ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Αναπαράσταση Τεχνικών Μηχανικής Μάθησης για Σύσταση Αντικειμένων στην Άλγεβρα Μονοπατιών RecGraph

Θεώνη Ε. Παλαιολόγου

Επιβλέποντες: **Ιωάννης Ιωαννίδης,** Καθηγητής

ΑΘΗΝΑ

Οκτώβριος 2021

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Αναπαράσταση Τεχνικών Μηχανικής Μάθησης για Σύσταση Αντικειμένων στην Άλγεβρα Μονοπατιών RecGraph

Θεώνη Ε. Παλαιολόγου

Α.Μ.: 1115201700201

ΕΠΙΒΛΕΠΟΝΤΕΣ: **Ιωάννης Ιωαννίδης,** Καθηγητής

ΠΕΡΙΛΗΨΗ

Ένα σύστημα συστάσεων είναι μια υποκατηγορία συστημάτων φιλτραρίσματος πληροφοριών που προτείνει περιεχόμενο στους χρήστες το οποίο είναι σχετικό με τα ενδιαφέροντά τους. Μελετώντας τον άναρχο χώρο των συστημάτων συστάσεων, την αφθονία των διαθέσιμων διαφορετικών προσεγγίσεων συστάσεων, τα προβλήματα που σχετίζονται με τον περίπλοκο, ετερογενή τομέα δεδομένων και τις συνέπειες αυτής της πολυπλοκότητας τόσο για τον χρήστη όσο και για το σύστημα, προτάθηκε το αλγεβρικό μοντέλο RecGraph. Το μοντέλο RecGraph συνιστά μία διαφορετική προσέγγιση στο πρόβλημα των συστάσεων η οποία επαναπροσδιορίζει την σύσταση ως ένα πρόβλημα υπολογισμού μονοπατιού σε ένα μοντέλο δεδομένων γραφήματος, επιτρέποντας την αναπαράσταση μίας ποικιλίας αλγορίθμων συστάσεων με ομογενοποιημένο τρόπο.

Στο πλαίσιο της πτυχιακής εργασίας θα εξετάσουμε την εκφραστικότητα της γλώσσας RecGraph και την καταλληλότητά της για την αναπαράσταση αλγορίθμων συστάσεων οι οποίοι βασίζονται σε τεχνικές μηχανικής μάθησης. Θα επικεντρωθούμε σε τεχνικές μηχανικής μάθησης για σύσταση με ένα κρυμμένο επίπεδο νευρώνων και θα δείξουμε την αναγωγή τους στην γλώσσα RecGraph. Η συγκεκριμένη αναγωγή αποδεικνύει ότι το σύστημα RecGraph μπορεί να περιγράψει, με τις κατάλληλες επεκτάσεις, ένα εύρος αλγορίθμων συστάσεων.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Συστήματα συστάσεων

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Συστήματα συστάσεων, Βάσεις Δεδομένων Γράφων, Αλγόριθμος Funk-Αποσύνθεσης ιδιαζουσών τιμών, Μηχανική Μάθηση

ABSTRACT

A recommendation system is a subclass of information filtering systems that offer content to users that is relevant to their interests. Studying the anarchic recommendation systems space, the abundance of different recommendation approaches available, the problems associated with the complex, heterogeneous data domain it requires, and the consequences of this complexity for both the user and the system: the algebraic model “RecGraph” was proposed. A different approach to the problem of recommendations which redefines the recommendation as a path computational problem in a graph data model. In this thesis we will examine the expressivity of the RecGraph language and its suitability for representing algorithmic recommendations based on machine learning techniques. We will focus on machine learning techniques for recommendation with a hidden level of neurons and we will show their reduction in RecGraph language. This reduction proves that the RecGraph system can describe, with the appropriate extensions, a range of recommendation algorithms.

SUBJECT AREA: Recommender Systems

KEYWORDS: Recommender Systems, Graph Databases, Funk-Singular Value Decomposition, Machine Learning

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου, Καθ. Ιωάννη Ιωαννίδη, και τους συνεργάτες της ομάδας MaDgIK [21], Θεόφιλο Μαΐλη, Ιωάννη Φούφουλα και Μαριαλένα Κυριακίδη για την άψογη καθοδήγηση και βοήθειά τους κατά την εκπόνηση της παρούσας Πτυχιακής Εργασίας.

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

1. ΕΙΣΑΓΩΓΗ

1.1 Εισαγωγή στις συστάσεις

Οι συστάσεις αποτελούν το δομικό στοιχείο διαφόρων εμπορικών εφαρμογών για την εξατομικευμένη και προσαρμοσμένη εμπειρία του χρήστη σε αυτές, ενώ ολοένα και περισσότερες τέτοιες εφαρμογές υιοθετούν συστήματα συστάσεων. Για παράδειγμα η εφαρμογή Spotify, μία υπηρεσία ροής μουσικού περιεχομένου, θα προτείνει στον χρήστη μουσικό περιεχόμενο βάσει των προηγούμενων προτιμήσεών του, η πλατφόρμα της Amazon για αγορά προϊόντων θα προτείνει στον χρήστη νέα προϊόντα βάσει προηγούμενων αναζητήσεων και αγορών του, ενώ ακόμα και εφαρμογές κοινωνικής δικτύωσης όπως το Facebook, το Youtube, και το Tictok, προσαρμόζουν τόσο το περιεχόμενο όσο και τις διαφημίσεις ανάλογα με τις αλληλεπιδράσεις των χρηστών τους εντός των συστημάτων αυτών. Ως αποτέλεσμα το τοπίο των συστημάτων συστάσεων εξελίσσεται ραγδαία και γίνεται ολοένα και πιο ποικιλόμορφο στις τρεις ακόλουθες διαστάσεις:

- **Χώρος πληροφοριών:** Ένας πλούτος διαφορετικών τύπων μεμονωμένων ή πολύπλοκων οντοτήτων και συσχετίσεων μεταξύ τους μπορεί να χρησιμοποιηθεί ως πληροφορία αναφοράς για τη δημιουργία συστάσεων σε μια εφαρμογή. Οι χρήστες αφήνουν μεγάλη ποικιλία διαδικτυακών ιχνών και τα διαθέσιμα δεδομένα είναι ετερογενή, αλληλένδετα, ενώ αυξάνονται σημαντικά σε μέγεθος. Για παράδειγμα το Facebook έχει 2.89 δισεκατομμύρια χρήστες και αναλύει διαφορετικού τύπου πληροφορίες οι οποίες σχετίζονται με τον γράφο των φίλων ενός χρήστη, τις αναρτήσεις του καθώς και τις αναρτήσεις για τις οποίες έδειξε ενδιαφέρον, καθώς και το ποιες διαφημίσεις επισκέπτεται ο κάθε χρήστης.
- **Στόχος:** Ακόμη και στον ίδιο χώρο πληροφοριών, διαφορετικοί τύποι οντοτήτων μπορεί να παίζουν το ρόλο υποκειμένων συστάσεων (λήψη συστάσεων) και αντικειμένων συστάσεων (που συνιστάται) για διαφορετικούς σκοπούς. Κάποιος μπορεί να προτείνει ένα ξενοδοχείο σε μια ομάδα ταξιδιωτών ή ένα πακέτο αεροπορικά-ξενοδοχείο-αυτοκίνητο σε έναν ταξιδιώτη (πωλήσεις), χρήστες μεταξύ τους (κοινωνική δικτύωση), ένα πιθανό νέο μέλος σε ένα γυμναστήριο (στοχευμένη διαφήμιση) ή έναν ηθοποιό σε μια ταινία (κάστινγκ).
- **Μηχανισμός:** Ένα πλούσιο σύνολο παλιών και αναδυόμενων αλγορίθμων μπορεί να χρησιμοποιηθεί για τη δημιουργία προτάσεων, από το συμβατικό φιλτράρισμα με βάση το περιεχόμενο και τη συνεργασία [8] έως την παραγοντοποίηση πινάκων (matrix factorization) [9] και τη βαθιά μάθηση (deep learning) [10].

Η αυξανόμενη ποικιλομορφία σε αυτές τις τρεις διαστάσεις έχει οδηγήσει στην ανάπτυξη ενός τεράστιου αριθμού αλγορίθμων συστάσεων που αντιμετωπίζουν φαινομενικά διαφορετικά προβλήματα με φαινομενικά διαφορετικές λύσεις. Κάθε περίπτωση αντιμετωπίζεται ως ξεχωριστή και από την αρχή, με έναν συγκεκριμένο αλγόριθμο που παρέχει μια προσαρμοσμένη λύση για μια συγκεκριμένη εργασία εντός ενός συγκεκριμένου τομέα και χώρου πληροφοριών. Ως εκ τούτου, τα συστήματα που προκύπτουν προσφέρουν περιορισμένες λύσεις που δεν μπορούν να επαναχρησιμοποιηθούν, να γενικευτούν, ή να προσαρμοστούν εύκολα σε διαφορετικές ή νέες παραλλαγές ενός προβλήματος. Αυτή η αποσπασματική (ακόμα και ελλιπής) μεθοδολογία ως προς τις συστάσεις είναι προβληματική για πολλούς λόγους: Πολλές προσπάθειες αναπαράγονται άσκοπα ξανά και ξανά· Οποιεσδήποτε πιθανές

βελτιστοποιήσεις απόδοσης πρέπει να γίνονται χειροκίνητα στον κώδικα που έχει αναπτυχθεί για τη κάθε περίπτωση· η πειραματική σύγκριση και αξιολόγηση διαφορετικών αλγορίθμων εμποδίζεται από τον τεράστιο αριθμό επιλογών και την επακόλουθη αβεβαιότητα για την ποιότητα των αποτελεσμάτων. Ο συνδυασμός διαφορετικών τύπων πληροφοριών υποβάθρου, θεμάτων και αντικειμένων συστάσεων και αλγορίθμων δημιουργεί ένα τεράστιο σύμπαν από δυνατότητες συστάσεων, των οποίων η συστηματική εξερεύνηση, εφαρμογή και διαχείριση είναι πολύ απαιτητική εργασία.

1.2 Το μοντέλο RecGraph

Οι τρέχουσες ερευνητικές προσπάθειες επικεντρώνονται στην προώθηση μεμονωμένων αλγοριθμικών τεχνικών, αγνοώντας την αλλαγή ρυθμίσεων στις σημερινές πιο απαιτητικές εφαρμογές.

Μελετώντας τον άναρχο χώρο των συστημάτων συστάσεων, την αφθονία των διαθέσιμων διαφορετικών προσεγγίσεων συστάσεων, τα προβλήματα που σχετίζονται με τον περίπλοκο, ετερογενή τομέα δεδομένων που απαιτεί και τις συνέπειες αυτής της πολυπλοκότητας τόσο για τον χρήστη όσο και για το σύστημα: προτάθηκε το αλγεβρικό μοντέλο RecGraph (Kyriakidi et al.) [6][7], μία διαφορετική προσέγγιση στο πρόβλημα των συστάσεων, το οποίο επαναπροσδιορίζει την σύσταση ως ένα πρόβλημα υπολογισμού μονοπατιού σε ένα μοντέλο δεδομένων γραφήματος. Το RecGraph μία γράφο-θεωρητική προσέγγιση σε συστάσεις, που αποτελείται από ένα μοντέλο δεδομένων και ένα μοντέλο υπολογισμού. Το μοντέλο δεδομένων είναι ένα γράφημα όπου:

- Οι κόμβοι είναι οντότητες σχετικές με τον τομέα, συμπεριλαμβανομένων εκείνων που μπορούν να προταθούν ή να λάβουν συστάσεις.

- Οι ακμές καταγράφουν ποικίλες σχέσεις, όπως συνδέσεις συγκεκριμένου τομέα (τμήμα-από, συσχετισμένο-με κ.α), καταστάσεις (φίλος-του, παρόμοια-με κ.α.) ενέργειες (βαθμολογημένες, επισκέψεις, κ.α.), και στάσεις και προτιμήσεις (μου αρέσει, εμπιστεύεται, έχει σημασία, πιστεύει κ.α.).

Το υπολογιστικό μοντέλο βασίζεται σε μια άλγεβρα μονοπατιών: Εμπνευσμένο από προηγούμενες εργασίες για άλγεβρες μονοπατιών [11], όλη η επεξεργασία εκφράζεται χρησιμοποιώντας ένα πολύ μικρό σύνολο γενικών τελεστών μονοπατιών, οι οποίοι εγκαθίστανται σε συγκεκριμένες συναρτήσεις ανάλογα με τη στρατηγική σύστασης και το υπό εξέταση πεδίο ενδιαφέροντος. Η φύση τους ακολουθεί ένα παράδειγμα σειρών-παράλληλων, όπου ορισμένοι τελεστές συνθέτουν (compose) διαδοχικές ακμές κατά μήκος ενός μονοπατιού και μερικές συναθροίζουν (aggregate) ή συγχωνεύουν (fuse) ακμές μεταξύ των ίδιων κόμβων. Συσχετίζεται με την άλγεβρα μονοπατιών μια δηλωτική γλώσσα ερωτημάτων τύπου Datalog που χρησιμοποιεί αυτούς τους τελεστές για να γράψει ομοιόμορφα αλγόριθμους συστάσεων, ως παράγωγα μονοπάτια νέων ακμών που σχετίζονται με συστάσεις. Οι τελεστές μονοπατιών μπορούν να επιδείξουν ένα επιθυμητό σύνολο ιδιοτήτων, που μπορεί να οδηγήσει σε ευκαιρίες βελτιστοποίησης της απόδοσης.

Το θεμελιώδες πλεονέκτημα του RecGraph [6][7] είναι ότι μπορεί κανείς να αναπτύξει ένα ενιαίο, τελειοποιημένο, γενικό σύστημα σύστασης πάνω από μια μηχανή γραφημάτων και, στη συνέχεια, να το τοποθετήσει σε πολλές ειδικές συστάσεις μέσω

της κατάλληλης παραμετροποίησης. Η ομοιομορφία στη σύλληψη των συστάσεων που παρέχονται με αυτόν τον τρόπο έχει πολλά πλεονεκτήματα.

a) **Εκφραστική ισχύς:** Οι αφαιρέσεις του μοντέλου δεδομένων γραφήματος μπορούν να συλλάβουν, να εξερευνήσουν και να παρακολουθήσουν την εξέλιξη διαφορετικών ετερογενών χώρων πληροφοριών. Οι αφαιρέσεις του μοντέλου υπολογισμού μονοπατιών μπορούν να συλλάβουν διαφορετικούς στόχους συστάσεων και μια μεγάλη ποικιλία αλγορίθμων χρησιμοποιώντας την ίδια ομοιόμορφη λογική και λίγους τελεστές.

b) **Ευχρηστία:** Η γλώσσα RecGraph βοηθά τους σχεδιαστές να επικεντρωθούν περισσότερο στη λογική των συστάσεων παρά στις λεπτομέρειες υλοποίησης, κάτι που δυνητικά οδηγεί σε καλύτερους και/ή καθαρότερους αλγόριθμους και ταχύτερη ανάπτυξη. Ένας αλγόριθμος μπορεί να δηλωθεί ευκολότερα σε μία δηλωτική γλώσσα. Έτσι, η χρήση της γλώσσας ερωτήματός μας οδηγεί σε μειωμένο σχεδιασμό, χρόνο επίλυσης προβλημάτων και γενικά σε ταχύτερη ανάπτυξη.

c) **Ταχύτητα επεξεργασίας:** Η χρήση μιας υπερσύγχρονης μηχανής γραφημάτων προσφέρει αξιοπιστία και υπόσχεται σημαντικές βελτιώσεις στην απόδοση, καθώς οι βάσεις δεδομένων γραφημάτων έχουν σχεδιαστεί για επεκτασιμότητα δεδομένων. Επιπλέον, ο βελτιστοποιητής (optimizer) μιας μηχανής γραφήματος μπορεί να εκμεταλλευτεί τις μαθηματικές ιδιότητες των τελεστών RecGraph για να επιλέξει βέλτιστες εκτελέσεις ανεξάρτητα από τον τρόπο γραφής τους.

1.3 RecGraph & Αλγόριθμοι Μηχανικής Μάθησης

Καθώς ένας σημαντικός αριθμός αλγορίθμων συστάσεων βασίζεται σε τεχνικές μηχανική μάθησης, έχει ξεχωριστό ενδιαφέρον να εξετάσουμε τον βαθμό στον οποίο το μοντέλο και η γλώσσα RecGraph θα μπορούσαν να περιγράψουν και να ενσωματώσουν ενδογενώς τέτοιους αλγορίθμους καθώς και να μελετήσουμε τις επεκτάσεις της αρχικής γλώσσας που θα μας δώσουν την δυνατότητα προκειμένου να εκφράσουμε ένα πολύπλοκο αλγόριθμο μηχανικής μάθησης στην συγκεκριμένη γλώσσα.

