Πανεπιστημιο Πατρων
Πολυτεχνικη Σχολη
Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών
και Πληροφορικής

Διπλωματική Εργασία

Μελέτη και Υλοποίηση Αλγορίθμων Συνεργατικής Διήθησης, για την Ευφυή Παραγωγή Συστάσεων υπό Συνθήκες Κρύας Εκκίνησης

Χρυσάνθη Γ. Λαγοδήμου
Α.Μ. 4208

Επιβλέπων: Ιωάννης Γαροφαλάκης, Καθηγητής
Συνεπιβλέπων: Αθανάσιος Νικολακόπουλος, Υποψήφιος Διδάκτωρ

Πάτρα, Νοέμβριος 2015

Περίληψη

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ:
ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ :

Ευχαριστίες

Πάτρα, 13 Νοεμβρίου 2015

Περιεχόμενα

Κεφάλαιο 1

Εισαγωγή

Κεφάλαιο 2

Εισαγωγή στους Αλγορίθμους Συνεργατικής Διήθησης

2.1 Συστήματα Παραγωγής Συστάσεων

Τα συστήματα παραγωγής συστάσεων είναι εργαλεία λογισμικού και τεχνικές που στοχεύουν στο να προτείνουν είδη που ενδιαφέρουν τους χρήστες.[11][1] Προέκυψαν από την ανάγκη διευκόλυνσης των χρηστών να επιλέξουν ανάμεσα από μια πληθώρα διαθέσιμων ειδών.

Με δεδομένα τα σύνολα χρηστών, ειδών και άμεσων ή έμμεσων βαθμολογιών των χρηστών για τα είδη, τα συστήματα παραγωγής συστάσεων προσπαθούν είτε να προβλέψουν τις βαθμολογίες των χρηστών για είδη με τα οποία δεν έχουν αλληλεπιδράσει είτε να προτείνουν σε κάποιο χρήστη μία λίστα ειδών που μπορεί να τον ενδιαφέρουν. [10]

Τα συστήματα παραγωγής συστάσεων διακρίνονται σε έξι κατηγορίες[1]:

1. Βάσει περιεχομένου (Content-based): Υπολογίζουν την ομοιότητα των ειδών βάσει των χαρακτηριστικών τους και προτείνουν στο χρήστη παρόμοια είδη με αυτά για τα οποία έχει δείξει ενδιαφέρον.

2. Συνεργατικής διήθησης (Collaborative filtering): Βασίζονται στην παρατήρηση ότι οι άνθρωποι συχνά επιλέγουν ανάμεσα στα διάφορα είδη βάσει των συστάσεων που παρέχονται από άλλους. Θεωρούν ότι αν ένας χρήστης στο παρελθόν είχε παρόμοιες προτιμήσεις με κάποιους άλλους χρήστες, τότε οι συστάσεις που προέρχονται από αυτούς θα τον ενδιαφέρουν και στο μέλλον. Είναι από τις πιο υλοποιημένες προσεγγίσεις [11] και κρίνεται από τις πιο επιτυχημένες.[10]

[1] Κύρια πηγή του παρόντος κεφαλαίου είναι το [11].

3. Δημογραφικά (Demographic): Προτείνουν με βάση τη χώρα που βρίσκεται ο χρήστης, το φύλο, την ηλικία του και άλλα στοιχεία που γνωρίζουν για αυτόν.

4. Βάσει γνώσης (Knowledge-based): Παρέχουν συστάσεις υπολογίζοντας κατά πόσο τα χαρακτηριστικά των ειδών καλύπτουν τις ανάγκες των χρηστών.

5. Βάσει κοινότητας (Community-based): Οι συστάσεις που παρέχουν σε κάποιο χρήστη βασίζονται στις προτιμήσεις των φίλων του καθώς αποδεικνύεται ότι οι άνθρωποι τείνουν να εμπιστεύονται περισσότερο τις συστάσεις που προέρχονται από φίλους τους από αυτές παρόμοιων αλλά άγνωστων σε αυτούς χρηστών. [12]

6. Υβριδικά (Hybrid): Αποτελούν μίξη των παραπάνω κατηγοριών, με τη μία κατηγορία να προσπαθεί να καλύψει τα μειονεκτήματα της άλλης. Για παράδειγμα, τα συστήματα συνεργατικής διήθησης δε μπορούν να προτείνουν είδη για τα οποία δεν έχουν καθόλου βαθμολογίες. Η μίξη ενός τέτοιου συστήματος με ένα που βασίζεται στο περιεχομένο μπορεί να παράξει συστάσεις και για τέτοια είδη.

2.2 Η Συνεργατική Διήθηση

Τα συστήματα συνεργατικής δίηθησης δεν αντιμετωπίζουν κάποια από τα μειονεκτήματα των συστημάτων που βασίζονται στο περιεχομένο επειδή βασίζονται στις βαθμολογίες των χρηστών. Μπορούν να προτείνουν είδη των οποίων το περιεχομένο δεν είναι γνωστό ή δε μπορεί να ανακτηθεί, αλλά και είδη που ανήκουν σε πολύ διαφορετικές κατηγορίες, αν ο χρήστης έχει δείξει ενδιαφέρον για αυτές. Επιπλέον, βασίζονται στην ποιότητα των ειδών όπως αυτή έχει διαμορφωθεί από τις βαθμολογίες των χρηστών και όχι στο περιεχομένο, το οποίο πολλές φορές δεν αποτελεί κριτήριο για την ποιότητα ενός είδους.[11]

Οι κύριες κατηγορίες συστημάτων συνεργατικής διήθησης είναι αυτή των κοντινότερων γειτόνων (nearest-neighbors) και η βασιζόμενη στο μοντέλο (model-based).

2.2.1 Συνεργατική Διήθηση Κοντινότερων Γειτόνων

Η προσέγγιση που βασίζεται στους κοντινότερους γείτονες είναι αρκετά δημοφιλής λόγω της απλότητας, της αποτελεσματικότητας και της ικανότητάς της να παράγει ακριβείς και προσωποιημένες συστάσεις.[11] Τα συστήματα που ακολουθούν αυτή την προσέγγιση χρησιμοποιούν απευθείας τις βαθμολογίες των χρηστών για να προβλέψουν τις βαθμολογίες των νέων για εκείνους ειδών. Η χρήση των βαθμολογιών μπορεί να γίνει είτε με τρόπο που βασίζεται στους χρήστες (user-based) είτε στα είδη (item-based). Τα πρώτα εκτιμούν το ενδιαφέρον του χρήστη

για ένα είδος χρησιμοποιώντας τις βαθμολογίες άλλων χρηστών με τους οποίους έχει βαθμολογήσει με παρόμοιο τρόπο τα κοινά τους είδη (γείτονες). Στα βασιζόμενα στα είδη, η πρόβλεψη για τη βαθμολογία ενός χρήστη γίνεται με βάση το πώς έχει βαθμολογήσει παρόμοια με το υπό εξέταση είδος. Δύο είδη θεωρούνται παρόμοια αν πολλοί χρήστες τα έχουν βαθμολογήσει με παρόμοιο τρόπο.

Τα συστήματα αυτά υστερούν όσον αφορά την πρόβλεψη βαθμολογιών σε σχέση με τους καλύτερους αλγορίθμους που βασίζονται στο μοντέλο.[7, 13] Όμως έχει γίνει πλέον σαφές ότι η ακρίβεια των προβλέψεων δεν είναι ο μόνος παράγοντας που διασφαλίζει την αποτελεσματικότητα του συστήματος. Επιτυχημένη πρόβλεψη δεν είναι μόνο αυτή που προτείνει στο χρήστη απλά ένα νέο είδος, αλλά και αυτή (και αυτό είναι το πιο δύσκολο) που του δίνει την ευκαιρία να ανακαλύψει είδη ή κατηγορίες που μόνος του δε θα το έκανε.[4] Δεδομένου ότι αυτά τα συστήματα αξιοποιούν τις σχέσεις μεταξύ των ειδών, είναι πιο πιθανό να προτείνουν σε κάποιο χρήστη κάποιο είδος που δεν έχει συνάφεια με αυτά που έχει ήδη δει, αν έχει λάβει καλή βαθμολογία από ένα γείτονά του. Αυτό δεν εγγυάται την επιτυχία της σύστασης, αλλά μπορεί να βοηθήσει τον χρήστη να διευρύνει τους ορίζοντες του με επιθυμητό για εκείνον τρόπο.

Τα κύρια πλεονεκτήματα αυτών των μεθόδων είναι [11]:

- Απλότητα. Είναι πιο εύκολα κατανοητές διαισθητικά και σχετικά πιο απλές στην υλοποίησή τους.
- Δυνατότητα παροχής δικαιολόγησης για τις προβλέψεις που υπολογίζουν. Μπορούν να επιτρέψουν στο χρήστη να καταλάβει πώς προέκυψαν οι συστάσεις προς αυτόν και μπορούν να χρησιμεύσουν σε ένα διαδραστικό σύστημα, όπου οι χρήστες θα μπορούν να επιλέξουν τους γείτονες στους οποίους δίνουν οι ίδιοι σημασία.
- Αποτελεσματικότητα: Δεν απαιτούν κοστοβόρες φάσεις εκπαίδευσης και οι κοντινότεροι γείτονες μπορούν να προϋπολογιστούν για πιο γρήγορες συστάσεις και να αποθηκευτούν με μικρό κόστος στη μνήμη. Αυτό τους επιτρέπει να χρησιμοποιούνται σε εφαρμογές με εκατομμύρια χρήστες και είδη.
- Σταθερότητα που οφείλεται στο ότι επηρεάζονται ελάχιστα από την αύξηση των χρηστών, των ειδών και των βαθμολογιών στο σύστημα. Για την παροχή συστάσεων σε νέους χρήστες δε χρειάζεται να ξαναϋπολογιστούν οι ομοιότητες μεταξύ ειδών. Όταν ένα νέο είδος λάβει κάποιες βαθμολογίες, οι μόνες ομοιότητες που υπολογίζονται είναι αυτές που το αφορούν.

Η επιλογή ανάμεσα στα δύο είδη συνεργατικής διήθησης βασίζεται στα ακόλουθα κριτήρια[11]:

- Ακρίβεια: Είναι σημαντική η αναλογία χρηστών και ειδών στο σύστημα. Είναι προτιμότερο να προκύπτουν λιγότεροι γείτονες για τους οποίους όμως

μπορεί να υπολογιστεί υψηλής εμπιστοσύνης ομοιότητα.

- Αποτελεσματικότητα: Και σε αυτή την περίπτωση έχει σημασία η αναλογία χρηστών και ειδών. Στις περισσότερες περιπτώσεις ο αριθμός των χρηστών υπερβαίνει κατά πολύ αυτό των ειδών και ο υπολογισμός των γειτονικών ειδών είναι προτιμότερος από άποψης απαιτούμενης μνήμης και χρόνου υπολογισμού των ομοιοτήτων. (Ο χρόνος που απαιτείται για την παραγωγή των συστάσεων είναι ο ίδιος.) Σε μεγαλύτερα συστήματα και λαμβάνοντας υπόψη ότι οι χρήστες πρακτικά βαθμολογούν λίγα αντικείμενα, μπορεί να υπάρξει αποδοτική υλοποίηση αν για κάθε χρήστη αποθηκεύονται μόνο οι καλύτερες ομοιότητες. Με τον ίδιο τρόπο δε χρειάζεται να ελέγχονται όλα τα ζεύγη χρηστών ή ειδών.

