας για τους 4 χρήστες:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***id*** | 513 | 400 | 559 | 492 | 500 | 373 | 436 | 378 | 515 | 514 |
| ***score*** | 1,7% | 1,6% | 1,6% | 1,6% | 1,5% | 1,5% | 1,5% | 1,5% | 1,4% | 1,4% |

Πίνακας 6: Final Friends Recommendation list using Jaccard Coefficient (%) for Node107

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***id*** | 916 | 1750 | 1230 | 1530 | 1004 | 1238 | 1172 | 1791 | 1789 | 1597 |
| ***score*** | 48,7% | 44,0% | 43,6% | 42,3% | 42,0% | 41,7% | 40,1% | 40,0% | 37,4% | 37,2% |

Πίνακας 7 Final Friends Recommendation list using Jaccard Coefficient (%) for Node1126

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***id*** | 2 | 140 | 17 | 162 | 111 | 333 | 44 | 137 | 19 | 243 |
| ***score*** | 56,2% | 52,9% | 47,4% | 43,8% | 38,1% | 35,3% | 31,2% | 29,2% | 24,0% | 21,1% |

Πίνακας 8 Final Friends Recommendation list using Jaccard Coefficient (%) for Node14

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***id*** | 321 | 11 | 12 | 15 | 18 | 37 | 43 | 74 | 114 | 209 |
| ***score*** | 66,7% | 50,0% | 50,0% | 50,0% | 50,0% | 50,0% | 50,0% | 50,0% | 50,0% | 50,0% |

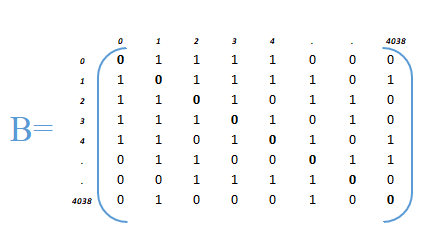
Πίνακας 9 Final Friends Recommendation list using Jaccard Coefficient (%) for Node35

Σύμφωνα με τα αποτελέσματα ο χρήστης 107 θα πρέπει να γίνει φίλος με τους χρήστες 513 400 559 492 500 373 436 378 515 και 514 καθώς με αυτούς έχει τις περισσότερες ομοιότητες. Αντίστοιχα ερμηνεύονται και τα αποτελέσματα για τους υπόλοιπους χρήστες. Με μία πρώτη ματιά φαίνεται ότι ο αλγόριθμος Common Neighbors και ο συντελεστής Jaccard έχουν αρκετές ομοιότητες όσο αφορά τους χρήστης 107 1126 και 14 γεγονός που δεν ισχύει και για τον χρήστη 35.

***Adamic and Adar function*** Είναι και αυτή μία συνάρτηση που υπολογίζει την πιθανότητα δύο χρηστών Α και Β να γίνουν φίλοι στο μέλλον και βασίζεται στην γειτονιά των κοινών τους φίλων με την διαφορά όμως ότι δίνει μεγαλύτερη βαρύτητα στους κοινούς τους φίλους οι οποίοι εμφανίζονται πιο σπάνιοι μέσα στο δίκτυο. Δηλαδή, εάν δύο χρήστες του δικτύου, οι οποίοι δεν είναι φίλοι μεταξύ τους έχουν έναν κοινό φίλο ο οποίος είναι “διάσημος” (έχει δηλαδή πολλούς φίλους), τότε αυτοί οι χρήστες είναι λιγότερο όμοιοι μεταξύ τους σε σχέση με δύο άλλους των οποίων ο κοινός φίλος έχει σχετικά λίγους φίλους. Η συνάρτηση Adamic and Adar υπολογίζεται με την παρακάτω συνάρτηση:

𝒔𝒄𝒐𝒓𝒆 (𝑨, 𝑩) **= ∑w∈Γ(u)∩Γ(v)=1/log| 𝑵𝑪 |** (4)