1.4 Στόχος

Στο πλαίσιο της πτυχιακής εργασίας θα εξετάσουμε την εκφραστικότητα της γλώσσας RecGraph και την καταλληλότητά της για την αναπαράσταση αλγορίθμων συστάσεων οι οποίοι βασίζονται σε τεχνικές μηχανικής μάθησης. Θα επικεντρωθούμε σε τεχνικές μηχανική μάθησης για σύσταση με ένα κρυμμένο επίπεδο νευρώνων και θα δείξουμε την αναγωγή τους στην γλώσσα RecGraph. Η συγκεκριμένη αναγωγή αποδεικνύει ότι το σύστημα RecGraph δύναται να περιγράψει, με τις κατάλληλες επεκτάσεις, ένα εύρος αλγορίθμων συστάσεων. Πιο συγκεκριμένα στα πλαίσια της πτυχιακής:

- Μελετήσαμε τον αλγόριθμο Funk-Αποσύνθεσης Ιδιαζουσών Τιμών (Funk-SVD) ο οποίος επιτρέπει να προβλέψουμε βαθμολογίες για κάθε ζεύγος χρήστη-αντικειμένου, βασισμένοι στις υπάρχουσες βαθμολογίες μεταξύ χρηστών-αντικειμένων. Ο αλγόριθμος Funk-SVD, χρησιμοποιεί τεχνικές μηχανικής μάθησης για να επιλύσει το συγκεκριμένο πρόβλημα σύστασης και συγκεκριμένα χρησιμοποιεί ένα κρυμμένο επίπεδο νευρώνων που περιγράφουν τα λανθάνοντα (κρυμμένα) χαρακτηριστικά των αντικειμένων.

- Εξετάσαμε την αναγωγή του αλγορίθμου Funk-SVD στην γλώσσα RecGraph, δείχνοντας ότι κάθε στάδιο εκπαίδευσης του αλγορίθμου μπορεί να αναπαρασταθεί ως

μία ακολουθία αλγεβρικών πράξεων χρησιμοποιώντας τους τελεστές της γλώσσας RecGraph στους κόμβους και τις ακμές που περιγράφουν την γνώση μας.

- Επεκτείναμε την αρχική υλοποίηση της γλώσσας RecGraph με τις κατάλληλες συναρτήσεις για Concatenation, Aggregation και Fusion, οι οποίες επιτρέπουν να προσαρμόσουμε τα βάρη των κρυμμένων ακμών βάσει του αλγορίθμου Funk-SVD. Συγκεκριμένα, ο αλγόριθμος Funk-SVD χρησιμοποιεί την μέθοδο ελάττωσης της παραγώγου (gradient descent) για κάθε εποχή εκπαίδευσης των λανθάνοντων χαρακτηριστικών. Η συγκεκριμένη μέθοδος μπορεί να αναπαρασταθεί ως μία ακολουθία από πράξεις Concatenation, Aggregation και Fusion.

- Εξετάσαμε πειραματικά την ορθότητα της αναγωγής τους αλγορίθμου Funk-SVD στην Άλγεβρα Μονοπατιών RecGraph. Συγκεκριμένα εξετάζουμε τα δεδομένα του dataset Movielens το οποίο περιγράφει αξιολογήσεις χρηστών για ταινίες. Χωρίσαμε τα δεδομένα μας σε ένα σύνολο εκπαίδευσης και σε ένα σύνολο πρόβλεψης, τα οποία χρησιμοποιήθηκαν αντίστοιχα για την προσαρμογή των παραμέτρων του Funk-SVD αλγορίθμου και την αξιολόγηση της εκπαίδευσης σε πραγματικά δεδομένα βαθμολόγησης ταινιών. Παρατηρήσαμε ότι με την επιλογή των κατάλληλων παραμέτρων εκπαίδευσης, η αναγωγή στην RecGraph γλώσσα παράγει τα ίδια αποτελέσματα με τον αρχικό Funk-SVD.

2. ΣΥΣΤΗΜΑΤΑ ΣΥΣΤΑΣΕΩΝ

Τα Συστήματα Συστάσεων (Recommender Systems) είναι εργαλεία και τεχνικές λογισμικού που στοχεύουν στην παροχή προτάσεων στοιχείων στο χρήστη (user) ενός συστήματος [8]. Ένα αντικείμενο (item) είναι ένας γενικός όρος που χρησιμοποιείται για να δηλώσει οτιδήποτε μπορεί να προταθεί σε έναν χρήστη. Ανάλογα με τη δεδομένη εφαρμογή, τα αντικείμενα μπορεί να είναι ποικιλόμορφα από βιβλία, τραγούδια, ταινίες, ειδήσεις, άτομα, εστιατόρια, ερωτήματα, ενέργειες εξερεύνησης, σχέδια ερωτημάτων, ροές εργασιών κ.α. Τα συστήματα συστάσεων είναι ένα πολυεπιστημονικό πεδίο. Περιλαμβάνει αλγόριθμους από μηχανική μάθηση, ανάκτηση πληροφοριών, ανάλυση κοινωνικών δικτύων, θεωρία πιθανοτήτων και στατιστικής, εξόρυξη δεδομένων κ.α.

Σε αυτό το κεφάλαιο περιγράφουμε το πως ορίζεται γενικά μια άλγεβρα μονοπατιών και ποιος είναι ο μαθηματικός φορμαλισμός της. Περιγράφουμε τους μαθηματικούς τελεστές που περιλαμβάνει ένα τέτοιο μοντέλο και τις ιδιότητες που έχουν. Ορίζουμε προβλήματος σύσταση και αναφέρουμε τους κλασικούς αλγόριθμους συστάσεων. Τέλος, περιγράφεται τι είναι μια βάση δεδομένων γράφου, ποια είναι τα βασικά τμήματα από τα οποία αποτελείται, ποιες οι διαφορές από τις κλασικές σχεσιακές βάσεις δεδομένων και τι πλεονεκτήματά μας προσφέρει η χρήση της.

2.1 Άλγεβρα Μονοπατιών

Πολλά προβλήματα που αντιμετωπίζουμε συχνά μπορούν να αναχθούν σε προβλήματα γραφημάτων (graph problems) . Ο τρόπος αντιστοίχισης προβλημάτων σε προβλήματα θεωρίας γραφημάτων μπορεί να μην είναι πάντα προφανής, αλλά αυτό είναι συχνά μια καλή στρατηγική για την επίλυση ή την απλή κατανόησή τους, καθώς τα γραφήματα είναι πολύ ισχυρά μοντέλα. Τα γραφήματα έρχονται σε πολλές διαφορετικές μορφές και με διάφορα ενδιαφέροντα χαρακτηριστικά. Ένα από τα πιο απλά χαρακτηριστικά ενός γραφήματος είναι ένα μονοπάτι (path). Γενικά, ένα μονοπάτι μεταξύ δύο κόμβων (nodes) υπάρχει εάν μπορείτε να συνδέσετε αυτούς τους κόμβους διασχίζοντας (traversing) ένα σύνολο ακμών (edges). Τα μονοπάτια αποτελούν συχνά αντικείμενο ενδιαφέροντος στα γραφήματα. Τα προβλήματα που αφορούν τον καθορισμό των μονοπατιών μπορούν να λάβουν διαφορετικές μορφές. Οι κοινές κατηγορίες περιλαμβάνουν «ακραία», δηλαδή όπου οι ακμές του γραφήματος συνδέονται με έναν πραγματικό αριθμό, και προβλήματα «απαρίθμησης», δηλαδή την εύρεση στοιχειωδών μονοπατιών από τον έναν κόμβο στον άλλο. Μερικά γνωστά παραδείγματα σε αυτές τις κατηγορίες περιλαμβάνουν προβλήματα συντομότερου μονοπατιού (shortest path), συνδεσιμότητας / προσβασιμότητας, ισχυρά συνδεδεμένα στοιχεία (components), μέγιστη ροή δικτύου και προβλήματα ελάχιστου επικαλύπτον δέντρου (spanning tree).

Μια αλγεβρική δομή, δηλαδή μια άλγεβρα μονοπατιών, εισήχθη [11] ως τρόπος διατύπωσης και επίλυσης μιας μεγάλης ποικιλίας προβλημάτων μονοπατιών. Τα γραφήματα θεωρούνταν επισημασμένα (labeled) με πραγματικούς αριθμούς ή λέξεις σε κάποιο αλφάβητο, που ήταν στοιχεία της άλγεβρας μονοπατιών. Στη συνέχεια αποδείχθηκε ότι πολλά προβλήματα μονοπατιών θα μπορούσαν να τεθούν ως επίλυση ενός συνόλου εξισώσεων στην άλγεβρα μονοπατιών. Η παραγωγή άμεσων και επαναληπτικών μεθόδων για την επίλυση τέτοιων εξισώσεων οδήγησε σε διαφορετικούς αλγόριθμους εύρεσης μονοπατιών.

2.1.1 Ορισμός Άλγεβρας Μονοπατιών

Μια άλγεβρα μονοπατιών είναι ένα σύνολο P εξοπλισμένο με δύο δυαδικές πράξεις ∨ και · που έχουν τις ακόλουθες ιδιότητες.

- Η πράξη ∨ είναι αυτοδύναμη, αντιμεταθετική και προσεταιριστική:
  - $x \vee x = x$ για όλα τα $x \in P$
  - $x \vee y = y \vee x$ για όλα τα $x, y \in P$
  - $(x \vee y) \vee z = x \vee (y \vee z)$ για όλα τα $x, y, z \in P$
- Η πράξη · είναι προσεταιριστική και επιμεριστική στο ∨:
  - $(x \cdot y) \cdot z = x \cdot (y \cdot z)$ για όλα τα $x, y, z \in P$
  - $x \cdot (y \vee z) = (x \cdot y) \vee (x \cdot z)$ και $(y \vee z) \cdot x = (y \cdot x) \vee (z \cdot x)$ για όλα τα $x, y, z \in P$
- Το σύνολο P περιέχει:
  - ένα μηδενικό στοιχείο ϕ τέτοιο ώστε $\phi \vee x = x$ για όλα τα $x \in P$, και $\phi \cdot x = x \cdot \phi$ για όλα τα $x \in P$
  - ένα στοιχείο μονάδας e έτσι ώστε $e \cdot x = x = x \cdot e$ για όλα τα $x \in P$

Τα ονόματα των πράξεων ∨ και · είναι σύζευξη (join) και πολλαπλασιασμός (multiplication), αντίστοιχα.

2.2 Το πρόβλημα της Σύστασης

Ένα πρόβλημα σύστασης μπορεί να περιγραφεί ως εξής: Υποθέστε ένα σύστημα συστάσεων όπου U είναι το σύνολο των χρηστών και I το σύνολο όλων των πιθανών προτεινόμενων αντικειμένων στο σύστημα. Έστω $f : U \times I \rightarrow R$ μια συνάρτηση χρησιμότητας (utility function) που μετρά τη χρησιμότητα ενός αντικειμένου i σε έναν χρήστη u, όπου το R είναι ένα σύνολο ταξινομημένο. Συνήθως, ο στόχος ενός συστήματος συστάσεων είναι να εκτιμήσει το f με βάση δεδομένα του παρελθόντος και να χρησιμοποιήσει το f για να προβλέψει τη χρησιμότητα του αντικειμένου $i \in I$ σε έναν χρήστη $u \in U$. Η χρησιμότητα συνήθως αντιπροσωπεύεται από αξιολογήσεις, αλλά μπορεί να είναι οποιαδήποτε συνάρτηση. Τα προηγούμενα δεδομένα μπορεί να περιλαμβάνουν ποικιλόμορφα πράγματα, όπως την προηγούμενη συμπεριφορά του χρήστη, τη σχέση του χρήστη με άλλους χρήστες, συναφείς πληροφορίες, ομοιότητα περιεχομένου μεταξύ των αντικειμένων κ.α. Κανονικά, η πληθικότητα του συνόλου αντικειμένων *I* είναι υψηλή και οι χρήστες βαθμολογούν μόνο μερικά αντικείμενα.

2.2.1 Προσεγγίσεις Συστάσεων

Υπάρχει μια ποικιλία διαφορετικών αλγορίθμων συστάσεων, ανάλογα με το τι λαμβάνει υπόψη το σύστημα συστάσεων για να κάνει μια σύσταση. Ανεξάρτητα από τις παραλλαγές, ο κλασικός τρόπος με τον οποίο κατηγοροποιούνται [22] οι συστάσεις είναι ως εξής: (α) βάση περιεχομένου (content-based), (β) συνεργατικό φιλτράρισμα (collaborative filtering), και (γ) υβριδικό (hybrid)[12].

2.2.2 Συστάσεις βάσει περιεχομένου

Στις συστάσεις βάσει περιεχομένου (content-based recommendations) το σύστημα αναλύει τις προηγούμενες επιλογές ενός χρήστη και μαθαίνει να προτείνει αντικείμενα που είναι παρόμοια με αυτά που άρεσαν στον χρήστη στο παρελθόν. Το περιεχόμενο μπορεί να οριστεί ως μεταδεδομένα, περιγραφή, θέματα κ.α. Η ομοιότητα των αντικειμένων υπολογίζεται με βάση τα χαρακτηριστικά που σχετίζονται με τα συγκεκριμένα αντικείμενα. Για παράδειγμα, εάν ένας χρήστης έχει αξιολογήσει θετικά μια ταινία που ανήκει στο είδος κωμωδίας, τότε το σύστημα μπορεί να μάθει να προτείνει άλλες ταινίες από αυτό το είδος. Η βασική διαδικασία που εκτελείται συνίσταται στη δημιουργία ενός προφίλ χρήστη και ενός προφίλ αντικειμένου και, στη συνέχεια, αντιστοιχεί τα χαρακτηριστικά ενός προφίλ χρήστη, στο οποίο αποθηκεύονται οι προτιμήσεις και τα ενδιαφέροντα του, με τα χαρακτηριστικά ενός αντικειμένου περιεχομένου (αντικείμενο), προκειμένου να συσταθούν στον χρήστη νέα ενδιαφέροντα αντικείμενα. Εάν ένα προφίλ αντικατοπτρίζει με ακρίβεια τις προτιμήσεις των χρηστών, είναι τεράστιο πλεονέκτημα για την αποτελεσματικότητα μιας διαδικασίας πρόσβασης σε πληροφορίες. Συνήθως, είναι δυνατό να γίνει διάκριση μεταξύ δύο ειδών σχετικών προτιμήσεων από τους χρήστες: θετικές πληροφορίες (δηλαδή συμπερασματικά χαρακτηριστικά που αρέσουν στον χρήστη) και αρνητικές πληροφορίες (δηλαδή συμπέρασμα χαρακτηριστικών που ο χρήστης δεν ενδιαφέρεται). Μπορούν να υιοθετηθούν δύο διαφορετικές τεχνικές για την καταγραφή των προτιμήσεων των χρηστών. Όταν ένα σύστημα απαιτεί από τον χρήστη να αξιολογήσει ρητά τα αντικείμενα, αυτή η τεχνική αναφέρεται συνήθως ως ρητή ανατροφοδότηση (explicit feedback) · η άλλη τεχνική, που ονομάζεται άρρητη ανατροφοδότηση (implicit feedback), δεν απαιτεί ενεργή συμμετοχή του χρήστη, με την έννοια ότι η ανατροφοδότηση προέρχεται από την παρακολούθηση και την ανάλυση των αλληλεπιδράσεων του χρήστη με το σύστημα. Η ρητή ανατροφοδότηση έχει το πλεονέκτημα της απλότητας, αλλά αυξάνει το φορτίο αλληλεπίδρασης του χρήστη με το σύστημα του και μπορεί να μην είναι επαρκής. Οι μέθοδοι άρρητης ανατροφοδότησης βασίζονται στην εκχώρηση βαθμολογίας συνάφειας σε συγκεκριμένες ενέργειες χρήστη σε ένα αντικείμενο, όπως αποθήκευση, απόρριψη, εκτύπωση, σελιδοδείκτης κ.α.

2.2.3 Συστάσεις για συνεργατικό φιλτράρισμα

Στις συστάσεις φιλτραρίσματος συνεργασίας (Collaborative filtering) το σύστημα αναλύει δεδομένα χρήσης και προτείνει αντικείμενα που άρεσαν στο παρελθόν σε άλλους χρήστες με παρόμοιες προτιμήσεις. Τα δεδομένα χρήσης μπορεί να περιλαμβάνουν αξιολογήσεις, αγορές, λήψεις κ.λπ. Η ομοιότητα στην προτίμηση δύο χρηστών υπολογίζεται με βάση την ομοιότητα στο ιστορικό συμπεριφοράς των χρηστών.