- Σταθερότητα: Για την παροχή ενός σταθερού συστήματος, παίζει ρόλο η συχνότητα και ο ρυθμός αλλαγής του αριθμού των χρηστών. Ο υπολογισμός ομοιοτήτων με βάσει το είδος είναι προτιμότερος σε συστήματα που ο αριθμός τους αυξάνει πιο αργά σε σχέση με αυτόν των χρηστών και αντίστροφα.

- Δυνατότητα παροχής δικαιολόγησης: Οι βασιζόμενες στα είδη μέθοδοι είναι προτιμότερες στις περιπτώσεις που είναι σημαντικό να παρέχεται στο χρήστη και ο λόγος που το σύστημα του προτείνει ένα είδος εκτός από τις περιπτώσεις που ο χρήστης γνωρίζει τους άλλους χρήστες (κοινωνικά δίκτυα).

- Δυνατότητα παροχής απρόσμενων αλλά καλών συστάσεων: Οι μέθοδοι που εξετάζουν την ομοιότητα μεταξύ των χρηστών είναι προτιμότερες σε αυτή την περίπτωση, ιδιαίτερα όταν οι γειτονιές των χρηστών είναι μικρές.

Το γεγονός ότι σε πραγματικές εφαρμογές οι χρήστες βαθμολογούν λίγα αντικείμενα και το ότι ο υπολογισμός ομοιότητας μεταξύ χρηστών προκύπτει από τις κοινές τους βαθμολογίες οδηγεί την προσέγγιση αυτή να αντιμετωπίζει δυο σημαντικές προκλήσεις: τη μειωμένη κάλυψη και την ευαισθησία στα αραιά δεδομένα.

Οι λίγες βαθμολογίες οδηγούν σε λιγότερους γείτονες και έτσι οι μέθοδοι αυτοί δυσκολεύονται να εντοπίσουν ζεύγη χρηστών με παρόμοιες προτιμήσεις, αλλά χωρίς κοινές βαθμολογίες. Αυτό έχει ως αποτέλεσμα λιγότερες συστάσεις, καθώς προτείνονται μόνο είδη που έχουν βαθμολογηθεί από γείτονες, και επομένως μικρότερη κάλυψη του χώρου των ειδών.

Η αραιότητα (sparsity) λόγω των μειωμένων βαθμολογιών είναι πρόβλημα που αντιμετωπίζουν τα περισσότερα συστήματα παραγωγής συστάσεων και στην περίπτωση των συστημάτων που βασίζονται στους γείτονες επηρέαζει την ακρίβεια των συστάσεων. Σε συνδυασμό και με την προσθήκη νέων χρηστών και ειδών στο σύστημα, προκύπτει το πρόβλημα της κρύας εκκίνησης (cold-start)[2], το οποίο μπορεί να οδηγήσει σε έλλειψη δικαιοσύνης στο σύστημα.

[2]Στο πρόβλημα αυτό θα γίνει αναφορά στην επόμενη ενότητα.

Για την επίλυση των παραπάνω προβλημάτων υπάρχουν δύο αρκετά δημοφιλείς προσεγγίσεις. Η πρώτη βασίζεται στη μείωση της διάστασης των αναπαραστάσεων χρηστών και ειδών, αξιοποιώντας τα πιο σημαντικά χαρακτηριστικά τους. Με αυτόν τον τρόπο μπορεί να ανακαλύψει συσχετίσεις ανάμεσα σε χρήστες και είδη, ακόμα και αν εχουν βαθμολογήσει διαφορετικά είδη ή έχουν βαθμολογηθεί από διαφορετικούς χρήστες. Η δεύτερη εφαρμόζει μεθόδους από τη θεωρία γραφημάτων και μέσω αυτών εντοπίζει μεταβατικές σχέσεις ανάμεσα σε χρήστες και δεδομένα. Έχει επιπλέον τη δυνατότητα να προτείνει και μη αναμενόμενες, αλλά καλές, συστάσεις.

2.2.2 Συνεργατική Διήθηση Βασιζόμενη Στο Μοντέλο

Τα συστήματα αυτής της κατηγορίας χρησιμοποιούν τις βαθμολογίες για να εκπαιδεύσουν ένα μοντέλο πρόβλεψης. Στόχος τους είναι να ανακαλύψουν τα λανθάνοντα χαρακτηριστικά χρηστών και ειδών που βρίσκονται πίσω από τις βαθμολογίες. Σε αυτή την κατηγορία ανήκουν δημοφιλείς μέθοδοι που περιλαμβάνουν μοντέλα που προκύπτουν απο την παραγοντοποίηση του μητρώου βαθμολογιών χρηστών-ειδών. Είναι γνωστές ως μέθοδοι που βασίζονται στον αλγόριθμο SVD (SVD-based) και διακρίνονται για την ακρίβεια και την ικανότητά τους να ανταποκρίνονται καλά στην αύξηση των αριθμών χρηστών, ειδών και βαθμολογιών. Επιπλέον, προσφέρουν ένα μοντέλο που μπορεί να αποθηκευτεί αποδοτικά και μπορεί να εκπαιδευτεί σχετικά εύκολα.

Στη βασική τους μορφή, μοντελοποιούν τις αλληλεπιδράσεις χρηστών-ειδών μετασχηματίζοντας χρήστες και είδη στον ίδιο χώρο λανθάνοντων παραγόντων (latent factors). Κάθε χρήστης και κάθε είδος συσχετίζονται με ένα διάνυσμα των οποίων τα στοιχεία καταγράφουν κατά πόσο ο χρήστης ενδιαφέρεται για τους διάφορους παράγοντες και κατά πόσο το είδος τους διαθέτει. Το εσωτερικό τους γινόμενο απεικονίζει το ενδιαφέρον του χρήστη για τα χαρακτηριστικά του είδους. Στη συνέχεια το μοντέλο εκπαιδεύεται, χρησιμοποιώντας τα διαθέσιμα δεδομένα, για να προβλέψει τις βαθμολογίες χρηστών για είδη που δεν έχουν δει. Τα συστήματα αυτά διευρύνουν την κατηγοριοποίηση των ειδών καθώς είναι σε θέση να εντοπίσουν το ενδιαφέρον ενός χρήστη για είδη που έχουν χαρακτηριστικά που εξάγονται αυτόματα από το σύστημα και επομένως να προτείνει πιο στοχευμένα σε αυτόν είδη. Αλγόριθμοι αυτού του είδους, όπως ο SVD++ [7] μπορούν να χρησιμοποιήσουν και άλλα στοιχεία του ιστορικού του χρήστη πέρα από τις βαθμολογίες, αυξάνοντας την ακρίβεια των προβλέψεων. Μέθοδοι αυτής της κατηγορίας μπορούν ακόμα να λάβουν υπόψη τους τις αλλαγές στις προτιμήσεις των χρηστών και στα χαρακτηριστικά των ειδών στη διάρκεια του χρόνου, βελτιώνοντας και άλλο την ποιότητα των προβλέψεων. [11]

2.3 Το πρόβλημα της Κρύας Εκκίνησης

Το πρόβλημα της κρύας εκκίνησης αναφέρεται στην συμπεριφορά του συστήματος στην εισαγωγή νέων χρηστών και ειδών και στη δυσκολία παραγωγής αξιόπιστων συστάσεων λόγω της αρχικής έλλειψης βαθμολογιών. Είναι πρόβλημα που αντιμετωπίζουν όλα τα συστήματα παραγωγής συστάσεων, τόσο αυτά που βασίζονται στο περιεχομένο όσο και αυτά που εφαρμόζουν συνεργατική διήθηση. Για τα δεύτερα είναι πιο σοβαρό πρόβλημα καθώς βασίζονται αποκλειστικά στις βαθμολογίες των χρηστών. Μπορεί να θεωρηθεί και ως πρόβλημα μειωμένης κάλυψης του χώρου των ειδών.[11] Διακρίνεται κυρίως σε τρεις κατηγορίες [10]:

- Πρόβλημα Νέας Κοινότητας (New Community Problem): Στην αρχή της λειτουργίας ενός συστήματος, οι βαθμολογίες είναι αναγκαστικά λίγες. Αυτό οδηγεί αραιά σύνολα δεδομένων και δεν επιτρέπει στα συστήματα συνεργατικής διήθησης να εντοπίσουν τις απαραίτητες συσχετίσεις ανάμεσα σε χρήστες και είδη.

- Προβλημα Νέων Χρηστών (New Users Problem): Το πρόβλημα αυτό εμφανίζεται με την εισαγωγή νέων χρηστών στο σύστημα, οι οποίοι δε μπορούν να λάβουν προσωποιημένες συστάσεις, αφού έχουν ελάχιστες βαθμολογίες.

- Πρόβλημα Νέων Ειδών (New Items Problem): Επειδή τα νέα είδη που εισάγονται στο σύστημα εκ των πραγμάτων δεν έχουν βαθμολογηθεί αρκετά, δε μπορούν να συσχετιστούν με τα υπόλοιπα και επομένως να συμπεριληφθούν στις συστάσεις προς τους χρήστες.

Κεφάλαιο 3

Ο Αλγόριθμος HIR

Η αραιότητα (sparsity) είναι εγγενές χαρακτηριστικό των συστημάτων παραγωγής συστάσεων και αποτελεί ένα από τα πιο δύσκολα ζητήματα που έχουν να αντιμετωπίσουν οι αλγόριθμοι συνεργατικής δίηθησης. Σε αυτό ακριβώς το πρόβλημα, ακόμα και στην ακραία του μορφή που είναι το πρόβλημα της κρύας εκκίνησης, απαντάει ο Hierarchical Itemspace Rank (HIR). Ο HIR εκμεταλλεύεται την ιεραρχική δομή του χώρου των ειδών, ώστε να ξεπεράσει τους περιορισμούς στην ποιοτική παραγωγή συστάσεων[10].[1]

3.1 Εισαγωγή

Το έναυσμα για τον αλγόριθμο HIR προέκυψε από τα αποτελέσματα ενός καινοτόμου πλαισίου εργασίας στον Παγκόσμιο Ιστό [9], το οποίο κατέδειξε τα οφέλη της απεικόνισης των έμμεσων σχέσεων μεταξύ των ιστοσελίδων που προκύπτουν από την εγγενή ιεραρχική δομή του Παγκόσμιου Ιστού και της εκμετάλλευσής τους για την καλύτερη κατάταξη των ιστοσελίδων που επιστρέφονται από μία μηχανή αναζήτησης.

Ο αλγόριθμος HIR εκμεταλλεύεται την ιεραρχική δομή που υπάρχει εκ φύσεως στο χώρο των ειδών, ώστε να χαρακτηρίσει την σχέσεις μεταξύ των ειδών σε ένα μακροσκοποπικό επίπεδο. Για το σκοπό αυτό αναλύει το χώρο των ειδών σε σύνολα στενά συνδεδεμένων στοιχείων και χρησιμοποιεί αυτή την ανάλυση για να εκμεταλλευτεί τις έμμεσες σχέσεις που προκύπτουν ανάμεσα στα είδη.