Αρχικά θα πρέπει από τον αρχικό πίνακα γειτνίασης Α να υπολογιστούν οι φίλοι που έχουν όλοι οι χρήστες του δικτύου (| 𝑵𝑪 |) και στην συνέχεια ο παρονομαστής της παραπάνω συνάρτησης (1/log| 𝑵𝑪 |). Το αποτέλεσμα θα είναι ένα διάνυσμα με 4039 στοιχεία. Εν συνεχεία, θα πρέπει να βρούμε ποιοι είναι οι κοινοί φίλοι του κάθε χρήστη για τον οποίον ενδιαφερόμαστε με όλους τους υπόλοιπους χρήστες. Πολλαπλασιάζοντας τον πίνακα Α με τον εν λόγω χρήστη θα πάρουμε έναν νέο πίνακα Β (έναν για κάθε χρήστη) με διαστάσεις 4039x4039. Η τιμή “1” δηλώνει αυτή την φορά το που υπάρχει κοινή φιλία (βλέπε πίνακα 10). Τέλος το αποτέλεσμα του γινομένου του πίνακα Β με το διάνυσμα θα μας δώσει το αποτέλεσμα της συνάρτησης (4) με όλους τους υπόλοιπους χρήστες.



Πίνακας 10: Ο πίνακας Β είναι το αποτέλεσμα του πολλαπλασιασμού του πίνακα Α με το διάνυσμα του χρήστη 107. Ο χρήστης 0 είναι και αυτός φίλος, όπως και ο 107, με τους χρήστες 1 2 3 4. Αντίστοιχα οι χρήστες 0 2 3 4 και 4038 είναι φίλοι τόσο με τον 1 όσο και με τον 107.

Να σημειωθεί σε αυτό το σημείο ότι καθώς log1=0, η διαίρεση 1/ log1 δεν ορίζεται. Επομένως ο πίνακας Β θα έχει σε μερικές περιπτώσεις τις τιμές NaN και Inf. Προκειμένου να μην αλλοιωθούν τα τελικά αποτελέσματα θα πρέπει αυτές οι τιμές να μην ληφθούν υπόψιν.

Σύμφωνα με τα παραπάνω, ο τελικές προτάσεις για κάθε έναν από τους 4 χρήστες είναι οι εξής:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***id*** | 513 | 400 | 559 | 500 | 492 | 373 | 378 | 436 | 524 | 514 |
| ***score*** | 4,22 | 4,04 | 3,83 | 3,70 | 3,64 | 3,63 | 3,41 | 3,39 | 3,23 | 3,23 |

Πίνακας: Final Friends Recommendation list using Adamic and Adar function for Node1126

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***id*** | 916 | 1238 | 1750 | 1230 | 1004 | 1791 | 1530 | 1172 | 1570 | 1597 |
| ***score*** | 23,26 | 21,17 | 20,19 | 19,90 | 19,13 | 19,09 | 18,16 | 18,06 | 17,37 | 17,00 |

Πίνακας ς 11Final Friends Recommendation list using Adamic and Adar function for Node1126

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***id*** | 2 | 17 | 140 | 111 | 162 | 137 | 333 | 19 | 44 | 243 |
| ***score*** | 2,97 | 2,93 | 2,93 | 2,59 | 2,26 | 2,20 | 2,00 | 1,82 | 1,51 | 1,18 |

Πίνακας ς 12Final Friends Recommendation list using Adamic and Adar function for Node14

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***id*** | 46 | 68 | 99 | 131 | 175 | 177 | 225 | 227 | 278 | 321 |
| ***score*** | 0,57 | 0,57 | 0,57 | 0,57 | 0,57 | 0,57 | 0,57 | 0,57 | 0,57 | 0,57 |

Πίνακας 13Final Friends Recommendation list using Adamic and Adar function for Node35

Σύμφωνα με τα αποτελέσματα ο χρήστης 107 θα πρέπει να γίνει φίλος με τους χρήστες 513 400 559 500 492 373 378 436 524 και 514 καθώς με αυτούς έχει τις περισσότερες ομοιότητες. Αντίστοιχα ερμηνεύονται και τα αποτελέσματα για τους υπόλοιπους χρήστες. Με μία πρώτη ματιά φαίνεται ότι η συνάρτηση των Adamic και Adar πρότεινε σχεδόν τους ίδιους χρήστες με τον αλγόριθμο Common Neighbors και για τους 4 χρήστες.