Οι συνεργατικές μέθοδοι φιλτραρίσματος μπορούν να ομαδοποιηθούν στις δύο κατηγορίες (α) γειτνίασης (neighborhood) και (β) με βάση ένα μοντέλο. Σε μοντέλα που βασίζονται στην γειτνίαση, το σύστημα αποθηκεύει τις αλληλεπιδράσεις χρήστη-αντικειμένου (π.χ. αξιολογήσεις, αγορές) και τις χρησιμοποιεί απευθείας για να προβλέψει τις βαθμολογίες για νέα αντικείμενα. Τα μοντέλα γειτνίασης μπορούν περαιτέρω να κατηγοριοποιηθούν ως βασισμένα σε χρήστες και βασισμένα σε αντικείμενα. Οι μέθοδοι που βασίζονται σε χρήστες βρίσκουν χρήστες που έχουν την ίδια προτίμηση για αντικείμενα και προτείνουν νέα αντικείμενα με βάση το τι άρεσε σε αυτούς τους παρόμοιους χρήστες. Οι μέθοδοι που βασίζονται σε αντικείμενα προτείνουν αντικείμενα που είναι παρόμοια με αυτά που αρέσουν στον χρήστη. Σε αυτήν την περίπτωση, ωστόσο, η διαφορά με τις προσεγγίσεις που βασίζονται στο περιεχόμενο είναι ότι η ομοιότητα των αντικειμένων βασίζεται σε αξιολογήσεις αντικειμένων από όλους τους χρήστες στη βάση δεδομένων. Οι προσεγγίσεις που

βασίζονται σε μοντέλα (model-based) δεν χρησιμοποιούν τις αλληλεπιδράσεις άμεσα, αλλά τις χρησιμοποιούν για να μάθουν ένα μοντέλο πρόβλεψης. Η γενική ιδέα είναι να μοντελοποιηθούν οι αλληλεπιδράσεις χρήστη-αντικειμένου με παράγοντες (factors) που αντιπροσωπεύουν λανθάνων (latent) χαρακτηριστικά των χρηστών και των αντικειμένων, συμπιέζοντάς τα έτσι αποτελεσματικά σε μια αναπαράσταση χαμηλής διάστασης. Αυτά τα μοντέλα αρχικά εκπαιδεύονται χρησιμοποιώντας τα διαθέσιμα δεδομένα και αργότερα χρησιμοποιούνται για την πρόβλεψη των προτιμήσεων των χρηστών για νέα αντικείμενα. Οι μέθοδοι αυτής της κατηγορίας περιλαμβάνουν μεθόδους παραγοντοποίησης πινάκων, Μπαεζιανά δίκτυα, ομαδοποίηση κ.λπ.

Οι συνεργατικές προσεγγίσεις φιλτραρίσματος είναι απλές, αποτελεσματικές και μπορούν να δημιουργήσουν ακριβείς συστάσεις καλής ποιότητας. Απαιτούν ελάχιστες γνώσεις τομέα και καμία γνώση των χαρακτηριστικών χρήστη/αντικειμένου. Ξεπερνούν ορισμένους από τους περιορισμούς των προσεγγίσεων που βασίζονται στο περιεχόμενο. Ωστόσο, παρόλο που είναι εξαιρετικά δημοφιλή, υποφέρουν από τα ακόλουθα προβλήματα:

• **Ψυχρή εκκίνηση (cold start)**[23]**:** Για έναν νέο χρήστη ή αντικείμενο δεν υπάρχουν αρκετά δεδομένα για να γίνουν ακριβείς συστάσεις

• **Αραιότητα δεδομένων (data sparsity):** Ο αριθμός των διαθέσιμων αντικειμένων είναι εξαιρετικά μεγάλος, ωστόσο οι περισσότεροι χρήστες θα έχουν αλληλοεπιδράσει μόνο με ένα μικρό μέρος της συνολικής βάσης αντικειμένων. Έτσι, ακόμη και τα πιο δημοφιλή αντικείμενα έχουν πολύ λίγες βαθμολογίες. Όταν τα δεδομένα αλληλεπίδρασης είναι αραιά, δύο χρήστες ή αντικείμενα είναι απίθανο να έχουν κοινές αξιολογήσεις και, κατά συνέπεια, οι προσεγγίσεις που βασίζονται στην γειτνίαση θα προβλέπουν αξιολογήσεις χρησιμοποιώντας έναν πολύ περιορισμένο αριθμό γειτόνων.

• **Επεκτασιμότητα (scalability):** Σε πολλές από τις εφαρμογές στις οποίες αυτά τα συστήματα κάνουν συστάσεις, υπάρχουν εκατομμύρια χρήστες και αντικείμενα και καθώς απαιτούν υπολογισμό ομοιότητας μεταξύ χρηστών ή αντικειμένων, η ποσότητα της υπολογιστικής ισχύος αυξάνεται πολύ με αυτούς τους αριθμούς.

2.2.4 Υβριδικές συστάσεις

Τα υβριδικά συστήματα συστάσεων βασίζονται στον συνδυασμό διαφορετικών τεχνικών συστάσεων. Ο συνδυασμός στοχεύει στην επίτευξη καλύτερης απόδοσης χρησιμοποιώντας τα πλεονεκτήματα μιας μεθόδου για να διορθωθούν τα μειονεκτήματα μιας άλλης. Για παράδειγμα, ένας συνδυασμός συλλογικού φιλτραρίσματος που αντιμετωπίζει προβλήματα ψυχρής εκκίνησης, π.χ. νέα στοιχεία δεν μπορούν να προταθούν επειδή δεν έχουν ακόμη βαθμολογίες, μπορεί να επιλυθεί χρησιμοποιώντας από κοινού μια προσέγγιση βασισμένη στο περιεχόμενο, καθώς η πρόβλεψη για νέα στοιχεία βασίζεται στα χαρακτηριστικά τους που είναι συνήθως διαθέσιμα. Υπάρχουν διάφοροι τρόποι με τους οποίους μπορεί κανείς να συνδυάσει τεχνικές συστάσεων σε ένα νέο υβριδικό σύστημα, π.χ. σταθμισμένο (weighted) (συνδυασμός βαθμολογιών διαφορετικών συστατικών συστάσεων), εναλλαγή (switching) (διαθέτοντας διάφορα στοιχεία συστάσεων και επιλογή ενός για τη δημιουργία βαθμολογίας), μικτό (παρουσίαση διαφορετική σύσταση μαζί), συνδυασμός (mixed) χαρακτηριστικών (συνδυασμός χαρακτηριστικών που προέρχονται από διαφορετικές πηγές γνώσης). Μια

πιο πλήρης λίστα μπορεί να βρεθεί στο [12]. Το Netflix είναι ένα σύγχρονο παράδειγμα επιτυχημένες υβριδικής σύστασης, ωστόσο, ο σχεδιασμός μίας καλής υβριδικής σύστασης δεν είναι εύκολη υπόθεση, καθώς πρέπει να αποφασίσει κανείς για μια κατάλληλη παραλλαγή, ενώ επίσης λαμβάνει υπόψη ότι ορισμένες παραλλαγές είναι ευαίσθητες στα δυνατά σημεία και τις αδυναμίες του τεχνικές σύνθεσης.

2.3 Βάσεις δεδομένων γραφημάτων

Μια βάση δεδομένων γραφήματος (Graph Databases) είναι ένα σύστημα διαχείρισης βάσης δεδομένων που λειτουργεί σε ένα μοντέλο δεδομένων γραφήματος και υποστηρίζει λειτουργίες δημιουργίας, ανάγνωσης, ενημέρωσης και διαγραφής στους κόμβους και τις ακμές αυτού του γραφήματος [3]. Το μοντέλο δεδομένων είναι απλούστερο και πιο εκφραστικό από τα μοντέλα σχεσιακών ή άλλων βάσεων δεδομένων NoSQL. Σε αντίθεση με αυτές τις άλλες βάσεις δεδομένων, η πρώτη προτεραιότητα στις βάσεις δεδομένων γραφημάτων είναι οι σχέσεις μεταξύ των οντοτήτων, δηλαδή οι ακμές. Οι βάσεις δεδομένων γραφημάτων έχουν κατασκευαστεί για χρήση με συστήματα συναλλαγών.

Τα δύο κύρια συστατικά των τεχνολογιών βάσεων δεδομένων γραφημάτων είναι η αποθήκευση γραφημάτων και η μηχανή επεξεργασίας γραφημάτων. Η πλειοψηφία των βάσεων δεδομένων γραφημάτων χρησιμοποιεί εγγενή αποθήκευση γραφημάτων, πράγμα που σημαίνει ότι έχουν σχεδιαστεί ειδικά για την αποθήκευση και τη διαχείριση γραφημάτων. Ωστόσο, υπάρχουν άλλες που χρησιμοποιούν σχεσιακές ή αντικειμενοστραφής βάσεις δεδομένων. Αυτά με εγγενή αποθήκευση γραφημάτων είναι πιο γρήγορα. Επιπλέον, πολλές βάσεις δεδομένων γραφημάτων μπορούν να υποστηρίξουν εγγενή επεξεργασία γραφημάτων (δηλαδή γειτνίαση χωρίς ευρετήριο) που είναι ο πιο αποτελεσματικός τρόπος επεξεργασίας δεδομένων γραφήματος, καθώς οι συνδεδεμένοι κόμβοι δείχνουν φυσικά ο ένας στον άλλο στη βάση δεδομένων.

Οι βάσεις δεδομένων γραφημάτων έχουν επιπλέον τα πλεονεκτήματα της απόδοσης και της ευελιξίας. Η απόδοση στις παραδοσιακές βάσεις δεδομένων επιδεινώνεται εξαιρετικά κάθε φορά που εμφανίζονται ερωτήματα σχέσεων όπου ο αριθμός και το βάθος των σχέσεων αυξάνεται. Αντίθετα, η απόδοση της βάσης δεδομένων γραφημάτων παραμένει σταθερή ακόμη και όταν τα δεδομένα μεγαλώνουν σε όγκο με την πάροδο του χρόνου. Η ευελιξία στις βάσεις δεδομένων γραφημάτων προέρχεται από το υποστηριζόμενο μοντέλο γραφήματος. Οι ανάγκες της εφαρμογής ενδέχεται να αλλάξουν γρήγορα, αλλά δεν χρειάζεται να μοντελοποιηθεί εξαντλητικά ένας τομέας εκ των προτέρων. Ένα μοντέλο γραφήματος είναι ευέλικτο σε αλλαγές στη δομή και το σχήμα του, καθώς μπορεί κανείς να προσθέσει/αφαιρέσει οντότητες στο υπάρχον γράφημα χωρίς να επηρεάσει τη λειτουργικότητα.

Τα πλεονεκτήματα που προσφέρουν οι βάσεις δεδομένων γραφημάτων στα υποστηριζόμενα δεδομένα και λειτουργίες, τις καθιστούν ιδανικές για να επιβληθούν στον χώρο συστάσεων, καθώς οι εφαρμογές στον τομέα συστάσεων πρέπει επίσης να αντιμετωπίσουν εντατικό χειρισμό σχέσεων δεδομένων που υπόκεινται σε χρονικές αλλαγές και που αυξάνονται καθημερινά κατά αρκετές τάξεις μεγέθους.

3. ΜΟΝΤΕΛΟ RECGRAPH

3.1 Μοντέλο δεδομένων RecGraph

Το μοντέλο δεδομένων RecGraph [6][7] ορίζεται ως μία εξάδα RecGraph(V, E, L_V, L_E, A_V, A_E) που καταγράφει ένα κατευθυνόμενο γράφημα ως εξής:

- V είναι το σύνολο των κόμβων και *E* είναι το σύνολο των ακμών. Οποιαδήποτε οντότητα επηρεάζει άμεσα ή έμμεσα τη σύσταση, συμπεριλαμβανομένης οποιασδήποτε οντότητα σύστασης, είναι ένας κόμβος, και οποιαδήποτε σύνδεση μεταξύ τους αποτελεί μία ακμή.
- Τα L_V και L_E είναι τα σύνολα ετικετών για κόμβους και ακμές και A_V και A_E τα σύνολα χαρακτηριστικών για κόμβους και ακμές, αντίστοιχα. Κάθε ετικέτα κόμβος ή ακμής *l* σχετίζεται με ένα σύνολο χαρακτηριστικών A_V^l = $\{a_v | a_v \in A_V\}$ ή A_E^l = $\{a_e | a_e \in A_E\}$, αντίστοιχα
- Οι κόμβοι και οι ακμές αντιπροσωπεύουν είτε συγκεκριμένα αντικείμενα (στιγμιότυπα), π.χ. μια συγκεκριμένη κριτική χρήστη, είτε κατηγορίες αντικειμένων (το σχήμα), π.χ. την κλάση όλων των κριτικών από τους χρήστες.

Ένας κόμβος v

- έχει μια ετικέτα $l \in L_V$ που δηλώνει τον τύπο του και προσδιορίζεται από ένα μοναδικό αναγνωριστικό,
- έχει μια τιμή *w* για κάθε χαρακτηριστικό $a_v \in A_V^l$, η οποία απεικονίζεται ως ζεύγος $\langle a_v : w \rangle$ και αντλείται από τον πεδίο ορισμού του χαρακτηριστικού του *dom*(a_v),
- συμβολίζεται ως *l(id,* $\{a_i : w_i | 1 \leq i \leq k\}$*)* ή *l(id,* $a_i : w_i, \ldots, a_k : w_k$*)* ή απλά *l(id)* ανάλογα με το περιεχόμενο.

Για παράδειγμα, ένα χαρακτηριστικό βαθμολόγησης (rating) μπορεί να έχει έναν τομέα *dom(rating) = {1, 2, 3, 4, 5}*, ενώ ένας κόμβος αξιολόγησης (review) με το 17 ως το αναγνωριστικό του και δύο χαρακτηριστικά μπορεί να γραφτεί ως review (17, rating: 3 , review : κάποιο κείμενο) ή review(17), εάν τα χαρακτηριστικά του παραλείπονται.

Μια κατευθυνόμενη ακμή e από το u στο v (οι κόμβοι έναρξης και τέλους, αντίστοιχα),

- έχει μια ετικέτα (label) $l \in L_E$ που δηλώνει τον τύπο του και προσδιορίζεται από ένα μοναδικό αναγνωριστικό,

- έχει μια τιμή *w* για κάθε χαρακτηριστικό $a_e \in A_E^I$, η οποία απεικονίζεται ως ζεύγος $\langle a_e : w \rangle$ και αντλείται από το πεδίο ορισμού του *dom(*a_e*)*,
- συμβολίζεται ως *l(id, u, v, {*$a_i : w_i$*|1 ≤ i ≤ k})* ή *l(id, u, v,* $a_i : w_i, \ldots, a_k : w_k$*)* ή, σε απλούστερες μορφές *, l(id, u, v), l(u, v)* ή *l(id),* ανάλογα με το πλαίσιο,
- είναι ζευγοποιημένο (paired up) με μια ισοδύναμη αντίστροφη ακμή, που συμβολίζεται με ένα − πριν από τη δική του ετικέτα, π.χ., *−l(id, v, u)*.

Για παράδειγμα, μία ακμή *αρέσει* από *χρήστη*(5) σε *επιχείρηση*(13) με δύο χαρακτηριστικά μπορεί να γραφτεί ως *αρέσει*(*χρήστης*(5), *επιχείρηση*(13), *βαθμολογία*: 0,9, *αβεβαιότητα*: 0,3), μια *αξιολογημένη* ακμή από *χρήστη*(5) στην *επιχείρηση*(13) με τα χαρακτηριστικά της να παραλείπονται ως *αξιολογημένο*(*χρήστης*(5), *επιχείρηση*(13)), ενώ η αντίστροφη ακμή(*αξιολογημένο από*, εάν είχε ανεξάρτητα ετικέτα) ως -*αξιολογημένο*(*επιχείρηση*(13), χρήστη(5)).

3.2 Υπολογιστικό μοντέλο RecGraph

Μια σύσταση στο γράφημα εκφράζεται ως αποτέλεσμα υπολογισμών μονοπατιού (path) που χρησιμοποιούν τελεστές για συμπέρασμα ακμών. Συγκεκριμένα, ορίζεται μια άλγεβρα μονοπατιού με ένα πολύ μικρό σύνολο γενικών τελεστών μονοπατιού, δηλαδή CON, AGG και FUSE (Σχήμα 1, 2, 3). Αυτά είναι εμπνευσμένα από προηγούμενες άλγεβρες μονοπατιών [17] αλλά επεκτείνονται για να καλύψουν μια ευρύτερη κατηγορία προβλημάτων που βασίζονται σε μονοπάτια, συμπεριλαμβανομένων συστάσεων.

Η φύση τους ακολουθεί ένα παράλληλο παράδειγμα σειράς, όπου το CON συνθέτει διαδοχικές ακμές κατά μήκος ενός μονοπατιού και το AGG και το FUSE συνδυάζουν ακμές μεταξύ των ίδιων κόμβων. Αυτοί οι γενικοί τελεστές μονοπατιού εγκαθίστανται σε συγκεκριμένες συναρτήσεις, οι οποίες λειτουργούν σε χαρακτηριστικά κόμβου και ακμής στην πορεία, ανάλογα με τη στρατηγική σύστασης και τον υπό εξέταση τομέα.