Κεντρική του ιδέα είναι ο συνδυασμός των άμεσων σχέσεων που προκύπτουν από τις αλληλεπιδράσεις των χρηστών με τα είδη και τις έμμεσες που προκύπτουν από τις σχέσεις μεταξύ των ειδών με στόχο την αντιμετώπιση του προβλήματος της αραιότητας και τη βελτίωση της ποιότητας των συστάσεων.

[1] Κύρια πηγή του παρόντος κεφαλαίου ειναι η δημοσίευση [10]

3.2 Το πλαίσιο εργασίας

Στην ενότητα αυτή παρουσιάζεται το μοντέλο εργασίας του αλγορίθμου.

3.2.1 Σημειογραφία

- Όλα τα διανύσματα-στήλες αναπαριστώνται με έντονα πεζά γράμματα.
- Όλα τα μητρώα με έντονα κεφαλαία γράμματα.
- $\mathbf{c_j^T}$ είναι η j στήλη του μητρώου $\mathbf{C}$ και C_{ij} είναι το στοιχείο στη γραμμή i και στη στήλη j του μητρώου $\mathbf{C}$.
- Με κεφαλαία καλλιγραφικά γράμματα ορίζονται τα σύνολα και
- το $\triangleq$ χρησιμοποιείται στους ορισμούς.

3.2.2 Ορισμός του μοντέλου

Ορίζονται τα σύνολα:

- Χρηστών: $\mathcal{U} = \{u_1, u_2, \cdots, u_n\}$
- Ειδών: $\mathcal{V} = \{v_1, v_2, \cdots, v_m\}$
- Βαθμολογιών: $\mathcal{R}$, το οποίο αποτελείται από πλειάδες $t_{ij} = (u_i, v_j, r_{ij})$, όπου r_{ij} είναι η βαθμολογία του χρήστη u_i για το είδος v_j, η οποία μπορεί να αντιστοιχεί είτε σε ένα θετικό αριθμό είτε σε μια δυαδική μορφή που θα παρουσιάζει την αλληλεπίδραση ή μη του χρήστη με ένα είδος.
- Διαμέρισης του συνόλου $\mathcal{R}$: $\{\mathcal{L}, \mathcal{T}\}$, με το $\mathcal{L}$ να αποτελεί το σύνολο εκπαίδευσης και το $\mathcal{T}$ συνολο ελέγχου. Το $\mathcal{L}_i$ περιέχει τα είδη που έχει βαθμολογήσει ο u_i και ανήκουν στο σύνολο εκπαίδευσης. Το $\mathcal{T}_i$ περιέχει τα είδη που έχει βαθμολογήσει ο u_i και ανήκουν στο σύνολο ελέγχου: $\mathcal{L}_i \triangleq \{v_k : t_{ik} \in \mathcal{L}\}$, $\mathcal{T}_i \triangleq \{v_l : t_{il} \in \mathcal{T}\}$.

Κάθε χρήστης u_i συσχετίζεται με το διάνυσμα προτίμησής του, $\boldsymbol{\omega}^i$, του οποίου τα μη μηδενικά στοιχεία αντιστοιχούν στις βαθμολογίες που έχει δώσει ο χρήστης και ανήκουν στο $\mathcal{L}$ και για τον οποίο ισχύει $\mathcal{L}_i \neq \emptyset$. Το διάνυσμα προτίμησης είναι κανονικοποιημένο, ώστε τα στοιχεία του να αθροίζουν στη μονάδα.

$$\boldsymbol{\omega}^i = [\omega_1^i, \omega_2^i, \cdots, \omega_m^i]$$

Στόχος της μεθόδου που προτείνει ο αλγόριθμος HIR είναι η αντιστοίχιση του σε μία προσωποποιημένη κατανομή πάνω στο χώρο των ειδών.

Επιπλέον ορίζεται μια οικογένεια μη κενών συνόλων πάνω στο χώρο ειδών $\mathcal{V}$ με βάση κάποιο κριτήριο κατηγοριοποίησης των ειδών:

$$\mathcal{D} \triangleq \{\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_K\}, \text{ με } \mathcal{V} = \bigcup_{k=1}^{K} \mathcal{D}_k.$$

Τέλος, ορίζεται το $\mathcal{G}_v$, το οποίο αποτελεί την ένωση των συνόλων που περιέχουν το v και N_v το πλήθος αυτών:

$$\mathcal{G}_v \triangleq \bigcup_{v \in \mathcal{D}_k} \mathcal{D}_k.$$

Μητρώο Άμεσης Συσχέτισης $\mathbf{C}$

Το μητρώο άμεσης συσχέτισης $\mathbf{C}$ στοχεύει στο να απεικονίσει τις άμεσες σχέσεις που προκύπτουν ανάμεσα στα στοιχεία του συνόλου των ειδών $\mathcal{V}$. Κάθε στοιχείο του $\mathcal{V}$ συσχετίζεται με μία διακριτή κατανομή πάνω σε αυτό, η οποία ποσοτικοποιεί τις ομοιότητες των στοιχείων του.

$$\mathbf{c_v^T} = [c_1, c_2, \cdots, c_m]$$

Για να οριστεί το μητρώο $\mathbf{C}$, χρειάζεται να οριστεί η οικογένεια συνόλων $\mathcal{U}_{ij} \subseteq \mathcal{U}$, που αντιπροσωπεύει το σύνολο των χρηστών που έχουν βαθμολογήσει και το v_i και το v_j.

$$\mathcal{U}_{ij} \triangleq \begin{cases} u_k : (v_i \in \mathcal{L}_k) \wedge (v_j \in \mathcal{L}_k) & i \neq j \\ \emptyset & i = j \end{cases}$$

Ορίζεται τώρα το μητρώο $\mathbf{Q}$, με $\mathbf{Q}_{ij} \triangleq |\mathcal{U}_{ij}|$. Το μητρώο αυτό είναι συμμετρικό και η διαγώνιός του ειναι μηδενική. Η στοχαστική έκδοση $\hat{\mathbf{Q}}$ αντιστοιχεί στο γράφημα συσχέτισης των ειδών, όπως ορίζεται στο [5], όπου κάθε ακμή του γραφήματος έχει ως βάρος το αντίστοιχο στοιχείο του μητρώου. Το μητρώο αυτό σε αντίθεση με το $\mathbf{Q}$ δεν είναι συμμετρικό. Το μητρώο $\mathbf{C}$ ορίζεται ως:

$$\mathbf{C} \triangleq \hat{Q} + \tfrac{1}{m}\mathbf{a}\mathbf{e}^T$$

όπου το διάνυσμα $\mathbf{a} \in \Re^m$ έχει μονάδες στα στοιχεία που αντιστοιχούν στις μηδενικές γραμμές του $\mathbf{Q}$ και το $\mathbf{e}^T \in \Re^m$ είναι το μοναδιαίο διάνυσμα. Το μητρώο που προστίθεται στο $\hat{\mathbf{Q}}$ στοχεύει στο να αντικαταστήσει τις μηδενικές γραμμές του με μια ομοιόμορφη κατανομή στο χώρο των ειδών. Αυτή η μετατροπή συμβάλλει στο να προκύψει κάποια πληροφορία για ζεύγη ειδών όπου τα σύνολα των χρηστών που τις έχουν βαθμολογήσει είναι ξένα μεταξύ τους. Τέτοια σύνολα είναι πιο πιθανό να βρεθούν σε συστήματα συστάσεων που αντιμετωπίσουν το πρόβλημα της κρύας εκκίνησης.

Μητρώο Έμμεσης Συσχέτισης $\mathbf{D}$

Το μητρώο $\mathbf{D}$ στοχεύει στο να απεικονίσει τις έμμεσες σχέσεις ανάμεσα στο χώρο των ειδών όπως προκύπτουν από την ιεραρχική του δομή. Αυτή η δομή προκύπτει από το γεγονός ότι η έκφραση προτίμησης ενός χρήστη για ένα είδος

υποδηλώνει το ενδιαφέρον του για τις κατηγορίες στις οποίες ανήκει και κατέπέκταση και για τα είδη που ανήκουν σε αυτές. Η απεικόνιση αυτών των σχέσεων μπορεί να φανεί πολύ χρήσιμη για την αντιμετώπιση του προβλήματος της αραιότητας.

Με βάση τα παραπάνω, κάθε γραμμή του $\mathbf{D}$ συσχετίζεται με ένα διάνυσμα πιθανότητας $\mathbf{d}_v^T$, το οποίο κατανέμει ομοιόμορφα τη μάζα του στα N_v διαφορετικά σύνολα του $\mathcal{D}$ που αποτελούν το $\mathbf{G}_v$ και στη συνέχεια στα είδη που αποτελούν το καθέ ένα από αυτά. Κάθε στοιχείο του $\mathbf{D}$, ορίζεται ως:

$$\mathbf{D}_{ij} \triangleq \sum_{\mathbf{D}_k \in \mathcal{G}_{v_i}, v_j \in \mathbf{D}_k} \frac{1}{N_{v_i}|\mathcal{D}_k|}$$

Τόσο από τον ορισμό του, όσο και από τον ορισμό της οικογένειας συνόλων $\mathbf{D}$, προκύπτει ότι το $\mathbf{D}$ είναι στοχαστικό κατά γραμμές.

3.2.3 Ο Αλγόριθμος Ιεραρχικής Κατάταξης βάσει του χώρου των ειδών

Με βάση τα παραπάνω, μπορεί να οριστεί το προσωποποιημένο διάνυσμα κατάταξης του HIR, ως η κατανομή πιθανότητας πάνω στο χώρο των ειδών, όπως παράγεται από τον αλγόριθμο 1:

Αλγόριθμος 1 Ιεραρχική Κατάταξη Βάσει του Χώρου των Ειδών

Είσοδος: Μητρώα $\mathbf{C}$, $\mathbf{D} \in \Re^{m \times m}$, παράμετροι α, $\beta : \alpha, \beta > 0$, $\alpha + \beta < 1$ και το προσωποποιημένο διάνυσμα προτίμησης $\boldsymbol{\omega} \in \Re^m$

Έξοδος: Το διάνυσμα κατάταξης για τον χρήστη, $\boldsymbol{\pi} \in \Re^m$

1: $\boldsymbol{\pi}^T \leftarrow (1 - \alpha - \beta)\boldsymbol{\omega}^T$
2: **for all** $\omega_j \neq 0$ **do**
3: $\quad \boldsymbol{\pi}^T \leftarrow \boldsymbol{\pi}^T + \omega_j(\alpha \mathbf{c}_j^T + \beta \mathbf{d}_j^T)$
4: **end for**
5: **return** $\boldsymbol{\pi}$

Θεώρημα 1. *Για κάθε διάνυσμα προτίμησης $\boldsymbol{\omega}$, το προσωποποιημένο διανυσμα $\boldsymbol{\pi}$, το οποίο παράγεται από του αλγόριθμο HIR είναι ένα καλώς ορισμένο κανονικοποιημένο διάνυσμα συστάσεων, το οποίο απεικονίζει μία κατανομή πιθανότητας στο σύνολο των ειδών $\mathcal{V}$.*