Evaluation of the recommendation system

Στην συνέχεια θα εξεταστεί ο βαθμός συμφωνίας των παραπάνω μεθόδων. Για κάθε ζεύγος από αυτές (FoF-jaccard, Jaccard-Adamic, FoF-Adamic), θα υπολογιστεί το ποσοστό που συμφωνούν στις τελικές τους προτάσεις (αριθμός κοινών προτάσεων φιλίας στην τελική λίστα 10 προτάσεων), ενώ στο τέλος θα υπολογιστεί και ο μέσος βαθμός συμφωνίας των 3 μεθόδων. Αυτή την φορά θα εξεταστούν όσοι χρήστες έχουν id πολλαπλάσιο του 100. Θα εξεταστούν δηλαδή 40 χρήστες του δικτύου οι οποίοι είναι οι χρήστες με id : 100 200 300 400 500 600 700 800 900 1000 1100 1200 1300 1400 1500 1600 1700 1800 190 2000 2100 2200 2300 2400 2500 2600 2700 2800 2900 3000 3100 3200 3300 3400 3500 3600 3700 3800 3900 και 4000.

Όπως έχει αναφερθεί παραπάνω, αυτό το πρόβλημα θα το προσεγγίσουμε απεικονίζοντας το γράφημα με λίστες γειτνίασης. Θα χρησιμοποιήσουμε δηλαδή έναν Νx1 πίνακα Α, όπου Α[i] είναι αναφορά σε λίστα των κόμβων που γειτονεύουν με την κορυφή i. Αυτό θα γίνει κάνοντας χρήση του πακέτου “dplyr” στην R. Οι συναρτήσεις που υπάρχουν στο περιβάλλον της R όπως είναι οι union και intersect φάνηκαν πολύ χρήσιμες προκειμένου να αντλήσουμε τις πληροφορίες που θέλουμε και να υπολογίσουμε τους 3 αλγόριθμους. Επίσης, προκειμένου η λύση του προβλήματος να είναι πιο αποτελεσματική, φτιάξαμε ένα πρόγραμμα (function) το οποίο δέχεται σαν παράμετρο μία λίστα από χρήστες (ids) και εκτυπώνει τις προτάσεις φιλίας για κάθε έναν από αυτούς (τις καλύτερες 10 προτάσεις). Κάθε φορά περιορίζουμε την αναζήτηση στους χρήστες που απαρτίζουν την γειτονιά του κάθε χρήστη (input). Με τον τρόπο αυτό αποφεύγονται περιττές αναζητήσεις και επομένως μειώνεται ο χρόνος που απαιτείται για τον αλγόριθμο.

Πιο αναλυτικά το κάθε πρόγραμμα λειτουργεί ως εξής:

***Common Neighbors (friend-of-friend (FoF) method)***

**Input:** Έναν χρήστη (ή μία λίστα με χρήστες) X του αρχικού γραφήματος G= {V, E}

**Output**: Μία ταξινομημένη λίστα (σε φθίνουσα σειρά) που περιέχει τις 10 κορυφαίες προτάσεις (μεγαλύτερο βαθμό ομοιότητας) μεταξύ του κάθε χρήστη (input) και όλων των μη φίλων του.

Βήματα:

1. Βρες ποιοι είναι οι φίλοι του χρήστη X
2. Βρες τους φίλους των φίλων του χρήστη X
3. Υπολόγισε τους κοινούς φίλους που έχει ο αρχικός χρήστης με κάθε έναν από αυτούς του βήματος 2 (τομή)
4. Ταξινόμησε το αποτέλεσμα της λίστας του βήματος 3 σε φθίνουσα σειρά εξαιρώντας τους αρχικούς φίλους του χρήστη καθώς και το ίδιο από αυτήν.
5. Εμφάνισε την τελικά λίστα

***Jaccard’s Coefficient***

**Input:** Έναν χρήστη (ή μία λίστα με χρήστες) X του αρχικού γραφήματος G= {V, E}

**Output**: Μία ταξινομημένη λίστα (σε φθίνουσα σειρά) που περιέχει τις 10 κορυφαίες προτάσεις (μεγαλύτερο βαθμό ομοιότητας) μεταξύ του κάθε χρήστη (input) και όλων των μη φίλων του.