Πιο επίσημα, ορίζονται οι γενικοί τελεστές παρακάτω:

Συνένωση ακμών - Edge Concatenation (CON):

$$CON_f(l_1(s,u_1,\{a_{j_1} : w_{j_1}\}), l_2(u_1,u_2,\{a_{j_2} : w_{j_2}\}), \ldots, l_n(u_n,t,\{a_{j_{n+1}} : w_{j_{n+1}}\})) = l(s,t,\{a_j : w_j\}):$$

Λειτουργεί σε n διαδοχικές ακμές, η πρώτη αρχίζει από *s* και η τελευταία τελειώνει στο *t,* και παράγει μια ακμή από *s* έως *t*. Χρησιμοποιεί τις ετικέτες και τα χαρακτηριστικά των ακμών εισόδου, εφαρμόζοντας μια συνάρτηση f που υποδηλώνει το CON, όπου:

$$CON_f : L_E \times L_E \times \ldots \times L_E \rightarrow L_E, 2^{A_E} \times 2^{A_E} \ldots \times 2^{A_E} \rightarrow 2^{A_E}.$$

Συνάθροιση μονοπατιού - Path Aggregation (AGG):

$$AGG_f(\{l(s,t,\{a_j : w_j\})\}) = l(s,t,\{a_j : w_j\}):$$

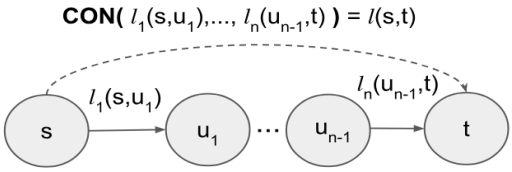
Λειτουργεί σε ένα σύνολο ακμών της ίδιας ετικέτας $l \in L_E$ μεταξύ των κόμβων s και t και παράγει μια ακμή της ίδιας ετικέτας. Χρησιμοποιεί τα χαρακτηριστικά και τις ετικέτες των ακμών εισόδου, εφαρμόζοντας μια συνάρτηση f που δημιουργεί στιγμιότυπο AGG, όπου

$AGG_f : 2^{L_E} \rightarrow L_E, 2^{A_E} \rightarrow A_E.$

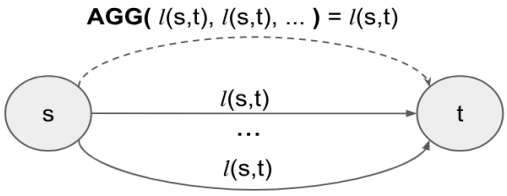
Συγχώνευση μονοπατιού - Path Fusion (FUSE):

$FUSE_f(l_1(s,t,\{a_{j_1} : w_{j_1}\}), l_2(s,t,\{a_{j_2} : w_{j_2}\}), \ldots, l_n(s,t,\{a_{j_n} : w_{j_n}\})) = l(s,t,\{a_j : w_j\}):$

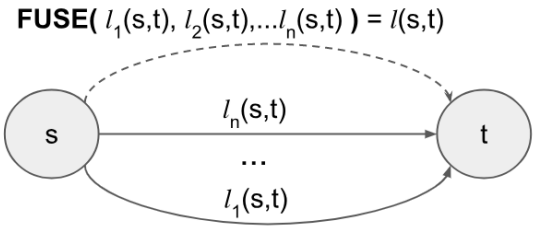
Λειτουργεί σε μια σειρά από παράλληλες ακμές από τον κόμβο s στον κόμβο t με διαφορετικές ετικέτες $l_i \in L_E$ και παράγει μια νέα ακμή από το s στο t. Όπως και με το CON, χρησιμοποιεί τις ετικέτες και τα χαρακτηριστικά των ακμών εισόδου, εφαρμόζοντας μια συνάρτηση f που δημιουργεί στιγμιότυπο FUSE, όπου $FUSE_f: LE \times LE \times \ldots \times LE \rightarrow LE, 2^{A_E} \times 2^{A_E} \ldots \times 2^{A_E} \rightarrow 2^{A_E}$.



Σχήμα 1: τελεστής CON



Σχήμα 2: τελεστής AGG



Σχήμα 3: τελεστής FUSE

3.3 Οι Συστάσεις ως πρόβλημα μονοπατιού

Κάθε φορά που θέλουμε να κάνουμε μια σύσταση, προσπαθούμε να συμπεράνουμε μια άκρη μεταξύ του υποκείμενου (subject) κόμβου (δηλαδή του κόμβου που λαμβάνει συστάσεις) και του αντικείμενου (object) κόμβου (δηλαδή του αντικειμένου που μπορεί να προταθεί). Διαισθητικά, ορίζουμε όλα τα προσβάσιμα μονοπάτια μεταξύ υποκειμένου και αντικειμένου και τα συνδυάζουμε. Πιο επίσημα:

Λαμβάνοντας υπόψη δύο κόμβους *u, v* στο γράφημα *G*, μια σύσταση του *v* στο *u* εξαρτάται από (α) την εξαγωγή μιας άρρητης ακμής από το *u* στο *v* μέσω κάποιου συνδυασμού των μονοπατιών που τους συνδέουν και (β) εκτίμηση των τιμών των ακμών με ετικέτα που προβλέπονται (predicted).

$$PREDICTED\,(u,v) \;=\; (AGG_f|FUSE_f)\{path_{\,u,v} = \; CON_f\}_i$$

Όπου οι τελεστές CON_f διασχίζουν μονοπάτια από το u έως το v και τις συνθέτουν σε ακμές, ενώ οι τελεστές AGG_f και $FUSE_f$ συναθροίζουν τις προκύπτουσες ομοιογενείς ή ετερογενείς ακμές, αντίστοιχα, για να εξαγάγουν την τελική προβλεπόμενη ακμή. Κάθε τελεστής μπορεί να δημιουργηθεί με τη δική του λειτουργία f.

3.4 Ιδιότητες τελεστών Recgraph

Όπως και σε άλλες άλγεβρες μονοπατιών, οι τελεστές μπορεί να παρουσιάζουν ή μπορεί να μη παρουσιάζουν ορισμένες ιδιότητες. Παρακάτω παρουσιάζεται το ιδανικό σύνολο ιδιοτήτων για την άλγεβρα μονοπατιών RecGraph. Το εάν κάθε ιδιότητα ισχύει ή όχι, εξαρτάται από τις ιδιαίτερες συναρτήσεις που δημιουργούν τους γενικούς τελεστές που χρησιμοποιούνται στη σχετική έκφραση μονοπατιού.

- Το CON_f είναι προσαιτεριστικό.

$$\forall s,u_1,u_2,t:\; CON_{f_{12-3}}(CON_{f_{12}}(l_1(s,u_1),l_2(u_1,u_2)),l_3(u_2,t)) \\ = \; CON_{f_{1-23}}(l_1(s,u_1),CON_{f_{23}}(l_2(u_1,u_2),l_3(u_2,t)))$$

- Το AGG_f είναι προσαιτεριστικό.

$$\forall s,t.:\; AGG_f|(\{\,l(i_1,s,t),l(i_2,s,t)),l(i_3,s,t)\,\}) \\ = \; AGG_f|(\{\,AGG_f|(\,\{l(i_1,s,t),l(i_2,s,t)\,\}),l(i_3,s,t)\,\}) \\ = \; AGG_f|(\,\{l(i_1,s,t),AGG_f|(\,\{l(i_2,s,t),l(i_3,s,t)\}\,)\}\,)$$

- Το $FUSE_f$ είναι προσαιτεριστικό.

$$\forall s,t.:\; FUSE_f(\,FUSE_f(l_1(s,t),l_2(s,t)),l_3(s,t)\,) \;=\; FUSE_f(\,l_1(s,t),FUSE_f(\,l_2(s,t),l_3(s,t))\,)$$

- Το $FUSE_f$ είναι αντιμεταθετικό.

$$\forall s,t.:\; FUSE_f(l_1(s,t),l_2(s,t)\,) \;=\; FUSE_f(l_2(s,t),l_1(s,t)\,)$$

- Το CON_f είναι επιμεριστικό στο $AGG_{f'}$.

$$\forall s,u,t.:CON_{f_1}\left(l_1(s,u),AGG_{f_2}\left(l^{(2)}(u,t),l^{(3)}(u,t)\right)\right) \\ = AGG_{f_2}\left(CON_{f_1}\left(l_1(s,u),l^{(2)}(u,t)\right),CON_{f_1}\left(l_1(s,u),l^{(3)}(u,t)\right)\right)$$

$$CON_{f_1}\left(AGG_{f_2}\left(l^{(2)}(u,t), l^{(3)}(u,t)\right), l_1(s,u)\right)$$

$$= AGG_{f_2}\left(CON_{f_1}\left(l^{(2)}(u,t), l_1(s,u)\right), CON_{f_1}\left(l^{(3)}(u,t), l_1(s,u)\right)\right),$$

$$where l^{(2)} = l^{(3)} = l$$

- Το CON_f είναι επιμεριστικό στο $FUSE_{f,}$.

$$\forall s,u,t.:\ CON_{f_1}(l_1(s,u), FUSE_{f_2}(l_2(u,t), l_3(u,t)))$$

$$= FUSE_{f_2}(CON_{f_1}(l_1(s,u), l_2(u,t)), CON_{f_1}(l_1(s,u), l_3(u,t)))$$

$$CON_{f_1}(FUSE_{f_2}(l_2(u,t), l_3(u,t)), l_1(s,u))$$

$$= FUSE_{f_2}(CON_{f_1}(l_2(u,t), l_1(s,u)), CON_{f_1}(l_3(u,t), l_1(s,u)))$$

Ανάλογα με την παρουσία ή την απουσία αυτών των ιδιοτήτων στους τελεστές της άλγεβρας, τα προβλήματα που εκφράζονται σε αυτές τις άλγεβρες μπορεί να είναι ή να μην είναι καλά διαμορφωμένα, αποσυνθέσιμα. Γενικά, όσο περισσότερες είναι οι ιδιότητες που διατηρούν, τόσο περισσότερες είναι οι ευκαιρίες βελτιστοποίησης [13].

3.5 Εκτεταμένοι κανόνες RecGraph

Ένας αλγόριθμος στο πλαίσιο RecGraph δημιουργείται χρησιμοποιώντας τους παραπάνω τελεστές και μια μηχανή γραφημάτων τον μεταφράζει σε πλάνο εκτέλεσης. Ο βελτιστοποιητής (optimizer) της μηχανής δημιουργεί εναλλακτικά υποψήφια πλάνα και εκτελεί το βέλτιστο, με βάση τις ιδιότητες του τελεστή και τα στατιστικά στοιχεία της βάσης δεδομένων.

- **Κανόνας 1.** Ένας n-ary τελεστής CON θα μπορούσε να αποσυντεθεί σε διαδοχικούς τελεστές CON μικρότερης πληθικότητα, π.χ. το πολύ n − 1 δυαδικούς τελεστές CON και αντίστροφα. Οι τελεστές που δημιουργήθηκαν πρόσφατα υλοποιούν τις δικές τους λειτουργίες που συλλογικά καταλήγουν στο ίδιο αποτέλεσμα που θα παρήγαγε η συνάρτηση του n-ary CON, ανάλογα με τη σημασιολογία των ετικετών στις οποίες εφαρμόζεται ο τελεστής.

$$CON_f(l_1(s,u_1),\ldots,l_n(u_{n-1},t))$$

$$= CON_{f'}(CON_{f_1}((l_1(s,u_1), l_2(u_1,u_2)),\ldots,CON_{f_n}(l_{n-1}(u_{n-2},u_{n-1}), l_n(u_{n-1},t))).$$

- **Κανόνας 2.** Ένας τελεστής AGG/FUSE που εφαρμόζεται σε ένα n-ary CON θα μπορούσε να αποσυντεθεί σε πολλούς τελεστές AGG/FUSE σε κάθε δημιουργημένο CON και αντίστροφα. Οι νέοι τελεστές εφαρμόζουν συλλογικά την ίδια λειτουργία που παρήγαγαν οι αρχικοί τελεστές.

$$AGG_f(\{CON_{f'}(l_1(s,u_1),\ldots,l_n(u_{n-1},t))\}) = CON_{f''}(AGG_{f_1}(\{CON_{f'1}((l_1(s,u_1),l_2(u_1,u_2))\}),\ldots, AGG_{f_n}(\{CON_{f'n}(l_{n-1}(u_{n-2},u_{n-1}),l_n(u_{n-1},t))\}))$$

Διαισθητικά, όσο πιο ομοιόμορφο είναι το γράφημα (δηλαδή η ίδια σημασιολογία) και όσο περισσότερες ιδιότητες έχουν οι συναρτήσεις (π.χ. οι τυπικοί τελεστές πρόσθεσης και πολλαπλασιασμού εμφανίζουν όλες τις παραπάνω αλγεβρικές ιδιότητες), τόσο το καλύτερο για ευκαιρίες βελτιστοποίησης. Στο RecGraph, η εφαρμογή τελεστών AGG και FUSE όσο το δυνατόν νωρίτερα είναι μια καλή στρατηγική, καθώς μειώνει τον αριθμό των μονοπατιών που δημιουργούνται. Ωστόσο, μπορεί να είναι πιο λογικό να αναβάλλουμε τις συναθροίσεις όταν η υπερβάθμια των κόμβων δεν είναι πολύ μεγάλη· οι μηχανές γραφημάτων μπορούν να εκτελούν διασχίσεις μονοπατιών πολύ γρήγορα, επομένως εάν τα μονοπάτια είναι μεγάλα αλλά λίγα, είναι πιο γρήγορο να διασχίσετε πρώτα χρησιμοποιώντας τελεστές CON και συγκεντρώστε αργότερα με AGG/FUSE.

3.6 Γλώσσα ερωτήματος RecGraph

Συσχετίζεται με την άλγεβρα μονοπατιών και είναι μια δηλωτική γλώσσα ερωτημάτων τύπου Datalog που χρησιμοποιεί αυτούς τους τελεστές για να γράψει ομοιόμορφα αλγόριθμους συστάσεων, ως παράγωγα βάσει μονοπατιών νέων ακμών που σχετίζονται με συστάσεις. Κατ' αρχήν, οι αλγόριθμοι που εκφράζονται σε αυτή τη γλώσσα ως κανόνες που το σύστημα μετατρέπει σε αλγεβρικές εκφράσεις (όπως στα συμβατικά συστήματα βάσεων δεδομένων) στην άλγεβρα μονοπατιού του RecGraph. Η απόδοση μπορεί στη συνέχεια να βελτιστοποιηθεί με βάση τις ιδιότητες τελεστή μονοπατιού που προσδιορίζονται παραπάνω. Η σύνταξη της γλώσσας επεξηγείται παρακάτω και δείχνει τους κανόνες που καταγράφουν τις γενικές εφαρμογές των τριών τελεστών (CON, AGG, FUSE) στη δυαδική τους μορφή:

```
% CON rule.

label(node1,noden;⟨attributej⟩) :=

label1(node1,node2;⟨attributei1⟩),...,labeln − 1(node − 1,noden;⟨attributein − 1⟩);

attributej = functionCON(⟨attributei1⟩,...,⟨attributein − 1⟩).

% AGG rule.

label(node1,node2;⟨attributek⟩) :=

[label(node1,node2;⟨attributeki⟩);attributek = functionAGG(⟨attributeki⟩)].

% FUSE rule.

label(node1,node2;⟨attributej⟩) :=

[label1(node1,node2;⟨attributei1⟩),...,labeln(node1,node2;⟨attributein − 1⟩);

attributek = functionFUSE(⟨attributei1⟩,...,⟨attributein − 1⟩)].
```

4. Ο ΑΛΓΟΡΙΘΜΟΣ FUNK-SVD ΚΑΙ Η ΓΡΑΦΙΚΗ ΤΟΥ ΜΟΝΤΕΛΟΠΟΙΗΣΗ

Σε αυτήν την ενότητα θα επικεντρωθούμε στον αλγόριθμο συστάσεων Funk-SVD και την αναπαράσταση των δομών του συγκεκριμένου αλγορίθμου με τη χρήση ενός κατευθυνόμενου γράφου με χαρακτηριζόμενες (labeled) ακμές. Ο αλγόριθμος Funk-SVD μας επιτρέπει να προβλέψουμε μία βαθμολογία για κάθε ζεύγος χρήστη-αντικειμένου χρησιμοποιώντας τεχνικές μηχανικής μάθησης. Για να επιλύσει το πρόβλημα σύστασης χρησιμοποιεί και εκπαιδεύει ένα κρυμμένο επίπεδο νευρώνων που περιγράφουν τα λανθάνοντα (κρυμμένα) χαρακτηριστικά των αντικειμένων του. Στο Κεφάλαιο 4.1, παρουσιάζουμε την βασική μορφή του SVD αλγορίθμου η οποία χρησιμοποιεί πίνακες για να περιγράψει τις βαθμολογίες και συστάσεις χρηστών για αντικείμενα. Στο Κεφάλαιο 4.2 παρουσιάζουμε τον αλγόριθμο Funk-SVD και την χρήση γράφου αντί πινάκων για την αναπαράσταση των βασικών δομών του Funk-SVD και των ισοδύναμων πράξεων που θα εκτελεστούν επάνω στον αντίστοιχο γράφο για να εκτελεστεί ο αλγόριθμος. Η συγκεκριμένη αναπαράσταση μας επιτρέπει την αναγωγή του προβλήματός μας στην γλώσσα RecGraph και τους τελεστές της η οποία θα παρουσιαστεί στο Κεφάλαιο 5.