Απόδειξη. Για κάθε διάνυσμα προτίμησης $\boldsymbol{\omega}$ (το οποίο είναι από τον ορισμό του μη αρνητικό) και για $\alpha, \beta > 0$, $\alpha + \beta < 1$, το $\boldsymbol{\pi}$ είναι μη αρνητικό διάνυσμα. Επομένως, αρκεί να δειχθεί ότι $\boldsymbol{\pi}^{\mathrm{T}}\mathbf{e} = 1$:

$$\boldsymbol{\pi}^{\mathrm{T}}\mathbf{e} = \{(1 - \alpha - \beta)\boldsymbol{\omega}^{\mathrm{T}} + \alpha \sum_{j:\boldsymbol{\omega}_j \neq 0} \boldsymbol{\omega}_j \mathbf{c}_j^{\mathrm{T}} + \beta \sum_{j:\boldsymbol{\omega}_j \neq 0} \boldsymbol{\omega}_j \mathbf{d}_j^{\mathrm{T}}\}\mathbf{e}$$

Δεδομένου ότι τα στοιχεία του διανύσματος $\boldsymbol{\omega}$ είναι κανονικοποιημένα να αθροίζουν στη μονάδα και τα μητρώα $\mathbf{C}$ και $\mathbf{D}$ είναι στοχαστικά κατά γραμμές, προκύπτει ότι:

$$\boldsymbol{\pi}^{\mathrm{T}}\mathbf{e} = (1-\alpha-\beta) + \alpha \sum_{j:\boldsymbol{\omega}_j \neq 0} \boldsymbol{\omega}_j + \beta \sum_{j:\boldsymbol{\omega}_j \neq 0} \boldsymbol{\omega}_j = (1-\alpha-\beta) + \alpha + \beta = 1$$

□

3.2.4 Ζητήματα Απαιτούμενης Μνήμης

Το μητρώο $\mathbf{C}$ είναι εγγενώς αραιό και κλιμακώνεται πολύ καλά με την αύξηση του αριθμού των χρηστών. Η αύξηση του αριθμού των χρηστών οδηγεί μόνο σε αύξηση του αριθμού των μη μηδενικών στοιχείων του, αφού η διάστασή του εξαρτάται μόνο απο το πλήθος των ειδών. Σε πραγματικές εφαρμογές αυτό το πλήθος αυξάνει πολυ αργά. [5]

Το μητρώο $\mathbf{D}$ είναι από τον ορισμό του χαμηλής τάξης για $K < m$. Το $\mathbf{D}$ μπορεί να παραγοντοποιηθεί και να επιτευχθεί έτσι αποδοτικότερη αποθήκευση του και μείωση του υπολογιστικού κοστους. Για το σκοπό αυτό, ορίζεται το μητρώο $\mathbf{A} \in \Re^{m \times K}$, ώστε:

$$\mathcal{A}_{ik} \triangleq \begin{cases} 1 & \text{αν } v_i \in \mathcal{D}_k \\ 0 & \text{αλλιώς} \end{cases}$$

Το $\mathbf{D}$ μπορεί να γραφεί ως το γινόμενο των στοχαστικών κατά γραμμές εκδόσεων του $\mathbf{A}$ και του αναστρόφου του, αντίστοιχα.

$$\mathbf{D} = \mathbf{XY},\ \mathbf{X} \in \Re^{m \times K},\ \mathbf{Y} \in \Re^{K \times m},\ \text{με}$$
$$\mathbf{X} \triangleq \mathbf{S}^{-1}\mathbf{A} \text{ και}$$
$$\mathbf{Y} \triangleq \mathbf{T}^{-1}\mathbf{A}^{\mathrm{T}}$$

όπου τα μητρώα $\mathbf{S} \triangleq \mathrm{diag}(\mathbf{Ae})$ και $\mathbf{T} \triangleq \mathrm{diag}(\mathbf{A}^{\mathrm{T}}\mathbf{e})$ είναι διαγώνια.

Λήμμα 1. *Τα μητρώα $\mathbf{X}$ και $\mathbf{Y}$ είναι καλώς ορισμένα για οποιαδήποτε παραγοντοποίηση του $\mathbf{D}$ ικανοποιώντας τον ορισμό του.*

Απόδειξη. Αρκεί να δειχθεί ότι τα $\mathbf{S}$ και $\mathbf{T}$ είναι αντιστρέψιμα για οποιοδήποτε μητρώο $\mathbf{A}$. Από τον ορισμό του $\mathbf{D}$ προκύπτει ότι κάθε γραμμή και κάθε στήλη του $\mathbf{A}$ αντιστοιχεί σε ένα μη μηδενικό διάνυσμα στον $\Re^K$ και $\Re^m$ αντίστοιχα.

Η ύπαρξη μηδενικής γραμμής στο μητρώο $\mathbf{A}$, θα σήμαινε ότι το σύνολο $\mathcal{V}$ δεν αποτελεί την ένωση της οικογένειας συνόλων $\mathcal{D}_k$, άτοπο από τον ορισμό του $\mathcal{D}$.

Η ύπαρξη μηδενικής στήλης στο μητρώο $\mathbf{A}$, θα προέκυπτε μόνο αν κάποιο από τα $\mathcal{D}_k$ ήταν κενό, το οποίο αντιτίθεται στον ορισμό τους και δε θα είχε νόημα βάσει του ορισμού του μοντέλου.

Τα παραπάνω οδηγούν στο συμπέρασμα ότι τα διανύσματα $\mathbf{Ae}$, $\mathbf{A}^{\mathrm{T}}\mathbf{e}$ είναι αυστηρά θετικά, πράγμα που διασφαλίζει την ανιστρεψιμότητα των μητρώων $\mathbf{S}$, $\mathbf{T}$. □

3.2.5 Υπολογιστικά Ζητήματα

Το κόστος εκτέλεσης του αλγορίθμου 1 είναι μικρό καθώς η δομή επανάληψης απαιτεί $\mathcal{O}(|\mathcal{V}|)$ πράξεις και εκτελείται $\frac{|\mathcal{L}_i|}{m}$ φορές καθώς οι χρήστες αλληλεπιδρούν με ένα πολύ μικρό μέρος των διαθέσιμων ειδών.

Η παραγοντοποίηση του $\mathbf{D}$ ως γινόμενο δύο εξαιρετικά αραιών μητρώων, καταργώντας την ανάγκη απευθείας υπολογισμού του, καθώς και το χαμηλής πυκνότητας μητρώο βαθμολογιών, μειώνουν και άλλο το υπολογιστικό κόστος καθώς επιτρέπουν τον μαζικό υπολογισμό των προσωποποιημένων διανυσμάτων κατάταξης που υπολογίζει ο αλγόριθμος 1.

Για τον υπολογισμό αυτό χρειάζεται να οριστεί ένα μητρώο $\mathbf{\Omega} \in \Re^{n \times m}$, τέτοιο ώστε:

$$\mathbf{\Omega} \triangleq \begin{bmatrix} (\boldsymbol{\omega^1})^{\mathrm{T}} \\ (\boldsymbol{\omega^2})^{\mathrm{T}} \\ \vdots \\ (\boldsymbol{\omega^n})^{\mathrm{T}} \end{bmatrix}$$

Ο μαζικός αυτός υπολογισμός του HIR υπολογίζεται με τον Αλγόριθμο 2, με τις γραμμές του μητρώου $\mathbf{\Pi} \in \Re^{n \times m}$, να περιέχουν τα διανύσματα συστάσεων για κάθε χρήστη του συνόλου $\mathcal{U}$.

Αλγόριθμος 2 Μαζικός Υπολογισμός του HIR

Είσοδος: Μητρώα $\mathbf{C}$, $\mathbf{X}$, $\mathbf{Y}$, παράμετροι α, β : $\alpha, \beta > 0$, $\alpha + \beta < 1$ και το προσωποποιημένο μητρώο προτίμησης $\mathbf{\Omega}$
Έξοδος: Το μητρώο κατάταξης για τους χρήστες, $\mathbf{\Pi} \in \Re^{n \times m}$

1: $\mathbf{\Pi} \leftarrow (1 - \alpha - \beta)\mathbf{\Omega}$
2: $\mathbf{\Pi} \leftarrow \mathbf{\Pi} + \alpha(\mathbf{\Omega C}) + \beta((\mathbf{\Omega X})\mathbf{Y})$
3: **return** $\mathbf{\Pi}$

3.3 Πειραματική Αξιολόγηση

Η πειραματική αξιολόγηση[10] του HIR, έγινε στους τομείς των συστάσεων ταινιών και μουσικής.

Για τον πρώτο τομέα, τα πειράματα έγιναν στα σύνολα δεδομένων MovieLens100K και MovieLens1M, με το διαχωρισμό των ταινιών σε κατηγορίες να αποτελεί το κριτήριο για τον ορισμό του μητρώου έμμεσης συσχέτισης. Τα σύνολα αυτά χρησιμοποιούνται ευρέως για τον έλεγχο απόδοσης των συστημάτων παραγωγής συστάσεων. Τα σύνολα αυτά περιλαμβάνουν:

Σύνολα Δεδομένων	Χρήστες	Είδη	Βαθμολογίες
MovieLens100K	943	1682	100.000
MovieLens1M	6040	3883	1.000.209

Για το δεύτερο τομέα, τα πειράματα έγιναν στο σύβολο δεδομένων Yahoo!R2Music με κριτήριο κατηγοριοποίησης τους καλλιτέχνες που ερμηνεύουν τα τραγούδια. Το σύνολο αυτό περιλαμβάνει:

Χρήστες	**Είδη**	**Βαθμολογίες**
1.823.179	136.736	717.872.016

Η σύγκριση του HIR έγινε με τους ακόλουθους αλγορίθμους παραγωγής συστάσεων που βασίζονται σε κατατάξεις:

- L και Katz, οι οποίοι βασίζονται σε ομοιότητα κόμβων
- First Passage Time (FP) και Matrix Forest Algorithm (FMA) που ακολουθούν την προσέγγιση του τυχαίου περιπάτου
- ItemRank [5] και
- PureSVD

Οι δύο τελευταίοι δε μπορούν να επωφεληθούν από την ιεραρχική δομή του χώρου των ειδών. Η υλοποίηση όλων των αλγορίθμων έγινε σε MATLAB.

Από τους παραπάνω αλγορίθμους μόνο ο ItemRank και ο HIR εξαρτώνται αποκλειστικά από το μέγεθος του χώρου των ειδών, του οποίου η διάσταση αυξάνεται πολύ αργά σε πραγματικές εφαρμογές. Σε όλα τα παραπάνω σύνολα δεδομένων ο HIR ήταν 10-15 φορές πιο γρήγορος.

Για τα πειράματα χρησιμοποιήθηκαν οι μετρικές:

- Spearman's ρ
- Kendall's τ
- Degree of Agreement (DOA)
- Normalized Distance-based Performance Measure

Η σύγκριση των αλγορίθμων έγινε πάνω στα προβλήματα της Νέας Κοινότητας, Νέων Χρηστών και Νέων Ειδών που αποτελούν τις τρεις εκφάνσεις του προβλήματος της κρύας εκκίνησης.

Για τη μοντελοποίηση του προβλήματος της Νέας Κοινότητας, φτιάχτηκαν τρία τεχνητά σύνολα δεδομένων που περιείχαν το 10%, 20% και 30% του συνόλου των βαθμολογιών, τα οποία θεωρήθηκε ότι αντιπροσωπεύουν τις αρχικές φάσεις του συστήματος.