Βήματα:

1. Βρες ποιοι είναι οι φίλοι του χρήστη X
2. Βρες τους φίλους των φίλων του χρήστη X
3. Υπολόγισε τους κοινούς φίλους που έχει ο αρχικός χρήστης με κάθε έναν από αυτούς του βήματος 2 (τομή)
4. Υπολόγισε τους συνολικούς φίλους που έχει ο αρχικός χρήστης με κάθε έναν από αυτούς του βήματος 2 (ένωση)
5. Υπολόγισε τον συντελεστή Jaccard (τομή/ένωση)
6. Ταξινόμησε το αποτέλεσμα της λίστας του βήματος 5 σε φθίνουσα σειρά εξαιρώντας τους αρχικούς φίλους του χρήστη καθώς και το ίδιο από αυτήν.
7. Εμφάνισε την τελικά λίστα

***Adamic and Adar function***

**Input:** Έναν χρήστη (ή μία λίστα με χρήστες) X του αρχικού γραφήματος G= {V, E}

**Output**: Μία ταξινομημένη λίστα (σε φθίνουσα σειρά) που περιέχει τις 10 κορυφαίες προτάσεις (μεγαλύτερο βαθμό ομοιότητας) για τον χρήστη X και όλων των μη φίλων του.

Βήματα:

1. Βρες ποιοι είναι οι φίλοι του χρήστη X
2. Βρες τους φίλους των φίλων του χρήστη X
3. Υπολόγισε για κάθε έναν από αυτούς τον αντίστροφο λογάριθμο της γειτονιά του (παρονομαστής της συνάρτηση Adamic and Adar)
4. Υπολόγισε την συνάρτηση Adamic and Adar του χρήστη X με κάθε ένα μη φίλο του αθροίζοντας τα αποτελέσματα του βήματος 3
5. Ταξινόμησε το αποτέλεσμα της λίστας του βήματος 4 σε φθίνουσα σειρά εξαιρώντας τους αρχικούς φίλους του χρήστη καθώς και το ίδιο από αυτήν.
6. Εμφάνισε την τελικά λίστα

Όσο αφορά την απόδοση των παραπάνω μεθόδων, δεν απαιτείται σημαντικά περισσότερος χρόνος σε σχέση με την πρώτη προσέγγιση του προβλήματος. Επιπλέον όσο αφορά καθαρά το προγραμματιστικό κομμάτι μας δίνει την δυνατότητα να εισάγουμε μία λίστα με χρήστες και να πάρουμε συγκεντρωτικά τα αποτελέσματα.

Εισάγοντας λοιπόν τους εν λόγω χρήστες σε ένα πρόγραμμα που υπολογίζει και τις 3 μεθόδους ταυτόχρονα και συγκρίνοντας τις κοινές προτάσεις των αλγόριθμων μεταξύ τους παίρνουμε τα εξής αποτελέσματα:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| ***id*** | ***FoF-jaccard*** | ***Jaccard-Adamic*** | ***FoF-Adamic*** |
| 100 | 20% | 50% | 30% |
| 200 | 90% | 90% | 80% |
| 300 | 60% | 100% | 60% |
| 400 | 80% | 100% | 80% |
| 500 | 90% | 90% | 80% |
| 600 | 10% | 100% | 10% |
| 700 | 30% | 90% | 30% |
| 800 | 90% | 100% | 90% |
| 900 | 70% | 90% | 70% |
| 1000 | 60% | 80% | 70% |
| 1100 | 80% | 100% | 80% |
| 1200 | 0% | 30% | 30% |
| 1300 | 50% | 90% | 40% |
| 1400 | 20% | 100% | 20% |
| 1500 | 70% | 0% | 70% |
| 1600 | 50% | 100% | 50% |
| 1700 | 10% | 90% | 10% |
| 1800 | 100% | 100% | 100% |
| 1900 | 70% | 100% | 70% |
| 2000 | 70% | 80% | 60% |
| 2100 | 90% | 100% | 90% |
| 2200 | 40% | 100% | 40% |
| 2300 | 10% | 60% | 20% |
| 2400 | 90% | 100% | 90% |
| 2500 | 100% | 100% | 100% |
| 2600 | 30% | 80% | 30% |
| 2700 | 90% | 100% | 90% |
| 2800 | 20% | 90% | 30% |
| 2900 | 70% | 90% | 70% |
| 3000 | 60% | 80% | 60% |
| 3100 | 50% | 100% | 50% |
| 3200 | 70% | 90% | 80% |
| 3300 | 60% | 90% | 70% |
| 3400 | 0% | 90% | 0% |
| 3500 | 20% | 100% | 20% |
| 3600 | 50% | 100% | 50% |
| 3700 | 90% | 100% | 90% |
| 3800 | 90% | 100% | 90% |
| 3900 | 10% | 80% | 20% |
| 4000 | 60% | 100% | 60% |