4.1 Αποσύνθεση Ιδιαζουσών Τιμών (SVD)

Η αποσύνθεση ιδιαζουσών τιμών (Singular Value Decomposition ή SVD)[24][25], μια κλασική μέθοδος από τη γραμμική άλγεβρα έχει γίνει δημοφιλής στον τομέα της επιστήμης δεδομένων και της μηχανικής μάθησης. Αυτή η δημοτικότητα οφείλεται στην εφαρμογή του στην ανάπτυξη συστημάτων συστάσεων. Υπάρχουν πολλές διαδικτυακές εφαρμογές με επίκεντρο τον χρήστη, όπως προγράμματα αναπαραγωγής βίντεο, προγράμματα αναπαραγωγής μουσικής, εφαρμογές ηλεκτρονικού εμπορίου, κ.λπ., όπου προτείνεται στους χρήστες περισσότερα στοιχεία για αλληλεπίδραση.

Η εύρεση και η σύσταση πολλών κατάλληλων αντικειμένων που θα αρέσουν και θα επιλεγούν από τους χρήστες είναι πάντα μια πρόκληση. Υπάρχουν πολλές τεχνικές που χρησιμοποιούνται στα συστήματα συστάσεων και η SVD είναι μία από αυτές τις τεχνικές.

Όπως έχει ειπωθεί και πιο πάνω, ένα σύστημα συστάσεων είναι ένα έξυπνο σύστημα που προβλέπει τη βαθμολογία και τις προτιμήσεις των χρηστών σε προϊόντα. Η κύρια εφαρμογή των συστημάτων συστάσεων είναι η εύρεση μιας σχέσης μεταξύ χρήστη και προϊόντων προκειμένου να μεγιστοποιηθεί η δέσμευση χρήστη-προϊόντος. Η κύρια εφαρμογή των συστημάτων συστάσεων είναι η πρόταση σχετικού βίντεο ή μουσικής για τη δημιουργία μιας λίστας αναπαραγωγής για τον χρήστη όταν ασχολείται με ένα σχετικό αντικείμενο.

Η SVD είναι μια τεχνική παραγοντοποίησης πινάκων, η οποία μειώνει τον αριθμό των χαρακτηριστικών ενός συνόλου δεδομένων μειώνοντας τη διάσταση του χώρου από N-διάσταση σε K-διάσταση (όπου K<N). Στο πλαίσιο του συστήματος συστάσεων, η SVD χρησιμοποιείται ως τεχνική συνεργατικού φιλτραρίσματος. Χρησιμοποιεί μια δομή πίνακα όπου κάθε σειρά αντιπροσωπεύει έναν χρήστη και κάθε στήλη αντιπροσωπεύει ένα αντικείμενο. Τα στοιχεία αυτού του πίνακα είναι οι βαθμολογίες που δίνονται σε αντικείμενα από τους χρήστες.

Η παραγοντοποίηση αυτού του πίνακα γίνεται με την αποσύνθεση της ιδιάζουσας τιμής (SVD). Όπου βρίσκει συντελεστές πινάκων από την παραγοντοποίηση ενός πίνακα

υψηλού επιπέδου (χρήστη-αντικείμενο-βαθμολόγηση). Η μέθοδος αποσύνθεσης ενός πίνακα σε τρεις άλλους πίνακες δίνονται παρακάτω:

$$A = USV^T$$

Πίνακας U: μοναδικός πίνακας (χρήστης* λανθάνοντες παράγοντες)

Πίνακας S: διαγώνιος πίνακας (δείχνει την ισχύ κάθε λανθάνοντος παράγοντα)

Πίνακας V: μοναδικός πίνακας (αντικείμενο*λανθάνοντες παράγοντες)

Από την παραγοντοποίηση πινάκων, οι λανθάνοντες παράγοντες δείχνουν τα χαρακτηριστικά των αντικειμένων. Ο πίνακας χρησιμότητας A παράγεται με σχήμα m*n, m χρήστες, n αντικείμενα. Η τελική έξοδος του πίνακα A μειώνει τη διάσταση μέσω της εξαγωγής λανθάνοντων παραγόντων. Ο πίνακα A, δείχνει τις σχέσεις μεταξύ χρηστών και αντικειμένων αντιστοιχίζοντας τον χρήστη και το αντικείμενο σε λανθάνοντα χώρο r-διάστασης. Το διάνυσμα x_i θεωρείται κάθε αντικείμενο και το διάνυσμα y_u θεωρείται ως κάθε χρήστης. Η βαθμολογία δίνεται από έναν χρήστη σε ένα αντικείμενο $r_{ui} = x_i^T \cdot y_u$. Το σφάλμα (error) μπορεί να ελαχιστοποιηθεί από την τετραγωνική διαφορά σφάλματος μεταξύ του γινομένου του r_{ui} και της αναμενόμενης βαθμολογίας.

$$Min(x,y) \sum_{(u,i)\in K} (r_{ui} - x_i^T \cdot y_u)^2$$

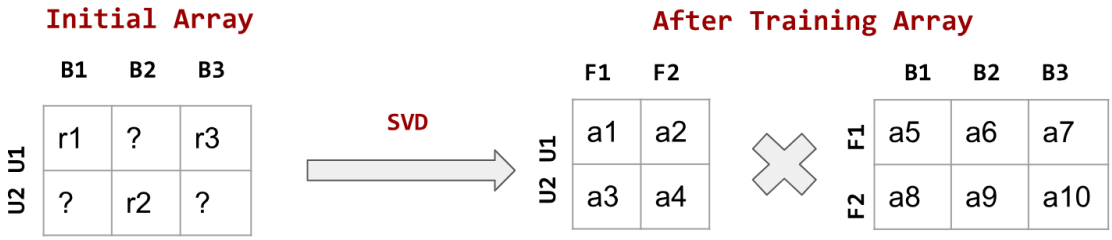
4.2 Η μέθοδος Funk-SVD στο Μοντέλο RecGraph

Η παραγοντοποίηση πινάκων οδηγεί σε ένα γινόμενο πινάκων και περιλαμβάνει μια ποικιλία αποσυνθέσεων πινάκων. Στα συστήματα συστάσεων, αντιστοιχεί σε μια κατηγορία συνεργατικών αλγορίθμων φιλτραρίσματος που αποσυνθέτουν το πίνακα αντικειμένου-χρήστη στο γινόμενο δύο πινάκων χαμηλότερης διαστάσεων. Οι αλληλεπιδράσεις μεταξύ χρηστών και αντικειμένων, με τις αξιολογήσεις να είναι οι πιο κλασικές, αποτυπώνονται ως ακμές στο μοντέλο μας. Οι αντιπροσωπευτικές μέθοδοι που μπορούν έτσι να εκφραστούν περιλαμβάνουν τις [14], [15], [16], [17], [18], [19]. Για να δείξουμε τη μοντελοποίηση σε αυτήν την κατηγορία, επιλέγουμε έναν αλγόριθμο που διαδόθηκε από τον *Simon Funk [14]* κατά τη διάρκεια του διαγωνισμού *Netflix Prize* [1],που συνήθως αναφέρεται ως *Funk SVD*. Παράδειγμα του *Funk SVD:*

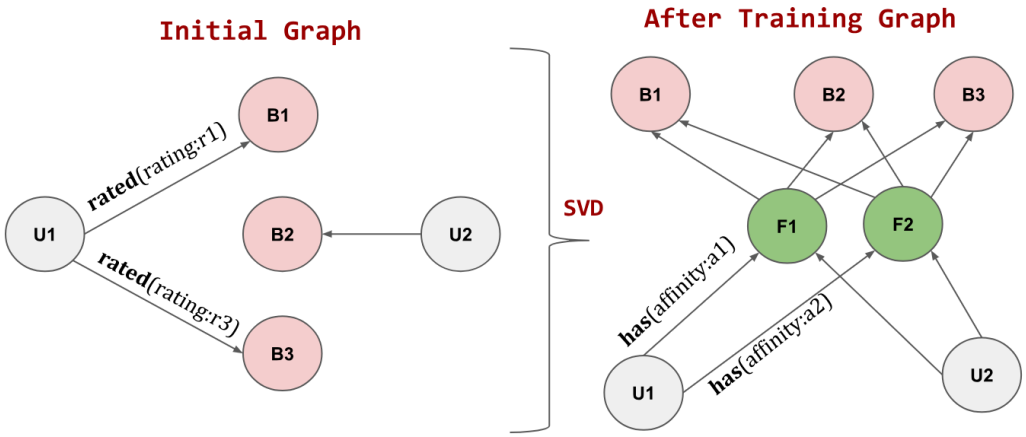
Συνήθως, η παραγοντοποίηση πινάκων υπολογίζει την προσέγγιση χαμηλής κατάταξης ελαχιστοποιώντας την τετραγωνική απώλεια σφάλματος. Θέλουμε να αποσυνθέσουμε αυτόν τον αραιό πίνακα σε δύο πίνακες χαμηλής κατάταξης που αντιπροσωπεύουν παράγοντες χρήστη και παράγοντες στοιχείων. Αυτό γίνεται χρησιμοποιώντας μια επαναληπτική προσέγγιση για ελαχιστοποίηση της λειτουργίας απώλειας. Μεταξύ των διαφόρων δυνατοτήτων, ο πιο συνηθισμένος τρόπος είναι η Στοχαστική Κάθοδος Κλίσης (*Stochastic Gradient Descent*). Αναλύουμε το πρόβλημα σε δύο φάσεις, την πρόβλεψη (*predicted*) και την εκπαίδευση (*training*), και δείχνουμε ότι και οι δύο φάσεις μπορούν να εκφραστούν ως προβλήματα μονοπατιού.

Ας υποθέσουμε τη συνήθη ρύθμιση όπου μας δίνεται ένα πίνακα αξιολογήσεων χρηστών (2x3 για το παράδειγμά μας) σε επιχειρήσεις με πολλές καταχωρήσεις που λείπουν. Θέλουμε να προβλέψουμε τη βαθμολογία προτίμησης του χρήστη U1 για την

επιχείρηση B2. Μετά την εφαρμογή του Funk-SVD, οι προκύπτοντες πίνακες παρουσιάζονται στο Σχήμα 4. Οι ίδιες πληροφορίες στο μοντέλο μας φαίνονται στο Σχήμα 5.



Σχήμα 4: Παραδοσιακή αποσύνθεση Funk-SVD σε πίνακα



Σχήμα 5: Αποσύνθεση Funk-SVD σε γράφο

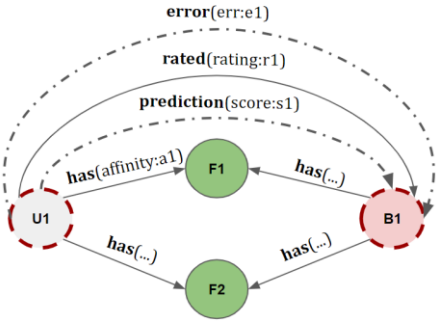
Θεωρούμε το γινόμενο των δύο πινάκων ως ακμές που δηλώνουν συγγένεια (*affinity*) (δηλαδή με ετικέτα έχει (*has*)) προς τους λανθάνοντες παράγοντες, οι οποίοι είναι νεοδημιουργημένοι κόμβοι στο γράφημα. Οι τελεστές για το τμήμα πρόβλεψης είναι αρκετά απλοί και ακολουθούν το γράφημα στο Σχήμα 6.

Χωρίς απώλεια γενικότητας, αν αγνοήσουμε τους όρους μεροληψίας, η προβλεπόμενη βαθμολογία υπολογίζεται ως $prediction(u,b) = \sum_f a_{uf}\ a_{fb}$, δηλ. ως πολλαπλασιασμός πίνακα που μπορεί να αναπαρασταθεί με τους ακόλουθους κανόνες.

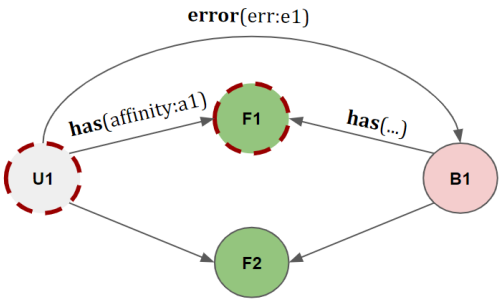
$$\%\ AGG\ and\ CON\ rules\ for\ rating\ prediction$$

$$\boldsymbol{prediction}\,(U(i),B(j);score) := [\,(\boldsymbol{has}\,(U(i),F(k);affinity = a1),$$

$$-\boldsymbol{has}\,(F(k),B(j);affinity = a2);rating = a1 * a2);score = addition\,(rating)\,].$$



Σχήμα 6: Πρόβλεψη (Prediction) γράφου



Σχήμα 7: Εκπαίδευση (Training) γράφου

Για το κομμάτι της εκπαίδευσης έχουμε δύο επιλογές.

Μπορούμε είτε να κάνουμε κλήσεις σε μια εξωτερική βιβλιοθήκη για να εκπαιδεύσουμε πρώτα το γράφημα και να εφαρμόσουμε το αποτέλεσμα στο μοντέλο μας, είτε μπορούμε να εφαρμόσουμε τους δικούς μας εκπαιδευτικούς τελεστές. Εμφανίζοντας το τελευταίο, δημιουργούμε το γράφημα στο Σχήμα 7. Στο σχήμα, δείχνουμε μόνο ακμές που σχετίζονται με ένα παράδειγμα εκπαίδευσης μεμονωμένου στοιχείου χρήστη. Αγνοούμε και πάλι τον όρο κανονικοποίησης. Ο ρυθμός εκμάθησης (learning rate) g και οι ακμές συγγένειας (affinity) αρχικοποιούνται σε ορισμένες επιλεγμένες τιμές. Έχουμε μία ακμή σφάλματος για να εξαγάγουμε το σφάλμα πρόβλεψης και να το χρησιμοποιήσουμε με τις αντίστοιχες ακμές συγγένειας για να τις ενημερώσουμε. Οι κανόνες ενημέρωσης (update rules) είναι αυτοί της στοχαστική κλίση κάθοδος (SGD). Παρουσιάζουμε τους κανόνες για την ενημέρωση του πρώτου πίνακα, δηλαδή της χρήστης-χαρακτηριστικό, καθώς οι κανόνες για την ενημέρωση του δεύτερου (επιχείρησης) είναι παρόμοιοι.

$$\%\ Rules\ for\ Training\ \ User-Feature\ edges$$

$$\%\ FUSE\ for\ computing\ error$$

$$\mathbf{error}\,(U(i),B(j)\,;err) := [\,(\mathbf{rated}\,(U(i),B(j)\,;rating=r1),$$
$$\mathbf{prediction}\,(U(i),B(j)\,;score=s1)\,;err=r1-s1\,].$$

$$\%\ Update\ AGG\ and\ CON\ rules\ for\ edges\ has\ (u,f)$$

$$\mathbf{has}\,(U(i),F(k);affinity) := [$$
$$(\ \mathbf{error}\,(U(i),B(j);\ err=e),\mathbf{has}\,(B(j),F(k);$$
$$affinity=a);affinity=g*e*a);affinity=addition(affinity)\,].$$

$$\%\ Rules\ for\ Training\ \ Business-Feature\ edges$$

$$\%\ Update\ AGG\ and\ CON\ rules\ for\ edges\ has\ (b,f)$$

$$\mathbf{has}\,(B(i),F(k);affinity) := [$$
$$(\ \mathbf{error}\,(B(i),U(j);\ err=e),\mathbf{has}\,(U(j),F(k);$$
$$affinity=a);affinity=g*e*a);affinity=addition(affinity)\,].$$

5. ΑΝΑΓΩΓΗ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ FUNK SVD ΣΤΗΝ ΓΛΩΣΣΑ RECGRAPH

Στην προηγούμενη ενότητα παρουσιάσαμε τον αλγόριθμο Funk-SVD και την μετατροπή του από ένα πρόβλημα υπολογισμού που βασίζεται σε πράξεις πινάκων, σε ένα πρόβλημα υπολογισμού που βασίζεται σε πράξεις πάνω στους κόμβους και τις ακμές ενός γράφου. Σε αυτήν την ενότητα θα επικεντρωθούμε στην υλοποίηση της γλώσσας RecGraph που περιεγράφηκε στο Κεφάλαιο 3 και την αναγωγή του Funk-SVD στην συγκεκριμένη γλώσσα. Στην ενότητα 5.1 επικεντρωνόμαστε στις λεπτομέρειες υλοποίησης της γλώσσας και συγκεκριμένα στην επέκταση της πλατφόρμας γραφημάτων Neo4j ώστε να υποστηρίζει τους τελεστές της γλώσσας RecGraph. Στην ενότητα 5.2 περιγράφουμε τις λεπτομέρειες αναγωγής του Funk-SVD αλγορίθμου στην άλγεβρα μονοπατιών RecGraph. Συγκεκριμένα περιγράφουμε κάθε εποχή εκπαίδευσης του Funk-SVD αλγορίθμου με ένα σύνολο πράξεων Edge Concatenation, Path Aggregation, και Path Fusion. Η ορθότητα της μεθόδου μας αποδεικνύεται στο πειραματικό σκέλος της πτυχιακής που ακολουθεί στο Κεφάλαιο 6.