Στα πειράματα που έγιναν, ο HIR ανταποκρίθηκε πολύ καλά και αποδείχθηκε ότι παρά το γεγονός ότι οι άμεσες σχέσεις είχαν ελάχιστη συνεισφορά στις αρχές του συστήματος, οι έμμεσες σχέσεις που απεικονίζονται στο μητρώο $\mathbf{D}$ διατηρούνται περισσότερο. Αυτό είχε ως αποτέλεσμα, το διάνυσμα συστάσεων που επιστρέφει ο HIR να αποδεικνύεται λιγότερο ευαίσθητο στο πρόβλημα αυτό.

Για την προσομοίωση του δεύτερου προβλήματος, έγινε τυχαία επιλογή 200 χρηστών που είχαν βαθμολογήσει 100 ή περισσότερα είδη και τυχαία διαγραφή του 96%, 94% και 92% των βαθμολογιών τους, ώστε να κατασκευαστούν τρία σύνολα δεδομένων που να αφορουν νεοεισερχόμενους στο σύστημα χρήστες.

Από τις δοκιμές που διεξήχθησαν, για τις διάφορες τιμές του β (της μεταβλητής που καθορίζει τη συμβολή του μητρώου έμμεσης συσχέτισης), προέκυψε η θετική συνεισφορά του μητρώου $\mathbf{D}$ στην ποιότητα της κατάταξης των συστάσεων ακόμα και για μικρές τιμές του β, αλλά και για την περίπτωση που υπήρχε διαθέσιμο μόνο το 4% των βαθμολογιών των υπό δοκιμή χρηστών.

Το αποτέλεσμα αυτό ήταν αναμενόμενο λόγω της θεώρησης του αλγορίθμου HIR καθώς παρά το ότι δεν είναι γνωστές αρκετές προτιμήσεις των χρηστών, η εκμετάλλευση της έμμεσης πληροφορίας που παρέχεται από τις πρώτες βαθμολογίες τους, δίνει στον HIR συγκριτικό πλεονέκτημα στην επιτυχή παραγωγή καλών συστάσεων.

Για τον έλεγχο της περίπτωσης των νέων ειδών, επιλέχθηκαν τυχαία το 10%, 12,5% και 15% των ειδών που είχαν τουλάχιστον 30 βαθμολογόες και αφαιρέθηκε τυχαία το 90% αυτών για τη δημιουργία 3 συνόλων δεδομένων.

Και σε αυτή την περίπτωση η επίδοση του HIR ήταν πολύ καλή. Στο Yahoo!R2Music για τις διάφορες μετρικές κατέκτησε από την 1η ως την 3η θέση και για το MovieLens1M την 1η για όλες τις μετρικές που χρησιμοποιήθηκαν, επιδεικνύοντας σταθερότητα στις κατατάξεις που έδινε και έλλειψη ευαισθησίας στην αραιότητα.

Η καλή του επίδοση οφείλεται στη θεώρησή του για τις έμμεσες σχέσεις που προκύπτουν από τις βαθμολογίες των ειδών, δείχνοντας ότι ακόμα και για ανεπαρκή αριθμό βαθμολογιών, τα νέα είδη αντιμετωπίζονται πιο δίκαια.

Στα πειράματα που έγιναν, στο MovieLens100K με τη χρήση κατάλληλων μετρικών φάνηκε ότι ο αλγόριθμος έδινε τις πιο ποιοτικές κατατάξεις από όλους τους αλγορίθμους με τους οποίους συγκρίθηκε.

3.4 Συμπέρασμα

Συμπερασματικά, ο αλγόριθμος HIR εκμεταλλεύεται αποτελεσματικά τις σχέσεις των ειδών που προκύπτουν από την κατηγοριοποίησή τους και παράγει ποιοτικές κατατάξεις. Η αραιότητα έχει μειωμένη επίδραση στο μοντέλο που προτάθηκε. Επιπλέον, το μοντέλο αυτό μπορεί να υπολογιστεί αποδοτικά τόσο λόγω του ότι εξαρτάται απόκλειστικά από τη διάσταση του χώρου των ειδών όσο και λόγω των

μαθηματικών ιδιοτήτων που διαθέτει. Από την πειραματική αξιολόγηση του αλγορίθμου προέκυψε η καλή συμπεριφορά του απέναντι στο πρόβλημα της κρύας εκκίνησης.

Το μοντέλο αυτό είναι επεκτάσιμο και μπορεί να εφαρμοστεί σε περαιτέρω κατηγοριοποίηση του χώρου των ειδών με απεικόνιση των διαφόρων υποκατηγοριών με την εισαγωγή νέων χαμηλής τάξης μητρώων $\mathbf{D}_1, \mathbf{D}_2, \ldots$ (και αντίστοιχων μεταβλητών $\beta_1, \beta_2, \ldots$) που θα απεικονίζουν με περισσότερη λεπτομέρεια την ιεραρχία του χώρου. Η εισαγωγή τέτοιων μητρώων δεν επηρεάζει τη διάσταση του μοντέλου.

Κεφάλαιο 4

Το LensKit

4.1 Εισαγωγή

Το LensKit αποτελεί ένα ανοιχτού κώδικα πακέτο λογισμικού για την πραγματοποίηση μελέτης και επαληθεύσιμης έρευνας πάνω σε συστήματα παραγωγής συστάσεων. Αποτελεί μία πλατφόρμα για την ανάπτυξη αλγορίθμων παραγωγής συστάσεων, μέτρησης της επίδοσής τους πάνω σε διαφορετικά σύνολα δεδομένων και σύγκρισης νέων ιδεών με τις τρέχουσες βέλτιστες πρακτικές. [2][1] Υλοποιήθηκε από τον Michael D. Ekstrand και αναπτύχθηκε από ερευνητές στο Texas State University και στο πανεπιστήμιο της Minnesota, με συνεισφορά από προγραμματιστές από όλο τον κόσμο.[2] Αποτελείται από περισσότερες από 48K γραμμές κώδικα με συνεισφορά από 27 προγραμματιστές. Είναι γραμμένο κυρίως σε Java (92%) και ένα σημαντικό μέρος του είναι σε Groovy (7%).[3] Ο πηγαίος κώδικας του LensKit είναι δημοσιευμένος στο GitHub.[4]

Για το σκοπό που αναπτύχθηκε προσφέρει:

- Διεπαφές προγραμματισμού εφαρμογών (APIs) για την παραγωγή συστάσεων και προβλέψεων. Επιτρέπει στους προγραμματιστές να χρησιμοποιήσουν τους υλοποιημένους αλγόριθμους ως «μαύρο κουτί».

- Υλοποίηση βασικών αλγορίθμων για παραγωγή συστάσεων και πρόβλεψη βαθμολογιών.

- Εργαλειοθήκη αξιολόγησης απόδοσης σε κοινά σύνολα δεδομένων με χρήση ποικιλίας μετρικών.

[1] Κύρια πηγή του παρόντος κεφαλαίου ειναι το [2].
[2] http://lenskit.org/
[3] Στατιστικά από BlackDuck | Open Hub https://www.openhub.net/p/lenskit
[4] https://github.com/lenskit/lenskit

- Κώδικα υποστήριξης για την ανάπτυξη νέων αλγορίθμων, μεθόδων αξιολόγησης και άλλων επεκτάσεων.

Ένας παραγωγός συστάσεων στο LensKit αποτελείται από ένα σύνολο διεπαφών που παρέχουν παραγωγή συστάσεων και πρόβλεψη βαθμολογιών, οι οποίες συνδέονται με μία πηγή δεδομένων χρησιμοποιώντας έναν ή περισσότερους αλγορίθμους παραγωγής συστάσεων. Η σύγκριση διαφορετικών αλγορίθμων γίνεται με τη χρήση script (σεναρίου) αφού δηλωθούν οι πηγές δεδομένων, οι αλγόριθμοι και οι μετρικές βάσει των οποίων συγκρίνονται.

4.2 Σχεδιασμός του LensKit

Ο σχεδιασμός του LensKit έγινε με στόχο τόσο την άναπτυξη και την έρευνα πάνω σε συστήματα παραγωγής συστάσεων όσο και τη χρήση του σε εκπαιδευτικούς σκοπούς. Το πανεπιστήμιο της Minnesota χρησιμοποιεί το LensKit τόσο σε μεταπτυχιακό του μάθημα όσο και σε MOOC πάνω σε συστήματα παραγωγής συστάσεων που προσέφερε μέσω της πλατφόρμας Coursera. Το LensKit αναπτύχθηκε με τρόπο που να υποστηρίζει τη διεξαγωγή πειραμάτων.[6]

Βασική αρχή στο LensKit είναι η υλοποίηση αλγορίθμων ως ένα σύνολο σχεδόν ανεξάρτητων τμημάτων. Ένας τυπικός αλγόριθμος αλγόριθμος, όπως και η υλοποίηση του αλγορίθμου HIR χρειάζεται τουλάχιστον δώδεκα διακριτά τμήματα (components) που επικοινωνούν μεταξύ τους μέσω καλώς ορισμένων διεπαφών. Αυτή η πρακτική διευκολύνει τη συντήρηση και τον έλεγχο ορθότητας καθενός από τα συστατικά της υλοποίησης.

Βασικός στόχος είναι η διασφάλιση της ορθότητας και στη συνέχεια της αποτελεσματικότητας. Στο LensKit υπάρχουν πολλά αμετάβλητα αντικείμενα (immutable objects), ώστε να διασφαλίζεται ότι ένα τμήμα δε θα επηρεάζει αρνητικά τη λειτουργικότητα ενός άλλου.

Το LensKit ακολουθεί το μοτίβο στρατηγικής (Strategy Pattern). Στο μοτίβο αυτό κατά την ανάπτυξη μιας υλοποίησης δε χρησιμοποιείται η κληρονομικότητα. Αντί για υποκλάσεις που θα υλοποιούν τους διαφορετικούς τρόπους που μία κλάση θα επιτελεί κάποια λειτουργία της, χρησιμοποιείται η χρήση ξεχωριστών τμημάτων που ορίζονται από διεπαφές.[3] Η στρατηγική αυτή έχει ως άμεσο αποτέλεσμα οι επιμέρους αλλαγές στα τμήματα που υλοποιούν τις διεπαφές να μην απαιτούν αλλαγές στις κλάσεις που τα χρησιμοποιούν. Επιπρόσθετα, επιτρέπει την επανάχρηση των τμημάτων αυτών από άλλους αλγορίθμους μειώνοντας τον απαιτούμενο κώδικα για την υλοποίηση ενός αλγορίθμου, αλλά και βοηθώντας τον ερευνητή να επικεντρωθεί στην υλοποίηση που χρειάζεται για να ελέγξει την υπόθεσή του.

Παρά το ότι οι υλοποιημένοι αλγόριθμοι στο LensKit είναι εξαιρετικά διαμορφώσιμοι, δεν απαιτείται από το χρήστη να εμβαθύνει σε κάθε ξεχωριστό τμήμα,

ώστε να τους χρησιμοποιήσει, καθώς όπου είναι δυνατό έχουν οριστεί προεπιλογές.