Πίνακας 14: Τα ποσοστά συμφωνίας σε % ανά ζεύγος αλγορίθμων για τους παραπάνω χρήστες

Με μία πρώτη ματιά οι αλγόριθμοι Jaccard Coefficient και Adamic and Adam φαίνεται να έχουν το μεγαλύτερο ποσοστό συμφωνίας για τους συγκεκριμένους χρήστες. Υπολογίζοντας το μέσο ποσοστό συμφωνίας μεταξύ των αλγόριθμων επαληθεύεται ακριβώς αυτό. Ο μέσο όρος συμφωνίας αυτού του ζεύγους είναι 90,7%. Αντίστοιχα το ζεύγος FoF-jaccard συμφωνεί στις προτάσεις του κατά 55,5%, ενώ οι FoF και Adamic κατά 57%.

Ως δεύτερη αξιολόγηση των μεθόδων προκειμένου να εξεταστεί εάν η εκάστοτε προτεινόμενη φιλία θα γίνει αποδεκτή από τον κάθε χρήστη θα υπολογίσουμε τον εξής αλγόριθμο:

1. Διάλεξε τυχαία ένα ζεύγος φίλων μέσα στο δίκτυο (π.χ. F1 και F2)
2. Αφαίρεσε το ζεύγος από τα δεδομένα
3. Υπολόγισε για κάθε έναν από τους F1 και F2 τις λίστες με τους προτεινόμενους φίλους
4. Προσδιόρισε την θέση του F1 μέσα στην λίστα του F2 (εάν εμφανίζεται)
5. Προσδιόρισε την θέση του F2 μέσα στην λίστα του F1 (εάν εμφανίζεται)
6. Υπολόγισε τη μέση θέση αυτών

Ο παραπάνω αλγόριθμος θα τρέξει για 100 διαφορετικά πραγματικά ζεύγη φίλων του δικτύου και καλύτερη μέθοδος θεωρείται αυτή με το μικρότερο αποτέλεσμα. Με άλλα λόγια αυτό που θα θέλουμε να εξετάσουμε είναι εάν τα αποτελέσματα των παραπάνω μεθόδων ανταποκρίνονται στην πραγματικότητα και αν ναι ποια είναι η μέθοδος που το κάνει καλύτερα. Ένας άλλος τρόπος για να εξετάσουμε τις συστάσεις θα ήταν να πάρουμε ένα στιγμιότυπο του γραφήματος την χρονική στιγμή t+1 και να δούμε εάν οι προτάσεις έγιναν αποδεκτές από τους χρήστες και σε τι ποσοστό. Μιας και αυτό όμως δεν είναι εφικτό τότε θα τους αξιολογήσουμε με βάση τις ήδη υπάρχουσες φιλίες που έχουν αναπτυχθεί.