5.1 Η γλώσσα υλοποίησης RecGraph

Πριν αναλύσουμε την υλοποίηση του Funk-SVD, θα δούμε τα στοιχεία συστήματος του πρέπει να έχει το RecGraph και πώς μπορεί να χρησιμοποιηθεί με βέλτιστο τρόπο. Για να μπορέσουμε να χρησιμοποιήσουμε τους τελεστές της γλώσσας RecGraph και να εκτελέσουμε ερωτήματα, δεδομένου ότι οι τελεστές είναι τελεστές γραφήματος που επιτρέπουν βελτιστοποιήσεις στην εκτέλεση ενός αλγορίθμου σε ένα γράφημα, χρειαζόμαστε μια μηχανή επεξεργασίας γραφήματος. Οι μηχανές γραφημάτων υποστηρίζουν φυσικά μοντέλα γραφημάτων, καθώς αποθηκεύουν, επεξεργάζονται και αναζητούν σχέσεις αποτελεσματικά. Οποιοδήποτε σύστημα βάσης δεδομένων γραφημάτων θα μπορούσε να παίξει το ρόλο της μηχανής γραφημάτων στο πλαίσιο μας, αλλά επιλέξαμε το Neo4j επειδή είναι μία ευρέως χρησιμοποιούμενη και καλά τεκμηριωμένη μηχανή γραφημάτων, μεταξύ των άλλων διαθέσιμων υποψηφίων.

Το Neo4j [3] είναι μια βάση δεδομένων γραφημάτων ανοιχτού κώδικα, NoSQL, που παρέχει ένα σύστημα υποστήριξης συμβατό με ACID/συναλλαγές για τις εφαρμογές σας. Είναι μια εγγενής βάση δεδομένων γραφημάτων. Σε αντίθεση με τον τρόπο που τα δεδομένα είναι διατεταγμένα στις παραδοσιακές βάσεις δεδομένων, δηλαδή σε γραμμές, στήλες και πίνακες, έχει μια δομή που ορίζεται από αποθηκευμένες σχέσεις μεταξύ εγγραφών δεδομένων που είναι αρκετά ευέλικτη.

Το Neo4j αναφέρεται ως βάση δεδομένων εγγενών γραφημάτων επειδή εφαρμόζει αποτελεσματικά το μοντέλο γραφήματος ιδιοτήτων μέχρι το επίπεδο αποθήκευσης.

Αυτό σημαίνει ότι τα δεδομένα αποθηκεύονται ακριβώς όπως τοποθετήθηκαν και η βάση δεδομένων χρησιμοποιεί δείκτες για την πλοήγηση και τη διάσχιση του γραφήματος. Σε αντίθεση με την επεξεργασία γραφημάτων ή τις βιβλιοθήκες μνήμης, το Neo4j παρέχει επίσης πλήρη χαρακτηριστικά βάσης δεδομένων, συμπεριλαμβανομένης της συμμόρφωσης συναλλαγών ACID, της υποστήριξης συμπλέγματος και του χρόνου εκτέλεσης, καθιστώντας το κατάλληλο για χρήση γραφημάτων για δεδομένα σε σενάρια παραγωγής. Κάθε εγγραφή δεδομένων ή κόμβος αποθηκεύει άμεσους δείκτες σε όλους τους κόμβους με τους οποίους είναι συνδεδεμένος. Το Neo4j μπορεί να εκτελεί ερωτήματα με πολλαπλές ακμές πιο γρήγορα και με μεγαλύτερο βάθος από άλλες βάσεις δεδομένων. Αυτό οφείλεται στον απλό, αλλά αποτελεσματικό σχεδιασμό βελτιστοποίησης.

Το Neo4j προσφέρει έναν αριθμό επιλογών για τη σύνδεση και την υποβολή ερωτημάτων στη βάση δεδομένων γραφημάτων Neo4j. Μία από τις επιλογές είναι να χρησιμοποιήσετε την ενσωματωμένη, επίσημα υποστηριζόμενη γλώσσα ερωτημάτων γραφήματος, Cypher [4]. Η Cypher είναι σαν την SQL μια δηλωτική, κειμενική γλώσσα ερωτημάτων, αλλά για γραφήματα. Περιγράφει οπτικά μοτίβα σε γραφήματα χρησιμοποιώντας το ASCII-Art συντακτικό. Χρησιμοποιείται για την υποβολή ερωτημάτων στο γράφημα, καθώς και για την ενημέρωση του γραφήματος. Το συντακτικό του Cypher παρέχει έναν οπτικό και λογικό τρόπο αντιστοίχισης μοτίβα κόμβων και σχέσεων στο γράφημα. Αποτελείται από προτάσεις, λέξεις-κλειδιά και εκφράσεις όπως κατηγορήματα και συναρτήσεις, πολλές από τις οποίες θα είναι γνωστές (όπως WHERE, ORDER BY, SKIP LIMIT, AND, p.unitPrice > 10). Σε αντίθεση με την SQL, το Cypher έχει να κάνει με την έκφραση μοτίβων γραφημάτων. Ένα παράδειγμα ερωτήματος στο Cypher παρέχεται παρακάτω:

```
% Cypher example clause
MATCH (:Person {name: 'Tom Hanks'})-[:DIRECTED]->(movie:Movie)
RETURN movie.title
```

Εκτός από το Cypher, μια άλλη επιλογή είναι να χρησιμοποιήσετε ένα από τα υποστηριζόμενα προγράμματα οδήγησης για τη δημιουργία API που παρέχουν την επιθυμητή λειτουργικότητα. Το Neo4j υποστηρίζει επίσημα προγράμματα οδήγησης για .Net, Java, JavaScript, Go και Python. Ωστόσο, παρέχονται πολλά περισσότερα, συμπεριλαμβανομένων των PHP, Ruby, R, Erlang, Clojure και C/C++. Επιλέξαμε το Java API για την υλοποίησή μας επειδή προσφέρει μεγαλύτερη ευελιξία, επιτρέπει βαθύτερες βελτιστοποιήσεις και αποτελεσματικά περισσότερο έλεγχο της υλοποίησης, σε αντίθεση με το Cypher, το οποίο είναι κάτι κρίσιμο από την πλευρά του προγραμματιστή. Η επιλογή της Java έναντι των άλλων προσφερόμενων υποστηριζόμενων API, έγινε με προσωπική προτίμηση, αλλά θα μπορούσε να είχε χρησιμοποιηθεί οποιοδήποτε άλλο πρόγραμμα οδήγησης.

Αρχιτεκτονική Πρωτότυπου Μια επισκόπηση των βασικών στοιχείων της πρώτης έκδοσης του Recgraph. Όπως αναφέραμε, το RecGraph είναι γραμμένο σε Java και τρέχει πάνω από το Neo4j[3], ένα σύστημα διαχείρισης βάσεων δεδομένων γραφημάτων με εγγενή αποθήκευση και επεξεργασία γραφημάτων, το οποίο είναι υλοποιημένο από την συνάδελφο Μαριαλένα Κυριακίδη [6][7] .

Η είσοδος του συστήματος RecGraph είναι ένα πρόγραμμα στη γλώσσα RecGraph. Το πρόγραμμα RecGraph αποστέλλεται στον Parser, ο οποίος είναι υπεύθυνος για την ανάγνωση της ακολουθίας των παρεχόμενων τελεστών και τη δημιουργία του αντίστοιχου AST (αφηρημένο δέντρο σύνταξης) για το πρόγραμμα. Στη συνέχεια, το AST μεταβιβάζεται στο Optimizer. Ο ρόλος του βελτιστοποιητή είναι να εκμεταλλεύεται τυχόν αλγεβρικές ιδιότητες που μπορεί να έχουν οι τελεστές, προκειμένου να τροποποιήσει την ακολουθία εκτέλεσης που παράγεται από τον αλγόριθμο που αρχικά παρεχόταν ως είσοδος. Ο βελτιστοποιητής πρέπει στη συνέχεια να εξερευνήσει τον χώρο λειτουργίας, προκειμένου να δημιουργήσει ισοδύναμα AST ως εναλλακτικά σχέδια εκτέλεσης. Ως τελικό βήμα, ο βελτιστοποιητής πρέπει να επιλέξει ένα τελικό AST μεταξύ των υποψηφίων, με δυνητικά πολύ καλύτερη ή ακόμα και βέλτιστη απόδοση. Το τελικό AST μεταβιβάζεται στο Execution Engine, το οποίο το ερμηνεύει και επικοινωνεί με τον μηχάνημα γραφήματος από κάτω. Όπως αναφέραμε ήδη, η αναζήτηση μιας βάσης δεδομένων Neo4j μπορεί να γίνει είτε με Cypher, είτε με πολλές διαφορετικές γλώσσες προγραμματισμού σε χαμηλότερο επίπεδο. Επιλέξαμε την Java για να έχουμε

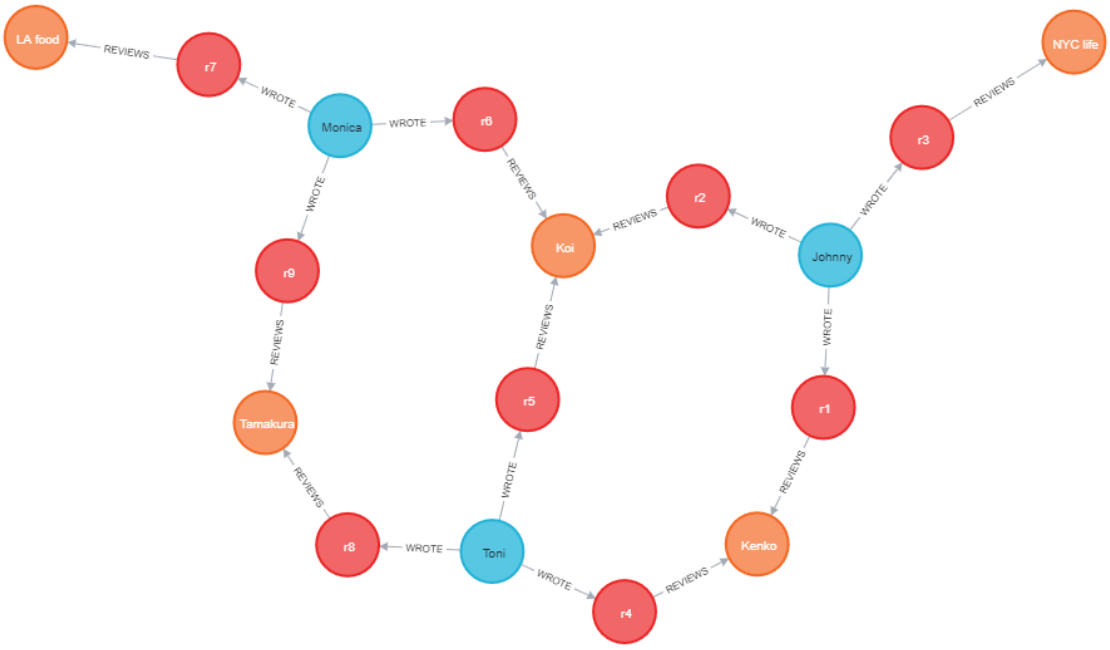
περισσότερο έλεγχο στον καλύτερο τρόπο εκτέλεσης ερωτημάτων χρησιμοποιώντας τη λογική τελεστών μας.

5.2 Υλοποίηση μεθόδου Funk-SVD στη γλώσσα RecGraph

Όπως έχει ειπωθεί και σε προηγούμενο κεφάλαιο, η μέθοδος SVD μπορεί να μας επιτρέψει να προβλέψουμε μια αξιολόγηση για κάθε ζεύγος χρήστη-επιχείρηση. Εάν μπορούμε να προβλέψουμε αξιολογήσεις με χαμηλό σφάλμα, μπορούμε να χρησιμοποιήσουμε αυτήν την προβλεπόμενη αξιολόγηση για να βρούμε την επιχείρηση που σχετίζεται με την υψηλότερη προβλεπόμενη αξιολόγηση. Κάτι που το SVD μπορεί να είναι χρήσιμο γιατί μας επιτρέπει να χρησιμοποιούμε μετρικές (metrics) που βασίζονται σε οπισθοδρόμηση (regression) όπως το mean square error (MSE) ή το mean average error (MAE) για να αξιολογήσουμε την απόδοση.

Επειδή ο SVD [24] από μόνος του μπορεί να επιτύχει καλά αποτελέσματα μόνο σε μη αραιά δεδομένα (non-sparse data), η υλοποίηση μας βασίζεται στον αλγόριθμο Funk SVD[14], ο οποίος αγνοεί τις τιμές που λείπουν και θα βρει έναν τρόπο να υπολογίσει τους λανθάνοντες παράγοντες χρησιμοποιώντας μόνο τις τιμές που γνωρίζουμε.

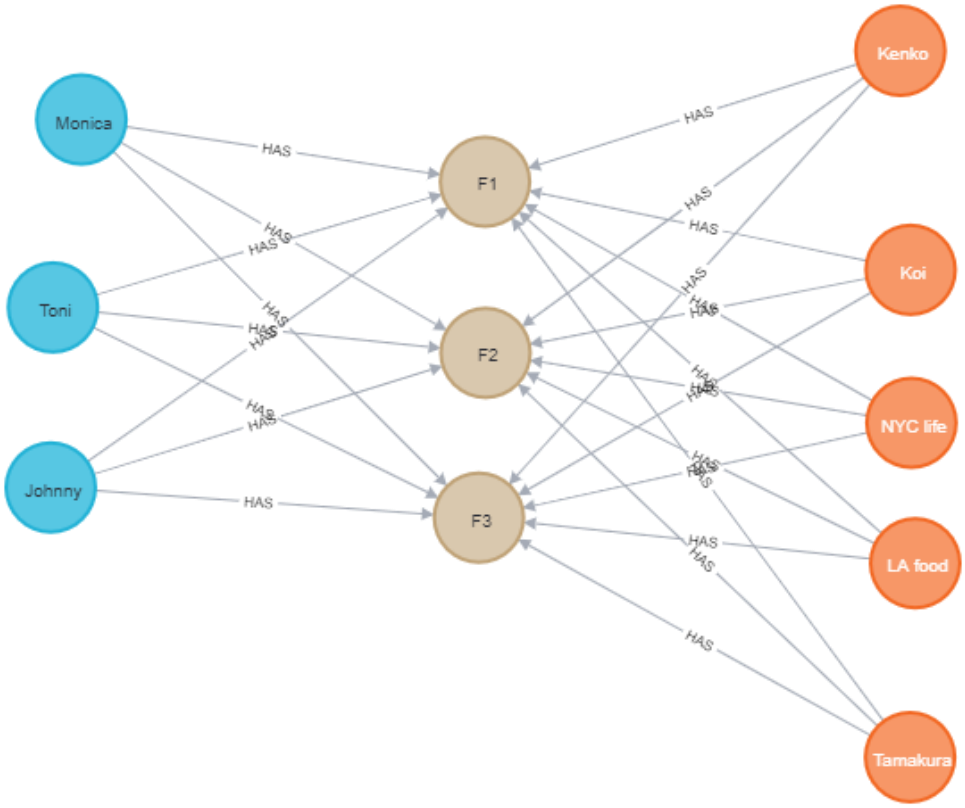
Για να εκπαιδεύσουμε και προβλέψουμε το μοντέλο μας χρησιμοποιούμε έναν υπογράφο "Yelp" που περιέχει *users*, *reviews*, *businesses*. Οι χρήστες *γράφουν* (*write*) αξιολογήσεις που *αξιολογούν(review)* επιχειρήσεις. Λάβετε υπόψη ότι ένας χρήστης μπορεί να έχει πολλές αξιολογήσεις για την ίδια επιχείρηση. Αρχικά δημιουργούμε ένα γράφο με κόμβους Χρήστη (User) , Επιχείρηση (Business), Αξιολόγηση (Review) και δημιουργούμε συνδέσεις (ακμές) για κάθε αξιολόγηση από το Χρήστη προς μία Επιχείρηση, το σχήμα8 αναπαριστά μία τέτοια αρχικοποίηση. Ο γράφος δεν έχει συγκεκριμένες πληροφορίες σχετικά με τον χρήστη ή την επιχείρηση, μόνο τα σχετικά αναγνωριστικά και τις βαθμολογίες τους. Η χρήση του Funk-SVD σε αυτόν τον γράφο μπορεί να μας επιτρέψει να βρούμε λανθάνοντα χαρακτηριστικά (latent factor) που σχετίζονται με τις επιχειρήσεις και τους χρήστες.