Τέλος, κατά το σχεδιασμό του έχει γίνει προσπάθεια ελαχιστοποίησης των υποθέσεων που αφορούν στο είδος των δεδομένων που θα θελήσουν να χρησιμοποιήσουν οι χρήστες. Παρόμοια προσπάθεια γίνεται και όσον αφορά την υλοποίηση των ίδιων των αλγορίθμων όσο και τον αξιολογητή.

4.3 Οργάνωση του κώδικα - Ενότητες

Ο κώδικας οργανώνεται σε δέκα ενότητες (modules):

1. API: Περιλαμβάνει τις διεπαφές για υψηλού επιπεδου εργασίες στην παραγωγή συστάσεων. Είναι ανεξάρτητο από τις υπόλοιπες ενότητες εκτός αυτής των δομών δεδομένων.

2. Δομές Δεδομένων (Data structures): Περιλαμβάνει τις βασικές δομές δεδομένων του LensKit.

3. Πηρύνας (Core): Περιέχει το κύριο μέρος του LensKit, εκτός του αξιολογητή και των υλοποιήσεων των αλγορίθμων. Παρέχει υποστήριξη για το χειρισμό δεδομένων και τη διαμόρφωση των αλγορίθμων.

4. Αξιολογητής (Evaluator): Περιέχει υποστήριξη για την αξιολόγηση και τη σύγκριση αλγορίθμων με χρήση γνωστών μετρικών.

5. Παραγωγοί Προβλέψεων (Predictors): Εξειδικευμένη υποστήριξη για την πρόβλεψη βαθμολογιών.

6. k-NN: Συνεργατική Διήθηση πλησιέστερου γείτονα (είδους-είδους και χρήστη-χρήστη).

7. SVD: Συνεργατική Διήθηση μέσω παραγοντοποίησης μητρώων.

8. Slope1: Υλοποίηση του αλγορίθμου Slope One.

9. Grapht: Τεχνικά δεν αποτελεί μέρος του. Είναι εργαλειοθήκη της Java που τη χρησιμοποιεί για το χειρισμό της εισαγωγής εξαρτήσεων (dependency injection).

10. CLI: Διεπαφή γραμμής εντολών για εκτέλεση αλγορίθμων, χειρισμό αρχείων δεδομένων και επιθεώρηση διαμορφώσεων αλγορίθμων.

4.3.1 Διεπαφή Παραγωγών Συστάσεων

Αποτελεί τη διεπαφή με την οποία το δημόσιο API είναι προσβάσιμο. Δεν παρέχει κάποια υλοποίηση, αλλά χρησιμοποιείται για την ενθυλάκωση των διεπαφών των διαφόρων τμημάτων που υλοποιούν έναν παραγωγό συστάσεων. Είναι διαχωρισμένο από τις υλοποιήσεις του και ανήκει στο δημόσιο API.

Κεντρικό τμήμα ενός παραγωγού συστάσεων είναι η υλοποίηση της διεπαφής του βαθμολογητή ειδών (Item Scorer). Ο βαθμολογητής ειδών αποτελεί γενίκευση της παραγωγής προβλέψεων και υπολογίζει προσωποποιημένες ως προς το χρήστη βαθμολογίες (ratings). Υπολογίζει τις βαθμολογίες των ειδών (scores) προβλέποντας τις βαθμολογίες των χρηστών για αυτά. Αυτό αποτελεί μια γενίκευση που επιτρέπει τη βαθμολογία ειδών ακόμα και από τις εκδηλώσεις προτίμησης των χρηστών. Ο βαθμολογητής ειδών χρησιμοποιείται έμμεσα ενώ άμεσα χρήσιμοποιούνται οι παραγωγοι συστάσεων ειδών και οι παραγωγοί προβλέψεων βαθμολογιών. Και οι τρεις έχουν την ίδια διεπαφή και αυτό που τους διαφοροποιεί είναι ότι οι τελευταίοι αναλαμβάνουν να διεκπεραιώσουν δευτερεύουσες εργασίες που απαιτούνται για την παραγωγή συστάσεων (όπως τη αντιστοίχιση των βαθμολογιών σε κάποιο εύρος), ώστε να κρατείται «καθαρός» ο κώδικας του βαθμολογητή και να αποφεύγεται η επικάλυψη κώδικα.

4.4 Μοντέλο Δεδομένων

Το μοντέλο δεδομένων του LensKit στοχεύει στην αναπαράσταση και στην πρόσβαση στα δεδομένα που απαιτεί ένας παραγωγός συστάσεων. Για το σκοπό αυτό χρησιμοποιεί της έννοιες χρήστες (users), είδη (items) και γεγονότα (events). Το μοντέλο αυτό είναι αρκετά ευέλικτο, ώστε να υποστηρίζει και άμεσες βαθμολογίες και έμμεση εκδήλωση προτίμησης.

Οι χρήστες και τα είδη αναπαριστώνται με αριθμητικά αναγνωριστικά (numerical identifiers (Java longs).

Ως γεγονότος ορίζεται η αλληλεπίδραση του χρήστη με κάποιο είδος. Για κάθε τύπο γεγονότος, ορίζεται διαφορετική διεπαφή (επεκτάση του βασικού τύπου). Μέχρι στιγμής έχουν υλοποιηθεί τρεις τύποι γεγονότων: η Βαθμολόγηση (Rating)[5], η Προτίμηση (Like)[6] και η Μαζική Προτίμηση (LikeBatch)[7]. Η Προτίμηση χρησιμοποιείται ως έκφραση απλής εκδήλωσης προτίμησης. Η Μαζική Προτίμηση αφορά τις Προτιμήσεις πολλών χρηστών.

Η επικοινωνία των διαφόρων τμημάτων του παραγωγού συστάσεων με τα δεδομένα γίνεται μεσω αντικειμένων για πρόσβαση σε δεδομένα (Data Access Objects (DAOs)). Υπάρχει ευελιξία όσον αφορά των τρόπο που μπορεί να είναι αποθηκευμένα τα δεδομένα καθώς μπορούν να υλοποιηθούν DAOs ανάλογα με τις ανάγκες

[5] http://lenskit.org/apidocs/org/grouplens/lenskit/data/event/Rating.html
[6] http://lenskit.org/apidocs/org/grouplens/lenskit/data/event/Like.html
[7] http://lenskit.org/apidocs/org/grouplens/lenskit/data/event/LikeBatch.html

του παραγωγού συστάσεων. Το LensKit παρέχει DAOs για χειρισμό αρχείων κειμένου και βάσεων δεδομένων. Έχουν υλοποιηθεί βασικές μέθοδοι, οι οποίες επιστρέφουν βασικές δομές δεδομένων της Java και μέθοδοι συνεχούς ροής (streaming) μέθοδοι, οι οποίες επιστρέφουν κέρσορες (cursors). Οι μέθοδοι συνεχούς ροής προσφέρουν αποδοτική διαχείριση μνήμης καθώς επιτρέπουν την επεξεργασία ενός αντικειμένου κάθε φορά χωρίς να φορτώνονται όλα τα απαιτούμενα αντικείμενα στη μνήμη. Υπάρχει η δυνατότητα επέκτασης των διεπαφών για αξιοποίηση μεταδεδομένων των χρηστών ή των αντικειμένων. Τα DAOs είναι τμήματα, όπως αυτά που επιτελούν τις λειτουργίες ενός παραγωγού συστάσεων.

4.4.1 Δομές Δεδομένων

Για την επεξεργασία των δεδομένων είναι συχνά αναγκαία η χρήση διανυσμάτων (vectors). Για το σκοπό αυτό το LensKit προσφέρει τα Sparse Vectors (Αραιά διανύσματα). Αποτελούν αντιστοίχιση (map) από long σε double και είναι αποδοτικά σε πράξεις γραμμικής άλγεβρας. Λειτουργούν ως παράλληλοι πίνακες αναγνωριστικών (IDs) χρηστών ή ειδών και τιμών και ταξινόμούνται βάσει ID. Ο ορισμός του χώρου κλειδιών (key domain) τους γίνεται κατά τη στιγμή της δημιουργίας του και αυτό έχει ως αποτέλεσμα αποδοτική διαχείριση της μνήμης. Τα κλειδιά για τα οποία έχει οριστεί τιμή αποτελούν το σύνολο κλειδιών (key set). Υπάρχει η αφηρημένη κλάση SparseVector , η οποία υποστηρίζει μεθοδους μόνο ανάγνωσης. Η κλάση ImmutableSparseVector, διασφαλίζει τη μη πραγματοποίηση αλλαγών στο διάνυσμα, σε αντίθεση με τη MutableSparseVector. Αν ο χώρος κλειδιών δεν είναι αρχικά γνωστός γίνεται χρήση πινάκων κατακερματισμού (hash map) και στη συνέχεια μετατροπή τους σε διανύσματα.

Το LensKit χρησιμοποιεί τη βιβλιοθήκη fastutil[8]και το Google Guava[9]. Η fastutil παρέχει συλλογές (collections), οι οποίες είναι συμβατές με τη διεπαφή συλλογών της Java (Java Collection API) και επιτρέπει στο LensKit τη χρήση λιστών, συνόλων και αντιστοιχίσεων. Επιπλέον παρέχει γρήγορη επανάληψη (fast iteration), η οποία μειώνει της απαιτήσεις δέσμευσης μνήμης.

Υπάρχει η πρόθεση να αντικατάστασης των Sparse Vectors[10] είτε μέσω των δομών που προσφέρει το fastutil είτε μέσω αυτών του HPPC[11]. Επίσης, υπάρχει η σκέψη δημιουργίας νέων τύπων για τα αποτελέσματα που να υποστηρίζουν την επιστροφή και τον χειρισμό περισσότερων πληροφοριών εκτός των συστάσεων, όπως το βαθμό εμπιστοσύνης σε αυτές.[12] Στόχος είναι η διατήρηση της αποδοτικότητας, αλλά με μείωση της πολυπλοκότητας για τους προγραμματιστές.[13]

[8] http://fastutil.di.unimi.it/
[9] https://github.com/google/guava
[10] https://github.com/lenskit/lenskit/wiki/ReplacingSparseVectors
[11] labs.carrotsearch.com/hppc.html
[12] https://github.com/lenskit/lenskit/wiki/DetailedResults
[13] http://mailman.cs.umn.edu/archives/lenskit/2015q1/000555.html

4.5 Modular αλγόριθμοι

Βασική αρχή σε όλες τις υλοποιήσεις είναι η τμηματοποίηση των αλγορίθμων, ώστε κάθε τμήμα να εκτελεί μία και μόνο λειτουργία με στόχο την επαναχρησιμοποίηση κώδικα και την ευκολότερη παραμετροποίηση και ανάπτυξη κώδικα. Το LensKit παρέχει τις απαραίτητες υποδομές, ώστε να διευκολύνει την υλοποίηση αλγορίθμων με αυτή τη στρατηγική.

Το μοτίβο στρατηγικής [6]χρησιμοποιεί το LensKit επιτρέπει να υπάρχουν ξεχωριστά τμήματα που επιτελούν λειτουργίες, όπως η κανονικοποίηση δεδομένων, που μπορούν να χρησιμοποιηθούν από τους υλοποιημένους αλγορίθμους.