Για να αξιολογήσουμε την απόδοση της κάθε μεθόδου θα εισάγουμε στο πρόγραμμα που χρησιμοποιήθηκε και στο παραπάνω ερώτημα με την διαφορά ότι αυτή τη φορά ενδιαφερόμαστε για την θέση στην οποία προτείνεται ο κάθε χρήστης στην λίστα του άλλου. Επίσης για λόγους συντόμευσης, αυτή τη φορά δεν θα εξετάσουμε τις κορυφαίες 10 προτάσεις για κάθε χρήστη αλλά όλη τη λίστα. Τα αποτελέσματα φαίνονται στον παρακάτω πίνακα (Πίνακας 15)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | FI-F2 | FoF  avg rec. position | Jaccard avg rec position | Adamic avg rec position |  | FI-F2 | FoF  avg rec. position | Jaccard avg rec position | Adamic avg rec position |
| 1 | 0-184 | 1,0 | 12,0 | 1,0 | 51 | 2544-1916 | 96,0 | 57,5 | 97,5 |
| 2 | 1122-1408 | 16,5 | 11,5 | 16,5 | 52 | 2551-2370 | 16,5 | 16,5 | 16,5 |
| 3 | 1123-1911 | 206,0 | 190,0 | 184,0 | 53 | 2590-1993 | 1,0 | 1,0 | 1,0 |
| 4 | 1161-1038 | 74,5 | 69,0 | 56,0 | 54 | 2590-2063 | 3,5 | 2,5 | 3,0 |
| 5 | 1245-1848 | 15,5 | 9,0 | 14,0 | 55 | 2593-2121 | 2,5 | 3,5 | 2,5 |
| 6 | 1260-1876 | 1,5 | 1,5 | 1,5 | 56 | 2615-2142 | 1,0 | 1,5 | 1,0 |
| 7 | 1279-1611 | 8,0 | 3,0 | 8,0 | 57 | 2616-2336 | 5,5 | 7,0 | 5,0 |
| 8 | 1298-1903 | 6,0 | 4,5 | 6,0 | 58 | 2619-2220 | 1,5 | 1,5 | 1,5 |
| 9 | 1302-1125 | 11,5 | 4,0 | 11,0 | 59 | 2624-2381 | 1,5 | 1,5 | 2,0 |
| 10 | 1523-1674 | 209,0 | 206,5 | 186,0 | 60 | 2661-3344 | 1,5 | 1,0 | 1,5 |
| 11 | 1530-1598 | 24,5 | 18,5 | 24,5 | 61 | 2664-2986 | 2,0 | 1,5 | 2,0 |
| 12 | 1563-1559 | 4,0 | 2,0 | 4,0 | 62 | 2689-3022 | 17,0 | 15,5 | 17,0 |
| 13 | 1574-517 | 5,5 | 15,0 | 4,5 | 63 | 2701-2937 | 18,5 | 2,5 | 14,5 |
| 14 | 158-315 | 24,5 | 36,5 | 23,0 | 64 | 2742-3267 | 2,5 | 4,0 | 3,0 |
| 15 | 1587-1742 | 30,0 | 32,5 | 31,0 | 65 | 2742-3344 | 2,0 | 2,0 | 2,0 |
| 16 | 1592-1628 | 17,5 | 22,5 | 16,0 | 66 | 2832-2668 | 67,5 | 39,5 | 63,0 |
| 17 | 1655-1728 | 19,0 | 11,5 | 16,0 | 67 | 2866-2912 | 3,0 | 1,0 | 3,0 |
| 18 | 1684-2851 | 1,0 | 51,0 | 1,0 | 68 | 2872-2912 | 3,0 | 1,0 | 3,0 |
| 19 | 1730-1367 | 1,0 | 1,0 | 1,0 | 69 | 2939-2901 | 6,0 | 4,0 | 7,0 |
| 20 | 1850-483 | 60,0 | 104,5 | 54,0 | 70 | 3015-1684 | 1,0 | 106,0 | 1,0 |
| 21 | 1864-1528 | 57,0 | 27,5 | 53,5 | 71 | 3036-2925 | 92,0 | 114,0 | 91,5 |