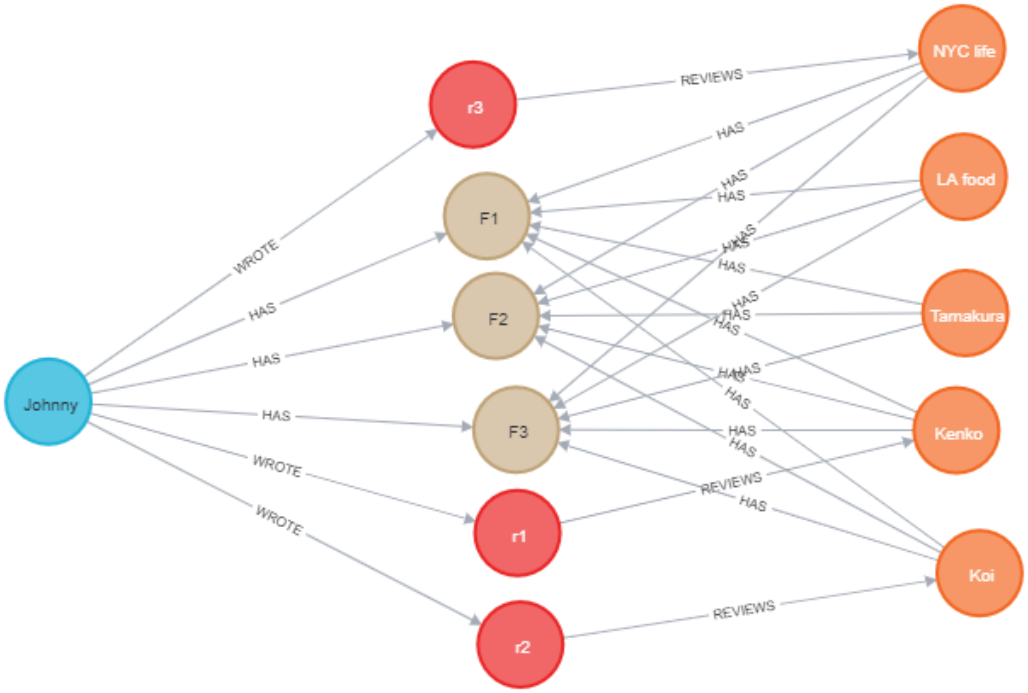
Σχήμα 8: Παράδειγμα αρχικοποίησης γράφου στο Neo4j

Για να επιτευχθεί αυτή η προσέγγιση παραγοντοποίησης γράφου με το Funk-SVD, ακολουθούνται τα παρακάτω βήματα για την εκπαίδευση (training) του γράφου μας :

1. Δημιουργούμε ενδιάμεσους κόμβους, όσο και ο αριθμός των λανθάνοντων παραγόντων που έχουμε επιλέξει, και δημιουργούμε συνδέσεις (relationship) μεταξύ κάθε παράγοντα (factor)-χρήστη και παράγοντα (factor)-επιχείρηση και αρχικοποιούμε τις συνδέσεις με τυχαίους αριθμούς με χαρακτηριστικό «συγγένεια» (affinity) (βλέπε Σχήμα 9, 10).



Σχήμα 9: Συνδέσεις User-Factor και Business-Factor μετά το βήμα 1



Σχήμα 10: Όλες οι συνδέσεις ενός χρήστη μετά το βήμα 1

2. Τα παρακάτω βήματα γίνονται για κάθε αξιολόγηση που υπάρχει μεταξύ χρήστη-επιχείρηση. Με την χρήση του CON του Recgraph διασχίζουμε και συνενώνουμε τις ακμές wrote, reviews μεταξύ των κόμβων χρήστη-επιχείρηση και δημιουργούμε μία ακμή temp_r με χαρακτηριστικό την βαθμολογία.
3. Για κάθε παράγοντα μέσω του CON διασχίζουμε και συνενώνουμε τις ακμές has, has μεταξύ των κόμβων χρήστη-επιχείρηση και δημιουργούμε προσωρινές ακμές temp_p με χαρακτηριστικό το γινόμενο των χαρακτηριστικών “συγγένεια" των δύο ακμών (a1_fi *a2_fi).
4. Με την χρήση του AGG, παίρνουμε το άθροισμα των γινομένων από κάθε παράγοντα και δημιουργούμε την ακμή πρόβλεψης (prediction).
5. Με την χρήση του FUSE, παίρνουμε την διαφορά των ακμών αξιολόγησης, πρόβλεψης και δημιουργούμε την ακμή σφάλματος (error) μεταξύ των κόμβων χρήστη-επιχείρηση. (Βλέπε Σχήμα 11)

Σχήμα 11: Όλες οι συνδέσεις ενός χρήστη-επιχείρηση μετά το βήμα 5

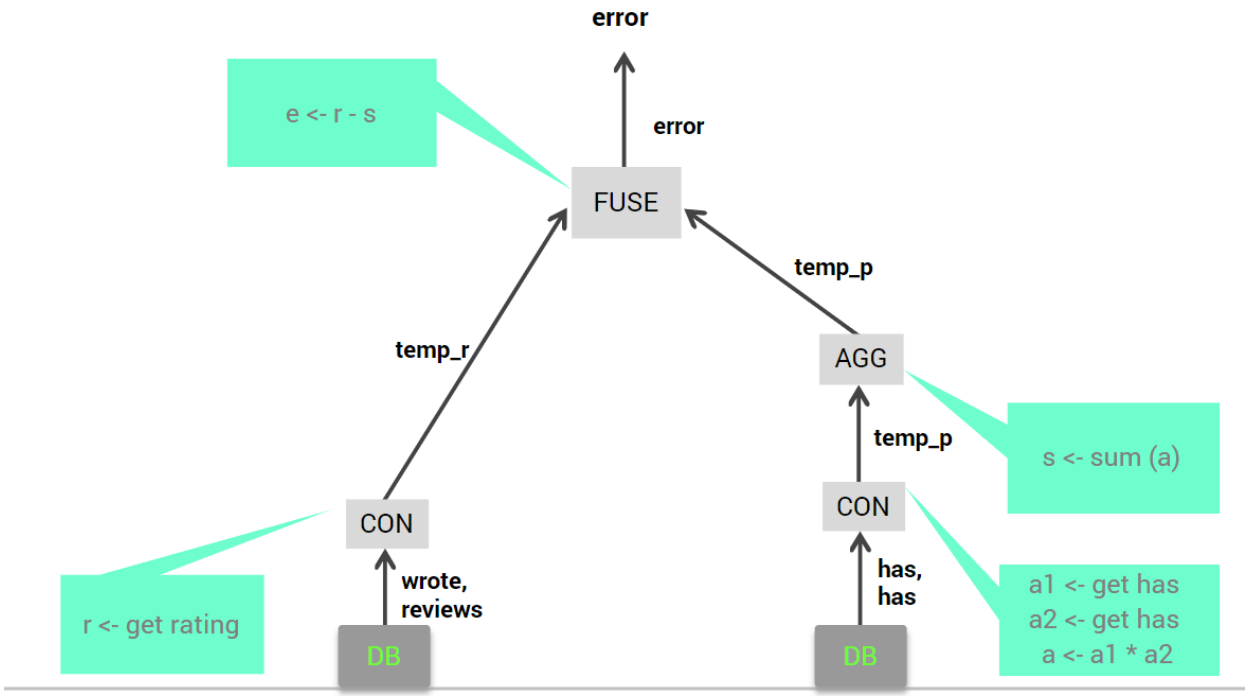
6. Για να ελαχιστοποιήσουμε το σφάλμα, ακολουθούνται οι κανόνες ανανέωσης του SGC, δηλ. ανανεώνουμε τιμές των ακμών has ακολουθώντας της εξής φόρμουλα:

   τιμή_χρηστη_παράγοντα_i = τιμή_χρηστη_παράγοντα_i +

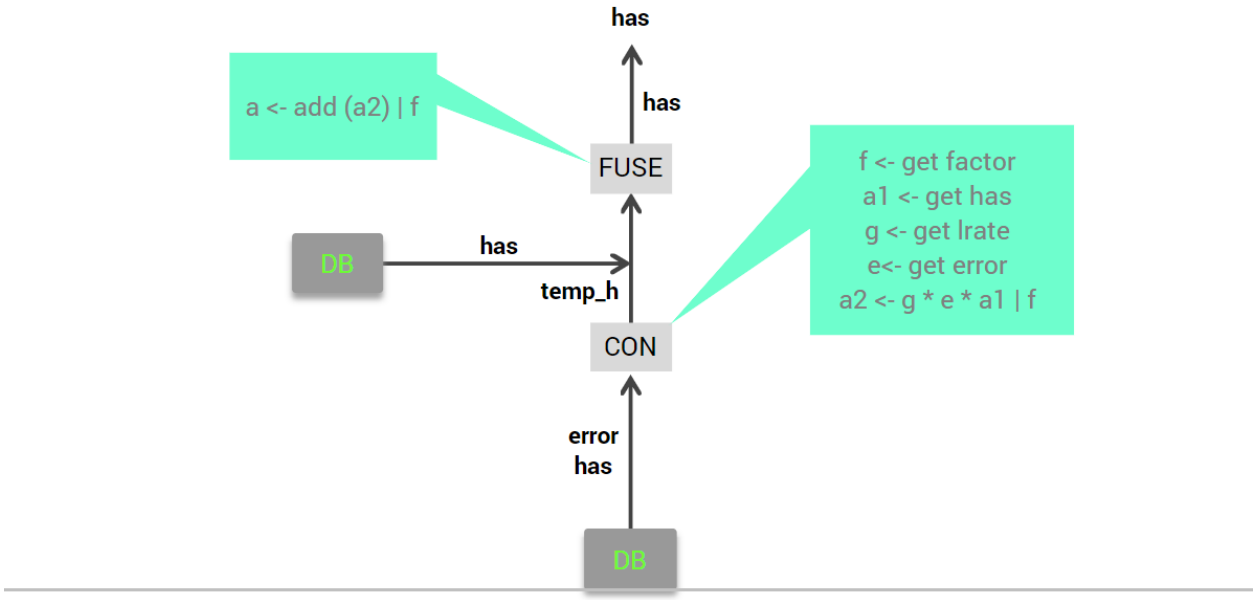
   (2 * ρυθμό_εκμάθησης * τιμή_σφάλματος * τιμή_επιχείρηση_παράγοντα_i), όπου ο ρυθμός εκμάθησης αρχικοποιείται στην αρχή του προγράμματος με μία πολύ μικρή τιμή πχ. 0.001. Την παραπάνω φόρμουλα την εφαρμόζουμε πρώτα για κάθε ακμή χρήστη-παράγοντα και στη συνέχεια για κάθε ακμή επιχείρηση-παράγοντα.
7. Με την χρήση του FUSE ανανεώνουμε τις τιμές των ακμών χρήστη-παράγοντα και επιχείρηση-παράγοντα.

Τα βήματα 2-5 επεικονίζοται στο Σχήμα 12. Τα βήματα 6-7 απεικονίζεται στο Σχήμα 13. Και στο σχήμα 14 επεικονίζεται το συνολικό πλάνο εκτέλεσης μίας επανάληψης για ένα χρήστη-επιχείρηση-αξιολόγηση. Στους σχολιασμούς δίπλα στους τελεστές

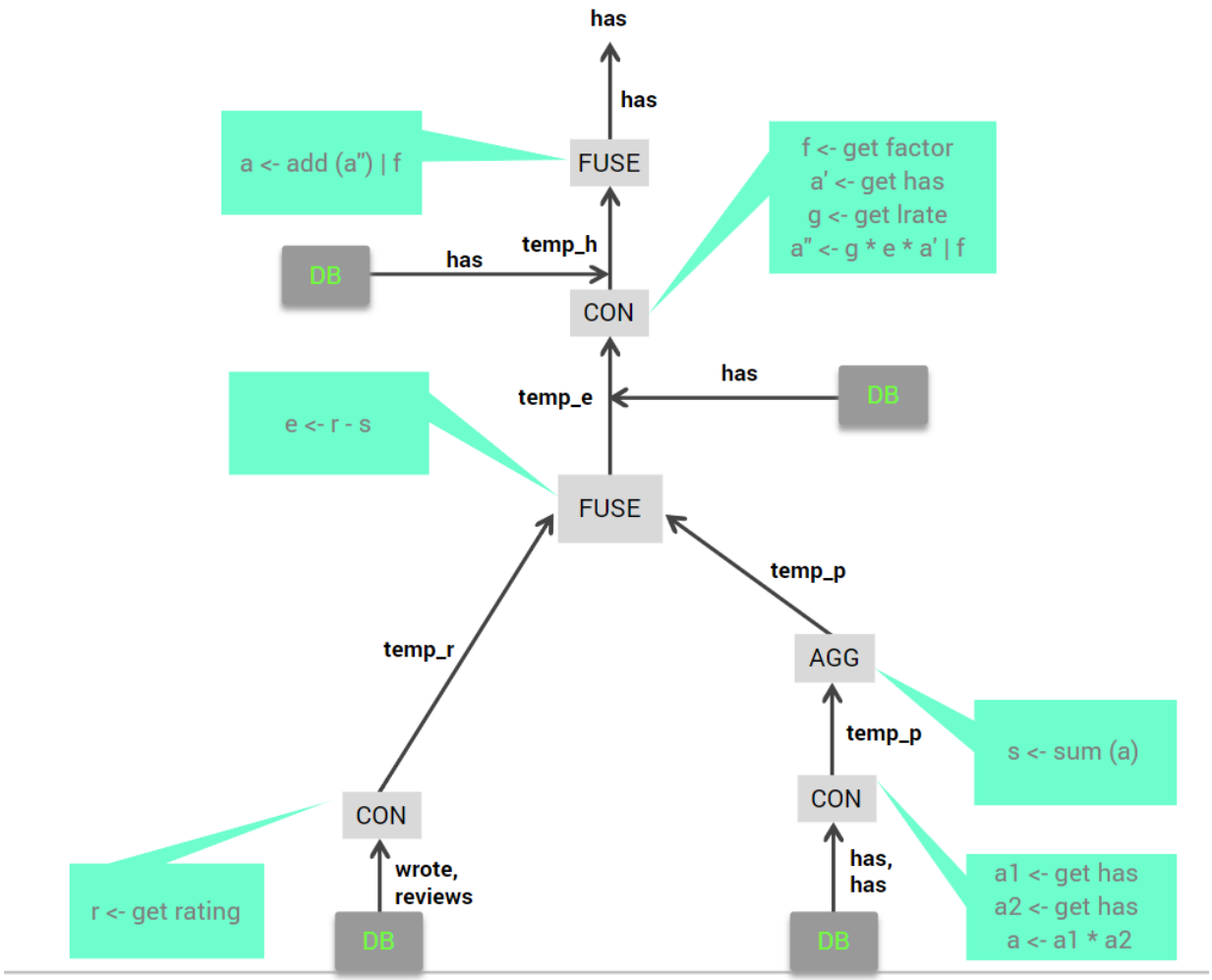
στο πλάνο, περιγράφεται η υλοποιούμενη συνάρτηση του τρέχοντος τελεστή και στα βέλη οι ακμές εισόδου/εξόδου του καθενός.



Σχήμα 12: Πλάνο εκτέλεσης - Εύρεσης Σφάλματος Πρόβλεψης



Σχήμα 13: Πλάνο εκτέλεσης - Κανόνες Ενημέρωσης



Σχήμα 14: Τελικό/Συνολικό Πλάνο εκτέλεσης μίας επανάληψης

Η εκπαίδευση του μοντέλου μας τελειώνει είτε όταν το μοντέλο ολοκληρώσει ένα συγκεκριμένο αριθμό επαναλήψεων, είτε σταματάει νωρίτερα όταν η διαφορά μεταξύ δύο συνεχόμενων τιμών του της ρίζας του μέσου Τετράγωνου Σφάλματος (RMSE) είναι ένα πολύ μικρό έψιλον (πχ. 1×10−5) ή όταν η τιμή του σταματάει να μικραίνει.

Μετά την εκπαίδευση, μπορεί να χρησιμοποιηθεί το μοντέλο για να κάνει πρόβλεψη για ένα χρήστη και να του προτείνει επιχειρήσεις βάσει των εκπαιδευμένων βαρών των παραγόντων. Σχετική απεικόνιση ενός σχετικού παραδείγματος στο Σχήμα 15.

Σχήμα 15: Πρόβλεψη 3 καλύτερων επιχειρήσεων για τον χρήστη “Johnny”

6. ΑΞΙΟΛΟΓΗΣΗ ΣΥΣΤΗΜΑΤΩΝ ΣΥΣΤΑΣΕΩΝ ΓΙΑ ΔΕΔΟΜΕΝΑ "MOVIELENS"

Σε αυτό το Κεφάλαιο εξετάζουμε πειραματικά την ορθότητα της αναγωγής τους αλγορίθμου Funk-SVD στην Άλγεβρα Μονοπατιών RecGraph. Συγκεκριμένα εξετάζουμε τα δεδομένα του dataset Movielens το οποίο περιγράφει αξιολογήσεις χρηστών για ταινίες. Χωρίζουμε τα δεδομένα μας σε ένα σύνολο εκπαίδευσης και ένα σύνολο πρόβλεψης, τα οποία χρησιμοποιούνται αντίστοιχα για την προσαρμογή τον παραμέτρων του Funk-SVD αλγορίθμου και την αξιολόγηση της εκπαίδευσης σε πραγματικά δεδομένα βαθμολόγησης ταινιών. Παρατηρούμε ότι με την επιλογή των κατάλληλων παραμέτρων εκπαίδευσης, η αναγωγή στην RecGraph γλώσσα παράγει τα ίδια αποτελέσματα με τον αρχικό Funk-SVD αλγόριθμο σε πίνακα έχοντας μέσο σφάλμα 0.93 στην αξιολόγηση ταινιών σε μία κλίμακα βαθμολόγησης από 1 έως 5.