Στο LensKit γίνεται εκτεταμένη χρήση κατασκευαστών (builders). Ακολουθείται η μέθοδος του διαχωρισμού των τμημάτων που περιέχουν τα δεδομένα (data containers) (και συνοδεύονται από διεπαφές για την πρόσβαση σε αυτά) και των κατασκευαστών που κάνουν υπολογισμούς για την παραγωγή (φαινομενικά αμετάβλητων) αντικειμένων. Έτσι μπορεί εύκολα να αλλάξει η στρατηγική πραγματοποίησης των απαιτούμενων υπολογισμών χωρίς να αλλάξει η υλοποίηση του υπόλοιπου αλγορίθμου.

Επιπλέον χρησιμοποιούν τη τεχνική Builder and Facade [3] (Κατασκευή και Εκπροσώπηση), η οποία περιλαμβάνει κατασκευαστές και μία διεπαφή που λειτουργεί ως εκπρόσωπος (διεπαφή) άλλων διεπαφών.

Για την επίτευξη των παραπάνω το LensKit χρησιμοποιεί τη μέθοδο της εισαγωγής εξαρτήσεων[8], δηλαδή ένα τμήμα κώδικα όταν καλείται πρέπει να του παρέχονται και τα αντικείμενα από τα οποία εξαρτάται αντί αυτά να ενεργοποιούνται από το ίδιο το τμήμα. Με την τεχνική αυτή είναι δυνατό να αλλάξει η υλοποίηση κάποιας εξάρτησης και να αλλάξει η συμπεριφορά του εξαρτώμενου τμήματος χωρίς αυτό να επαναπρογραμματιστεί. Για το σκοπό αυτό το LensKit χρησιμοποιεί την εργαλειοθήκη Grapht.

Τα τμήματα χωρίζονται σε αυτά που είναι προκατασκευασμένα (pre-built)από τα δεδομένα και μπορούν να χρησιμοποιηθούν από διαφορετικές κλήσεις και αυτά που χρειάζονται άμεση πρόσβαση στα δεδομένα.

4.5.1 Βασικές Υλοποιήσεις Τμημάτων

Λόγω της δομής του LensKit είναι δυνατόν να υλοποιηθεί κάποιος αλγόριθμος παραγωγής συστάσεων διαμορφώνοντας κάποιον υλοποιημένο item scorer ή υλοποιώντας μόνο το κομμάτι του αλγορίθμου που αφορά τον item scorer. Η προεπιλεγμένη υλοποίηση του item scorer είναι ο TopNItemRecommender[14].

[14] http://lenskit.org/apidocs/org/grouplens/lenskit/basic/TopNItemRecommender.html

4.5.2 Παραγωγοί Περιλήψεων Ιστορικού και Κανονικοποιητές

Ένας παραγωγός περιλήψεων ιστορικού παράγει από το ιστορικό γεγονότων του χρήστη ένα αραιό διάνυσμα προτιμήσεων του οποίου τα κλειδιά είναι τα αντικείμενα και οι τιμές είναι κάποιες πραγματικές τιμές που αντιπροσωπεύουν τις προτιμήσεις του χρήστη. Πολλά συστήματα ενώ δε χρησιμοποιούν άμεσες βαθμολογίες, παράγουν ένα διάνυσμα που σχετίζεται με τις προτιμήσεις του χρήστη. Επίσης αυτό το διάνυσμα συχνά κανονικοποιείται ως προς το μέσο όρο των βαθμολογιών. Προεπιλεγμένη κανονικοποίηση είναι η μοναδιαία.[15]

Οι κανονικοποιητές εφαρμόζονται πάνω σε διανύσματα αναφοράς και διανύσματα στόχους. Το διάνυσμα αναφοράς χρησιμοποιείται για τον υπολογισμό της βάσης της κανονικοποίησης και το διάνυσμα-στόχος είναι αυτό το οποίο τροποποιείται. Για λόγους αντιστρεψιμότητας υπάρχει η δυνατότητα δημιουργίας ενός μετασχηματισμού από ένα διάνυσμα αναφοράς. Ο μετασχηματισμός μετά μπορεί να εφαρμοστεί σε οποιοδήποτε διάνυσμα. Υποστηρίζονται και στοχευμένοι ως προς τον χρήστη ή το αντικείμενο κανονικοποιητές. Οι κανονικοποιητές αυτοί μπορούν να κάνουν χρήση επιπλέον πληροφορίας. Συχνά εξαρτώνται από DAOs για την πρόσβαση σε δεδομένα ή από κάποιο άλλο τμήμα το οποίο μπορεί να επωφεληθεί απο τη γνώση για το ποιο διάνυσμα κανονικοποιείται.

4.5.3 Βαθμολογητές Βάσης

Υπολογίζουν βαθμολογίες για τα είδη με χρήση απλών μέσων όρων. Χρησιμοποιούνται τόσο σε περίπτωση αποτυχίας του βασικού βαθμολογητή ειδών όσο και για κανονικοποίηση δεδομένων. Βασικοί αλγόριθμοι χρησιμοποιούν κανονικοποιημένα δεδομένα αντί για τις πραγματικές βαθμολογίες. Οι Βαθμολογητές βάσης υλοποιούν τη διεπαφή του βαθμολογητή ειδών και οποιοσδήποτε βαθμολογητής ειδών μπορεί να χρησιμοποιηθεί ως βαθμολογητής βάσης. Υποστηρίζει τη χρήση μιας παραμέτρου απόσβεσης (damping parameter), ώστε αντικείμενα με λίγες βαθμολογίες να μην παίρνουν αδικαιολόγητα μεγάλες τιμές. Αν ο χρήστης δεν έχει βαθμολογήσει αρκετά είδη, μπορούν να ανατεθούν οι τιμές του βαθμολογητή βάσης. Αν έχει χρησιμοποιηθεί ο γενικός μέσος όρος ως βαθμολογητής βάσης, τότε ο βαθμολογητής χρησιμοποιεί το μέσο όρο βαθμολογιών του χρήστη, ο οποίος εκφυλίζεται στο γενικό μέσο όρο όταν ο χρήστης δεν έχει βαθμολογίες.

4.5.4 Διαμόρφωση αλγορίθμων

Η διαμόρφωση των αλγορίθμων γίνεται μέσω του Grapht API με τη χρήση μιας ενσωματωμένης γλώσσας σεναρίων σε Groovy, η οποία έχει πιο «ελαφριά» σύνταξη σε σχέση με τη Java. Δίνει τη δυνατότητα οι ορισμοί των αλγορίθμων να

[15] http://lenskit.org/apidocs/org/grouplens/lenskit/transform/normalize/DefaultUserVectorNormalizer.html

αντιμετωπίζονται ως αρχεία διαμόρφωσης (configuration files) και να μην είναι ενσωματωμένα στον κώδικα των εφαρμογών. Η γραμμή εντολών του LensKit μπορεί να χειριστεί αυτά τα σενάρια.

4.5.5 Αξιολόγηση αλγορίθμων και Σύνολα Δεδομένων

Υποστηρίζεται offline, προσανατολισμένη στα δεδομένα αξιολόγηση της απόδοσης μέσω εκπαίδευσης και ελέγχου (train/test) με διαχωρισμό και επαλήθευση (cross-validation). Υποστηρίζεται επεξεργασία των συνόλων δεδομένων και αξιολόγηση αλγορίθμων πάνω σε πολλαπλά σύνολα δεδομένων και μέτρηση της απόδοσής τους. Στο LensKit 3 θα γίνεται μέσω της διεπαφής γραμμής εντολών (CLI) και του Gradle[16].[17] Ο αξιολογητής χειρίζεται δεδομένα κυρίως σε μορφή delimited (οριοθετημένων) αρχείων κειμένου και δυαδικών αρχείων βαθμολογιών. Οι βασικές εργασίες χειρισμού δεδομένων είναι:

- crossfold: Διαχωρισμός της πηγής δεδομένων σε N τμήματα για cross-validation.
- subsample: Δημιουργία ενός μικρότερου συνόλου δεδομένων με τυχαία επιλογή από ένα μεγαλύτερο βάσει κριτηρίου. Θα καταργηθεί στο LensKit 3.
- pack: Μετατροπή ενός συνόλου δεδομένων σε δυαδικό αρχείο για αποδοτική πρόσβαση.

4.5.6 Αξιολόγηση Αλγορίθμων και Μετρικές Απόδοσης

Ο αξιολογητής δέχεται ένα σύνολο αλγορίθμων και ζεύγη συνόλων δεδομένων εκπαίδευσης και ελέγχου και αξιολογεί την ακρίβεια κάθε αλγορίθμου πάνω σε κάθε ζεύγος. Οι διαθέσιμες μετρικές είναι δύο ειδών[18]: μετρικές πρόβλεψης[19] και TopN μετρικές[20]. Οι μετρικές πρόβλεψης περιλαμβάνουν:

- MAE: Μέσο απόλυτο σφάλμα της ακρίβειας των προβλεφθέντων δεδομένων βαθμολόγησης.
- RMSE: Ομοία με μέσο τετραγωνικό σφάλμα.
- Coverage (κάλυψη): Μετράει τον αριθμό των παρεχόμενων προβλέψεων βαθμολογιών σε και υπολογίζει την κάλυψη σε σχέση με το σύνολο των ζητούμενων.

[16] http://gradle.org/
[17] http://mailman.cs.umn.edu/archives/lenskit/2015q1/000560.html
[18] http://lenskit.org/documentation/evaluator/upgrading/
[19] http://lenskit.org/master/apidocs/org/lenskit/eval/traintest/predict/PredictMetric.html
[20] http://lenskit.org/master/apidocs/org/lenskit/eval/traintest/recommend/TopNMetric.html

- nDCG: Αξιολογεί τις προβλέψεις του παραγωγού συστάσεων μέσω κανονικοποιημένου μειούμενου συσσωρευτικού οφέλους. Τα είδη ταξινόμούνται βάσει της προβλεφθείσας προτίμησης και η μετρική υπολογίζεται χρησιμοποιώντας τις πραγματικές βαθμολογίες του χρήστη ως συνάρτηση ωφέλειας για κάθε είδος. Με αυτό τον τρόπο δεν «τιμωρεί» τον παραγωγό συστάσεων όταν προτείνει είδη τα οποία θα ήταν καλά για το χρήστη, αλλά για τα οποία δεν είχε δεδομένα.

Οι TopN μετρικές περιλαμβάνουν:

- Entropy (Εντροπία): Εξετάζει το εύρος των προτεινόμενων ειδών σε σχέση με το σύνολο των ειδών για όλους τους χρήστες. Όσο μικρότερη η τιμή που επιστρέφει η μετρική τόσο λιγότερα είναι τα είδη από τα οποία προτείνει ο παραγωγός. Στη χειρότερη περίπτωση τα είδη που προτείνονται είναι τα ίδια για όλους τους χρήστες.

- Length (Μήκος): Υπολογίζει τον αριθμό των συστάσεων.

- MAP: Υπολογίζει τη μέση ακρίβεια για τους μέσους όρους ακρίβειας κάθε λίστας που επιστρέφεται.

- MRR: Υπολογίζει τη μέση αμοιβαία κατάταξη.

- nDCG: Όμοια με την αντίστοιχη μετρική πρόβλεψης.

- Popularity (Δημοφιλία): Μετράει πόσο δημοφιλή είναι τα είδη που επιστρέφονται από τον παραγωγό συστάσεων.