| 22 | 1917-2409 | 2,0 | 2,0 | 1,5 | 72 | 3132-3082 | 1,0 | 1,0 | 1,0 |
| 23 | 1920-1939 | 26,5 | 25,0 | 27,0 | 73 | 313-60 | 57,0 | 101,0 | 55,0 |
| 24 | 2020-2428 | 3,0 | 3,5 | 3,0 | 74 | 3199-3172 | 27,5 | 12,0 | 26,5 |
| 25 | 2026-2607 | 89,0 | 118,5 | 85,0 | 75 | 3238-3166 | 45,0 | 8,0 | 39,5 |
| 26 | 2060-2266 | 2,0 | 32,5 | 2,0 | 76 | 3280-3288 | 8,5 | 11,0 | 7,5 |
| 27 | 2094-2487 | 1,0 | 1,0 | 1,0 | 77 | 3343-3411 | 166,0 | 241,5 | 115,0 |
| 28 | 2142-1966 | 1,0 | 1,0 | 1,0 | 78 | 3348-2909 | 5,0 | 5,0 | 5,5 |
| 29 | 2142-2303 | 11,0 | 24,0 | 11,0 | 79 | 3357-2712 | 111,5 | 94,5 | 111,5 |
| 30 | 2174-2151 | 13,0 | 4,0 | 13,0 | 80 | 3435-2827 | 46,0 | 51,0 | 43,0 |
| 31 | 2183-2002 | 13,5 | 10,0 | 13,5 | 81 | 3443-3909 | 52,5 | 55,5 | 55,5 |
| 32 | 2197-2091 | 1,0 | 1,0 | 1,0 | 82 | 348-551 | 5,5 | 8,0 | 5,5 |
| 33 | 2198-2319 | 35,5 | 38,5 | 37,0 | 83 | 3577-3841 | 1,0 | 1,0 | 1,0 |
| 34 | 2200-1953 | 14,5 | 9,5 | 14,0 | 84 | 3583-3931 | 58,0 | 52,5 | 55,5 |
| 35 | 2225-2627 | 6,0 | 1,0 | 4,5 | 85 | 3726-3861 | 413,5 | 322,5 | 413,5 |
| 36 | 2240-2586 | 1,0 | 1,0 | 1,0 | 86 | 375-404 | 16,5 | 5,0 | 16,0 |
| 37 | 2245-2281 | 1,0 | 1,0 | 1,0 | 87 | 3770-3671 | 27,0 | 2,0 | 24,0 |
| 38 | 2257-1983 | 4,0 | 7,5 | 4,5 | 88 | 3884-3737 | 13,5 | 11,0 | 13,5 |
| 39 | 2266-1953 | 1,5 | 19,0 | 1,5 | 89 | 3902-3488 | 1,0 | 1,0 | 1,0 |
| 40 | 2303-2391 | 96,5 | 76,0 | 87,5 | 90 | 391-515 | 2,5 | 1,0 | 2,5 |
| 41 | 2309-1971 | 2,0 | 2,0 | 2,0 | 91 | 402-425 | 39,5 | 35,0 | 41,5 |
| 42 | 2324-2302 | 76,0 | 104,5 | 68,0 | 92 | 483-1392 | 67,0 | 98,5 | 67,0 |
| 43 | 2350-2545 | 105,0 | 1,0 | 102,5 | 93 | 504-539 | 33,5 | 13,0 | 17,0 |
| 44 | 2356-2266 | 1,0 | 1,5 | 1,0 | 94 | 573-644 | 1,0 | 1,5 | 1,0 |
| 45 | 2374-2276 | 8,5 | 8,0 | 8,5 | 95 | 574-630 | 5,5 | 7,5 | 5,5 |
| 46 | 2398-2384 | 1,5 | 2,0 | 1,5 | 96 | 713-731 | 1,0 | 4,0 | 1,5 |
| 47 | 2409-2624 | 1,5 | 1,0 | 1,5 | 97 | 721-736 | 52,5 | 25,0 | 63,5 |
| 48 | 2476-2234 | 109,0 | 3,5 | 82,5 | 98 | 724-824 | 1,0 | 1,0 | 1,0 |
| 49 | 2521-1912 | 1,0 | 37,0 | 1,0 | 99 | 840-780 | 3,0 | 1,5 | 2,5 |
| 50 | 2526-2056 | 19,5 | 17,5 | 16,5 | 100 | 915-1844 | 7,0 | 5,5 | 6,5 |