6.1 Δεδομένα Movielens

Τα δεδομένα Movielens[5], συλλέχθηκαν από το GroupLens Research Project[20] στο Πανεπιστήμιο της Μινεσότα. Αυτό το σύνολο δεδομένων αποτελείται από: (α) 100.000 αξιολογήσεις (1-5) από 943 χρήστες σε 1682 ταινίες. (β) Κάθε χρήστης έχει βαθμολογήσει τουλάχιστον 20 ταινίες. (γ) Απλές δημογραφικές πληροφορίες για τους χρήστες (ηλικία, φύλο, επάγγελμα, Τ.Κ.). Τα δεδομένα συλλέχθηκαν μέσω του ιστότοπου MovieLens κατά την επτάμηνη περίοδο από τις 19 Σεπτεμβρίου, 1997 έως 22 Απριλίου 1998. Αυτά τα δεδομένα έχουν καθαριστεί, χρήστες που είχαν λιγότερες από 20 αξιολογήσεις ή δεν είχαν πλήρη δημογραφικά στοιχεία πληροφορίες αφαιρέθηκαν από αυτό το σύνολο δεδομένων.

Λεπτομερείς περιγραφές του το αρχείο δεδομένων αξιολόγης παρακάτω:

- userId, μοναδικός αριθμός χρήστη
- itemId, μοναδικός αριθμός ταινίας (αντικείμενο)
- rating, 100.000 αξιολογήσεις από 943 χρήστες, σε 1682 ταινίες, με βαθμολόγηση από 1 έως 5.

Παρατηρείται ότι ο αριθμός των αξιολογήσεων είναι αρκετά αραιός (93,6% για την ακρίβεια), καθώς συγκεντρώνει μόνο 100.000 αξιολογήσεις από ένα πιθανό 1.586.126 (943*1682) σύνολο δεδομένων. Για να μπορέσεις να κάνεις πρόβλεψη για κάθε χρήστη και να του προτείνεις ένα σύνολο ταινιών, δηλαδή να βρεθούν όλες τις τριάδες (χρήστης, ταινία , βαθμολογία) και να γίνει πρόβλεψη βαθμολογιών χρηστών για ταινίες, με βάση προηγούμενες αξιολογήσεις χωρίς άλλες πληροφορίες για τους χρήστες ή τις ταινίες. Κάτι που είχε θέσει το Netflix στο διαγωνισμό του Netflix Prize[1], όπου έθεσε ένα «κουίζ» που αποτελείται από ένα σωρό ερωτηματικά τοποθετημένα σε προηγουμένως κενές θέσεις και η δουλειά των συμμετεχόντων ήταν να συμπληρώσει τις καλύτερες αξιολογήσεις στη θέση τους.

Για τις ανάγκες της παρούσας υλοποίησης τα δεδομένα, έχουν χωριστεί σε δεδομένα εκπαίδευσης (80%) και δεδομένα πρόβλεψης (20%). Επίσης, έχει επιλεχτεί το μέσο τετράγωνο σφάλμα ως μέτρο ακρίβειας, που σημαίνει ότι αν μαντέψετε 1,5 και η πραγματική βαθμολογία ήταν 2, το σφάλμα θα είναι (2-1,5)^2 βαθμούς ή 0,25. (Στην πραγματικότητα, προσδιορίζεται το σφάλμα μέσου τετραγώνου της ρίζας, το οποίο αναφέρεται ως *RMSE*).

Με την χρήση της Αποσύνθεσης Ιδιαζουσών Τιμών (SVD), καθορίζουμε ένα μοντέλο για το πώς συγκεντρώνονται τα δεδομένα από έναν μικρότερο αριθμό τυχαίων

παραμέτρων, και εξάγουμε μιας μεθόδου συμπερασμάτων από τα δεδομένα αυτά, τα οποία είναι οι πραγματικοί μας παράμετροι. Υποθέτουμε ότι η βαθμολογία ενός χρήστη για μια ταινία αποτελείται από ένα άθροισμα προτιμήσεων σχετικά με τις διάφορα χαρακτηριστικά αυτής της ταινίας.

6.2 Εκπαίδευση δεδομένων:

Αρχικά γίνεται αξιολόγηση για κάθε υπάρχουσα βαθμολογία στη βάση δεδομένων εκπαίδευσης, για τον υπολογισμό του σφάλματος. Και στη συνέχεια ενημερώνονται οι τιμές των λανθάνοντων παραγόντων. Με *α να* είναι ο ρυθμός εκμάθησης, με την τιμή του να κυμαίνεται μεταξύ 0,001-0,008. Και *error* να είναι το υπολειπόμενο σφάλμα από την τρέχουσα πρόβλεψη (*prediction*). Το βήμα εκμάθησης συνίσταται στην εκτέλεση του αλγόριθμου SGD όπου για κάθε γνωστή βαθμολογία οι λανθάνοντες παράγοντες *(P, Q)* ενημερώνονται ως εξής:

$$prediction_{(u,i)} = \sum_f P_{u,f} * Q_{i,f}$$
$$error_{(u,i)} = rating_{(u,i)} - prediction_{(u,i)}$$
$$P_{u,f} = P_{u,f} + a * error_{(u,i)} * Q_{i,f}$$
$$Q_{i,f} = Q_{i,f} + a * error_{(u,i)} * P_{u,f}$$

Στην προσπάθεια να περιοριστεί η υπερπροσαρμογή (*overfitting*), εισάγεται στο βήμα εκμάθησης ο όρος κανονικοποίησης *λ (λάμδα*). Έτσι οι λανθάνοντες παράγοντες ενημερώνονται ως εξής:

$$P_{u,f} = P_{u,f} + a * (error_{(u,i)} * Q_{i,f} - \lambda * P_{u,f})$$
$$Q_{i,f} = Q_{i,f} + a * (error_{(u,i)} * P_{u,f} - \lambda * Q_{i,f})$$

Για τις ανάγκες του μοντέλου, η τιμή του όρου κανονικοποίησης λ αρχικοποιήθηκε στο 0.02.
Παρατηρείται ότι όσο μικραίνει η τιμή του α τόσο αυξάνεται ο αριθμός των επαναλήψεων (epoch). Κάνοντας διάφορα πειράματα με την τιμή του α, προτάθηκε ως τιμή το 0.008.

6.3 Πρόβλεψη δεδομένων:

Για το κομμάτι την πρόβλεψης, χρησιμοποιήθηκε το 20% των δεδομένων. Για τη βελτίωση την πρόβλεψης, πειραματίστηκα με το να περικόψω την πρόβλεψη στο εύρος 1-5 μετά την προσθήκη κάθε χαρακτηριστικού. Δηλαδή, κάθε χαρακτηριστικό περιορίζεται μόνο στην ταλάντευση της βαθμολογίας εντός του έγκυρου εύρους και κάθε περίσσεια πέρα από αυτό χάνεται αντί να μεταφερθεί. Έτσι, εάν το πρώτο χαρακτηριστικό προτείνει +10 σε μια κλίμακα 1-5 και το δεύτερο χαρακτηριστικό προτείνει -1, τότε αντί να πάρει ένα 5 για την τελική αποκομμένη βαθμολογία, παίρνει ένα 4 επειδή η βαθμολογία κόπηκε μετά από κάθε στάδιο. Η διαισθητική λογική εδώ είναι ότι τείνουμε να κρατάμε το πάνω μέρος της κλίμακας για την τέλεια ταινία και το κάτω μέρος για μια χωρίς καμία απολύτως αξιολόγηση, και έτσι υπάρχει ένα είδος μέτρησης από τις ακμές σε κάθε χαρακτηριστικό ανεξάρτητα. Το κόψιμο (clipping) εγγυάται ότι θα βελτιώσει την απόδοσή μας συνολικά.

6.4 Εκτέλεση πειραμάτων:

Για την εκτέλεση των πειραμάτων, επιλέξαμε ένα υποσύνολο των δεδομένων 3000~ αξιολογήσεις, έχουν επιλεχθεί οι χρήστες με τουλάχιστον 10 αξιολογήσεις και οι ταινίες που έχουν τουλάχιστον 50 φορές αξιολογηθεί, με τα δεδομένα μας να είναι αραιά κατά 98,6%. Και χωρίσαμε τα στοιχεία τους σε εκπαίδευση και πρόβλεψη.
Για να βρεθούν οι καλύτεροι υπερ-παράμετροι έγιναν διάφορα πειράματα και με την βοήθεια την άπληστης αναζήτησης:

- Αριθμός παραγόντων (factor number) := [15,30,45,55]
- Αριθμός επαναλήψεων (epoch number) :=[1:100]
- Ρυθμός εκμάθησης (learning rate) := [0.001, 0.004, 0.008]
- Όρος κανονικοποίησης (regularization term) := [0, 0.02]

Στον Πίνακα 1 μπορούμε να δούμε τα πειράματα με διαφορετικές υπερ-παραμέτρους και τα αποτελέσματα αξιολόγησής τους:

Πίνακας 1: Αποτελέσματα από τα πειράματα Funk-SVD

	FACTORS	EPOCHS	LRATE	REG.TERM	RMSE	TIME (Iter)
1	15	20	0.008	0.02	1.837	42 sec
2	30	20	0.008	0.02	1.339	95 sec
3	45	20	0.008	0.02	1.082	122 sec
4	**55**	**20**	**0.008**	**0.02**	**0.932**	**177 sec**
5	15	42	0.004	0.02	1.572	52 sec
6	30	42	0.004	0.02	1.632	122 sec

6.5 Παρατηρήσεις και ενδεικτικά αποτελέσματα:

Παρατηρούμε ότι η ιδανική τετράδα υπερ-παραμέτρων είναι η (55, 20, 0.008, 0.02) , τόσο από θέμα χρόνου εκτέλεσης, όσο και από θέμα ακρίβειας σφάλματος.

Κάνοντας δύο διαφορετικές υλοποιησείς με τις ίδιες υπερ-παραμέτρους στα ίδια δεδομένα, την μία με το μοντέλο μας του RecGraph μέσω του neo4j και την δεύτερη υλοποιόντας τον ίδιο αλγόριθμο (Funk SVD) με την παραγοντοποίση πινάκων (MF). Βλέπουμε στον πίνακα 2 ότι τα αποτελέσματα είναι αρκετά κοντά (οι βαθμολογίες των ταινιών που έγιναν πρόβλεψη ήταν μεταξύ 4 και 5).

Πίνακας 2: Αποτελέσματα πρόβλεψης ενός χρήστη με το μοντέλο RecGraph και MF

	Top 5 Movies (Recgraph) for UserId 101	Top 5 Movies (Matrix) for UserId 101
1	Man on the Moon (1999)	Doors, The (1991)
2	Doors, The (1991)	Man on the Moon (1999)
3	Boiler Room (2000)	Boiler Room (2000)
4	Happiness (1998)	Beach, The (2000)
5	Sweet Hereafter, The (1997)	Sweet Hereafter, The (1997)

Να σημειωθεί ότι λόγω των ιδιαιτεροτήτων του neo4j οι χρόνοι εκτέλεσεις είναι 1/100, με το μοντέλο MF να είναι πολύ πιο ταχύτερο από το μοντέλο RecGraph. Αλλά με το σφάλμα rmse στο μοντέλο MF να είναι 1.01, λίγο χειρότερο από το μοντέλο RecGraph που έχει σφάλμα 0.93.

7. ΣΥΜΠΕΡΑΣΜΑΤΑ

Στη πτυχιακή αυτή έγινε μία συνοπτική ανασκόπηση στο πως λειτουργούν τα συστήματα συστασεων, ποια είναι η χρήση και τα οφέλη του Funk-SVD. Ποιοι εναλλακτικοί μέθοδοι υπάρχουν. Τι είναι το μοντέλο RecGraph και πως συνδέετα με τον Funk-SVD αλγόριθμο. Παρατηρήθηκε ότι υπάρχουν κάποιες βελτιώσεις που μπορούν να γίνουν.

Αν και η μέθοδος Funk-SVD σε γράφο μας επιτρέπει εύκολα να βρούμε έναν τρόπο να αξιολογήσουμε ένα σύστημα συστάσεων και να δημιουργήσουμε έναν καλό γράφο για να κάνουμε συστάσεις ακόμα κι αν έχουμε πολύ αραιά δεδομένα. Δεν πρέπει να χρησιμοποιείται μόνη της, γιατί μπορεί να αντιμετωπίσουμε ένα κοινό πρόβλημα στα συστήματα συστάσεων που ονομάζεται «Πρόβλημα Ψυχρής Εκκίνησης». Αυτό το πρόβλημα σημαίνει ότι δεν μπορούμε να κάνουμε σύσταση για νέους χρήστες ή νέες επιχειρήσης. Μια καλή προσέγγιση είναι να συνδυάσετε τη μέθοδο Funk-SVD με μια λιγότερο προηγμένη μέθοδο, όπως είναι η μέθοδος με βάσει κατάταξη (ranked-based) ή με βάση το περιεχόμενο (content-based), κάτι που θα μπορούσε μελλοντικά να υλοποιηθεί με την χρήση του μοντέλου RecGraph.

Επίσης, η υλοποίηση του αλγορίθμου με την χρήση του neo4j σε μεγάλα σύνολα δεδομένων, καθιστά τον αλγόριθμο μη αποδοτικό. Μια εναλλακτική θα μπορούσε να ήταν να παίρνει τα δεδομένα από το γράφο, μέσω ερωτήσεων RecGraph, και να μεταφέρουμε σε ένα πρόγραμμα άλγεβρας πινάκων[26]. Κάτι που θα βελτιώνε το χρόνο εκτέλεσης κατά πολύ και θα επίλυε γρήγορα σύνθετες ερωτήσεις σε γράφημα με μεγάλο όγκο δεδομένων.

ΑΝΑΦΟΡΕΣ

[1] Netflix prize, Simon Funk, https://sifter.org/~simon/journal/20061211.html December 2006. Online, accessed on September 2021.
[2] Yelp open dataset, 2021.
[3] Neo4j, 2021.
[4] Cypher, 2021.
[5] Movielens dataset 2021. https://grouplens.org/datasets/movielens/ September 2021. Online, accessed on September 2021.
[6] Marialena Kyriakidi, Yannis E. Ioannidis: “Universality of Path Algebras for Recommendation Problems”, University of Athens, 2020
[7] Marialena Kyriakidi, Georgia Koutrika, Yannis E. Ioannidis: Recommendations as Graph Explorations. RecSys 2020: 289-298
[8] Francesco Ricci, Lior Rokach, and Bracha Shapira. Introduction to recommender systems handbook. In Recommender systems handbook, pages 1–35. Springer, 2011.
[9] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 426–434. ACM, 2008.
[10] Huizhi Liang and Timothy Baldwin. A probabilistic rating autoencoder for personalized recommender systems. In Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 23, 2015, pages 1863–1866, 2015.
[11] Carré Bernard. Graphs and Networks. Clarendon Press, Oxford, UK, 1979.
[12] Robin Burke. Hybrid web recommender systems. In The adaptive web, pages 377- 408. Springer, 2007.
[13] Rakesh Agrawal, Shaul Dar, and HV Jagadish. Direct transitive closure algorithms: Design and performance evaluation. ACM Transactions on Database Systems (TODS), 15(3):427–458, 1990.
[14] Simon Funk. Try this at home, 2006.
[15] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 426–434. ACM, 2008.
[16] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. Recommender systems with social regularization. In Proceedings of the fourth ACM international conference on Web search and data mining, pages 287–296, 2011.
[17] Yang Bo, Lei Yu, Liu Dayou, and L Jiming. Social collaborative filtering by trust. In International Joint Conference on Artificial Intelligence AAAI Press, pages 2747–2753, 2013.
[18] Guibing Guo, Jie Zhang, and Neil YorkeSmith. Trustsvd: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In *TwentyNinth AAAI Conference on Artificial Intelligence*, 2015.
[19] Yufei Wen, Lei Guo, Zhumin Chen, and Jun Ma. Network embedding based recommendation method in social networks. In Companion Proceedings of the The Web Conference 2018, pages 11–12, 2018.
[20] GroupLens, Social Computing Research at the University of Minnesota. https://grouplens.org/ September 2021. Online, accessed on September 2021.
[21] MaDgIK, Management of Data, Information, and Knowledge Group of the Department of Informatics & Telecommunications of the University of Athens and Athena Research, www.madgik.di.uoa.gr, 2021
[22] Charles LA Clarke, Maheedhar Kolla, Gordon V Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. Novelty and diversity in information retrieval evaluation. In Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, pages 659–666, 2008.
[23] Xuan Nhat Lam, Thuc Vu, Trong Duc Le, and Anh Duc Duong. Addressing coldstart problem in recommendation systems. In Proceedings of the 2nd international conference on Ubiquitous information management and communication, pages 208–211, 2008.
[24] Data Driven Investor. How Does the Funk Singular Value Decomposition Algorithm work in Recommendation Engines? https://medium.datadriveninvestor.com/how-funk-singular-value-decomposition-algorithm-work-in-recommendation-engines-36f2fbf62cac/. Online, accessed on September 2021.
[25] Towards data science. Matrix Factorization in Recommender Systems by Benjamin Wang https://towardsdatascience.com/matrix-factorization-in-recommender-systems-3d3a18009881/. Online, accessed on September 2021.
[26] Fuad T. Jamour, Ibrahim Abdelaziz, Panos Kalnis:A Demonstration of MAGiQ: Matrix Algebra Approach for Solving RDF Graph Queries. Proc. VLDB Endow. 11(12): 1978-1981 (2018)