Πίνακας 15: Ο πίνακας απεικονίζει την μέση θέση στην τελική λίστα σύστασης για τους 3 αλγόριθμους για 100 τυχαία ζεύγη του γραφήματος.

Όπως είναι ξεκάθαρο, στις περισσότερες των περιπτώσεων η μέση θέση στην τελική λίστα σύστασης της μεθόδου Jaccard είναι σε υψηλότερη θέση. Για την ακρίβεια αυτό συνέβη σε 43 περιπτώσεις, πράγμα που δηλώνει ότι έχει την καλύτερη εφαρμογή στα δεδομένα μας. Αντιθέτως η μέθοδος Adamic and Adar πρότεινε 17 φορές σε καλύτερη θέση ενώ η FoF μόλις σε 6. Όπως είναι φανερό υπάρχουν και αρκετές ισοπαλίες μεταξύ των μεθόδων. Σε αυτήν την περίπτωση δεν θεωρούμε ότι υπερτερεί κάποια μέθοδος σε σχέση με την άλλη και για τον λόγο αυτό δεν προσμετράτε στα παραπάνω αποτελέσματα.

Για το τέλος, θα θέλαμε να εξετάσουμε άλλη μία μέθοδο για την προσέγγιση του προβλήματος. Η μέθοδο; Αυτή ονομάζεται **resource\_allocation\_index** και υπολογίζεται από την συνάρτηση:

𝒔𝒄𝒐𝒓𝒆 (𝑨, 𝑩) **= ∑w∈Γ(u)∩Γ(v)=1/| 𝑵𝑪 |** (5)

Αυτή η μέθοδο; Μοιάζει αρκετά με την Adamic and Adar έχει όμως την διαφορά ότι αυτή τη φορά θεωρούμε ότι κάθε κοινός φίλος μεταξύ των χρηστών Α και Β, ο οποίος παίζει και εδώ το ρόλο του του μεταδότη, μοιράζει τους πόρους-πληροφορίες ισόποσα στους Α και Β. Με λίγα λόγια δεν δίνεται μεγαλύτερη βαρύτητα στους κοινούς τους φίλους οι οποίοι εμφανίζονται πιο σπάνιοι μέσα στο δίκτυο. Εφαρμόζοντας ξανά τον κώδικα από το ερώτημα 5 βλέπουμε ότι το μέσο ποσοστό συμφωνίας μεταξύ αυτής της μεθόδου με τις υπόλοιπες είναι 82,5% με την ΑΑ, 60% με την Jaccard και 55,5% με την FoF. Τέλος, δοκιμάζοντας πάλι και τις 4 μεθόδους αυτή τη φορά, σε ένα τυχαίο δείγμα 100 τυχαίων πραγματικών φίλων βλέπουμε ότι η resource allocation index έχει αρκετά καλά αποτελέσματα. Πιο αναλυτικά σε 27 περιπτώσεις η μέση θέση στην τελική λίστα συστάσεων ήταν υψηλότερη από τις άλλες μεθόδους. Αντιθέτως η Adamic το κατάφερε σε 32, ενώ η Jaccard εξακολουθεί να βρίσκεται στην πρώτη θέση με 40. Είναι αξιόλογο ότι η FoF δεν κατάφερε σε καμία των περιπτώσεων να προτείνει την εκάστοτε σύσταση σε υψηλότερη θέση σε σχέση με τις υπόλοιπες.

Συμπεράσματα

Ως συμπέρασμά θα μπορούσαμε να πούμε ότι για τα δεδομένα αυτού του γραφήματος οι αλγόριθμοι Jaccard Coefficient και Adamic and Adam έχουν τον μεγαλύτερο βαθμό συμφωνίας στις τελικές προτάσεις συστάσεων. Το ποσοστό των κοινών προτάσεων αγγίζει το 90,7% και είναι μακράν το μεγαλύτερο σε σύγκριση με τις άλλες μεθόδους. Όσο αφορά την αξιολόγηση των μεθόδων φαίνεται ότι η Jaccard ανταποκρίνεται καλύτερα στα δεδομένα. Για την ακρίβεια μετά τον έλεγχο 100 τυχαίων πραγματικών φιλιών του γραφήματος, ήταν αυτή που τις περισσότερες φορές τις συμπεριέλαβε στη υψηλότερη θέση της τελικής λίστας συστάσεων και μάλιστα με μεγάλη διαφορά σε σχέση με την Adamic and Adam. Φαίνεται λοιπόν ότι παρόλο το μεγάλο ποσοστό συμφωνίας των δύο αλγόριθμων στις τελικές συστάσεις, η Jaccard είναι αυτή που κατά κανόνα προτείνει την ίδια σύσταση σε πιο υψηλή θέση.