- Precision Recall: Ακρίβεια και ανάκληση σε σταθερού μήκους σύνολα ειδών υποψήφιων προς σύσταση και επιθυμητών ειδών. Αν στο δεύτερο σύνολο τοποθετηθούν μη επιθυμητά είδη, η μετρική μπορεί να ελέγξει αν ένας παραγωγός κάνει κακές συστάσεις.

4.5.7 Εισαγωγή Εξαρτήσεων

Πριν την αρχικοποίηση ενός αντικειμένου, όλες οι εξαρτήσεις πρέπει να είναι διαθέσιμες. Για την επίτευξη αυτού έχουν αναπτυχθεί εργαλειοθήκες, οι dependency injectors (εισαγωγείς εξαρτήσεων), οι οποίοι κάνουν αυτόματα την απαραίτητη διαδικασία. Το Grapht ακολουθεί πολιτική ευαίσθητη ως προς το περιεχόμενο, επιτρέποντας στα αντικείμενα να διαμορφώνονται βάσει του πού χρησιμοποιούνται. Διαχωρίζει την επίλυση εξάρτησης από την δημιουργία των στιγμιοτύπων των αντικειμένων και εκθέτει το γράφημα των επιλυμένων εξαρτήσεων των αντικειμένων ως ένα αντικείμενο που μπορεί να αναλυθεί και να διαχειρισθεί. Πολλά από τα χαρακτηριστικά του είναι υλοποιημένα με όρους μετασχηματισμού γραφημάτων.

Εισαγωγή εξάρτησης είναι σχεδιασμός που προκύπτει από την εφαρμογή αντιστροφής ελέγχου (Inversion of Control) στο πρόβλημα της δημιουργίας στιγμιοτύπων αντικειμένων που εξαρτώνται από άλλα. Μέσω αυτού αν ένα αντικείμενο Α εξαρτάται από ένα αντικείμενο Β, μπορεί να ζητήσει να του παρασχεθεί το Β μέσω ενός ορίσματος στο δημιουργό του. Έτσι, τα τμήματα κώδικα δε γνωρίζουν για την υλοποίηση των εξαρτήσεών τους. Τα τμήματα κώδικα μπορούν να επαναδιαμορφωθούν με αλλαγή των υλοποιήσεων των εξαρτήσεών τους, χωρίς να αλλάξουν τα ίδια. Διευκολύνεται ο έλεγχος λειτουργίας τους και οι εξαρτήσεις κάθε τμήματος δηλώνονται με άμεσο τρόπο.

Κεφάλαιο 5

Ανάλυση της Υλοποίησης

Κεφάλαιο 6

Συμπεράσματα και Μελλοντικές Κατευθύνσεις

Ο αλγόριθμος HIR απαντάει στο πρόβλημα της έλλειψης πληροφορίας για τις προτιμήσεις των νέων χρηστών και τη δημοφιλία των νέων ειδών που εντάσσονται σε ένα σύστημα παραγωγής συστάσεων.

Η υλοποίηση του αλγορίθμου αυτού σε ένα σύστημα όπως το LensKit, δίνει τη δυνατότητα σε αυτή να αποτελέσει ένα τμήμα μιας νέας ενότητας που θα αφορά αλγορίθμους συνεργατικής διήθησης που θα αντιμετωπίζουν το πρόβλημα της κρύας εκκίνησης. Τα εργαλεία που προσφέρει το LensKit παρέχουν την ευκαιρία σύγκρισης των διαφόρων αλγορίθμων που στοχεύουν στην ευφυή παραγωγή καλών συστάσεων σε αυτό το τόσο προκλητικό ζήτημα που αντιμετωπίζουν όλοι οι αλγόριθμοι παραγωγής συστάσεων.

Η υλοποίηση τέτοιων αλγορίθμων στο LensKit, βοηθα τους ερευνητές να χρησιμοποιήσουν προχωρημένες τεχνικές προγραμματισμού, οι οποίες έχουν ως αποτέλεσμα κομψές υλοποιήσεις τμήματα των οποίων μπορούν να επαναχρησιμοποιηθούν, βοηθώντας στη γρηγορότερη εξέλιξη της έρευνας στον τομέα.

Κεφάλαιο 7

Ορολογία

Ξενόγλωσσος Όρος	**Ελληνικός Όρος**
APIs	Διεπαφές Προγραμματισμού Εφαρμογών
Baseline Scorer	Βαθμολογητής Βάσης
batch	μαζικός
Batch Like	Μαζική Προτίμηση
builder	κατασκευαστής
Builder and Facade	Κατασκευή και Εκπροσώπηση
cold start	κρύα εκκίνηση
Collaborative Filtering	Συνεργατική Διήθηση
collection	συλλογή
Command Line interface	Διεπαφή Γραμμής Εντολών
χομμυνιτψ-βασεδ	βάσει κοινότητας
component	τμήμα
configuration files	αρχεία διαμόρφωσης
content-based	βάσει περιεχομένου
core	πηρύνας
cross-validation	διαχωρισμού-επαλήθευσης
damping parameter	παράμετρος απόσβεσης
DAO	Αντικείμενο για την πρόσβαση σε δεδομένα
data container	τμήμα που περιέχει δεδομένα
Data Structures	Δομές Δεδομενων
delimited files	οριοθετημένα αρχεία
demographic	δημογραφικός
Dependency injection	Εισαγωγή εξαρτήσεων
dependency injectors	εισαγωγείς εξαρτήσεων

Συνέχεια στην επόμενη σελίδα

Συνέχεια από τη προηγούμενη σελίδα

Ξενόγλωσσος Όρος	**Ελληνικός Όρος**
Entropy	Εντροπία
Evaluator	Αξιολογητής
Event	Γεγονός
fast iteration	γρήγορη επανάληψη
framework	πλαίσιο εργασίας
hash map	πίνακας κατακερματισμού
history summarizer	παραγωγός περιλήψης ιστορικού
HIR	Ιεραρχική Κατάταξη βάσει του χώρου των ειδών
hybrid	υβριδικός
ID	αναγνωριστικό
immutable object	αμετάβλητο αντικείμενο
Immutable Sparse Vectors	Αμετάβλητα Αραία Διανύσματα
Inversion of Control	Αντιστροφή Ελέγχου
item	είδος
item-based	βασιζόμενο στα είδη
Item Recommender	Παραγωγός Συστάσεων Ειδών
Item Scorer	Βαθμολογητής Ειδών
collection API	Διεπαφή συλλογών
key domain	χώρος κλειδιών
key set	σύνολο κλειδιών
knowledge-based	βάσει γνώσης
latent factor	λανθάνων παράγοντας
length	μήκος
Like	Προτίμηση
map	αντιστοίχιση
matrix factorizaion	παραγοντοποίηση μητρώου
Mean absolute error	Μέσο απόλυτο σφάλμα
model based	βασιζόμενη στο μοντέλο
module	ενότητα
MOOC	Μαζικό ανοιχτό διαδικτυακό μάθημα
Mutable Sparse Vectors	Μεταβλητά Αραία Διανύσματα
nearest-neighbors	κοντινότερων γειτόνων
New Community Problem	Πρόβλημα Νέας Κοινότητας
New Items Problem	Πρόβλημα Νέων Ειδών
New Users Problem	Πρόβλημα Νέων Χρηστών
normalizer	κανονικοποιητής
numeric identifiers	αριθμητικά αναγνωριστικά

Συνέχεια στην επόμενη σελίδα

Συνέχεια από τη προηγούμενη σελίδα

Ξενόγλωσσος Όρος	**Ελληνικός Όρος**
offline	εκτός σύνδεσης
pack	δέσμη
popularity	δημοφιλία
pre-built	προκατασκευασμένα
precision	ακρίβεια
predict	πρόβλεψη
predictor	παραγωγός προβλέψεων
Ranking	Κατάταξη
Rating	Βαθμολογία χρήστη
Rating Predictor	Παραγωγός Πρόβλεψης Βαθμολογίας
recall	ανάκληση
Recommender	Παραγωγός Συστάσεων
Root mean squared error	Μέσο τετραγωγικό σφάλμα
subsample	υπο-δείγμα
score	βαθμολογία είδους
script	σενάριο
Sparse Vectors	Αραιά Διανύσματα
sparsity	αραιότητα
Strategy Pattern	Μοτίβο στρατηγικής
streaming	συνεχούς ροής
SVD	Παραγοντοποίηση Ιδιαζουσών Τιμών
train/test	εκπαίδευσης και ελέγχου
toolkit	εργαλειοθήκη
user	χρήστης
user-based	βασιζόμενο στους χρήστες
vector	διάνυσμα

Κεφάλαιο 8

Συντμήσεις-Αρκτικόλεξα

Σύντμηση	Πλήρης Ανάπτυξη
API	Application programming interface
CLI	Command Line interface
DAO	Data Access Object
HIR	Hierarchical Itemspace Rank
HPPC	High Performance Primitive Collections for Java
MAE	Mean Absolute Error
MAP	mean average precision
MRR	Mean reciprocal rank
MOOC	Massive Open Online Course
nDCG	Normalized discounted cumulative gain
RMSE	Root mean squared error
SVD	Singular Value Decomposition

Βιβλιογραφία

[1] Burke, Robin. *The adaptive web*. chapter Hybrid Web Recommender Systems, pages 377–408. Springer-Verlag, Berlin, Heidelberg, 2007, ISBN 978-3-540-72078-2. http://dl.acm.org/citation.cfm?id=1768197.1768211.

[2] Ekstrand, Michael D.. *Towards Recommender Engineering: Tools and Experiments in Recommender Differences*. Ph.d thesis, University of Minnesota, Minneapolis, MN, July 2014. http://md.ekstrandom.net/research/thesis/.

[3] Gamma, Erich, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-oriented Software*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995, ISBN 0-201-63361-2.

[4] Good, Nathaniel, J. Ben Schafer, Joseph A. Konstan, Al Borchers, Badrul Sarwar, Jon Herlocker, and John Riedl. *Combining collaborative filtering with personal agents for better recommendations*. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence*, AAAI '99/IAAI '99, pages 439–446, Menlo Park, CA, USA, 1999. American Association for Artificial Intelligence, ISBN 0-262-51106-1. http://dl.acm.org/citation.cfm?id=315149.315352.

[5] Gori, Marco and Augusto Pucci. *Itemrank: A random-walk based scoring algorithm for recommender engines*. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, pages 2766–2771, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc. http://dl.acm.org/citation.cfm?id=1625275.1625720.

[6] Konstan, Joseph A., J. D. Walker, D. Christopher Brooks, Keith Brown, and Michael D. Ekstrand. *Teaching recommender systems at large scale:*

Evaluation and lessons learned from a hybrid mooc. ACM Trans. Comput.-Hum. Interact., 22(2):10:1–10:23, April 2015, ISSN 1073-0516. http://doi.acm.org/10.1145/2728171.

[7] Koren, Yehuda. *Factorization meets the neighborhood: A multifaceted collaborative filtering model*. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 426–434, New York, NY, USA, 2008. ACM, ISBN 978-1-60558-193-4. http://doi.acm.org/10.1145/1401890.1401944.

[8] Martin, Robert C.. *The dependency inversion principle*. C++ Report, 8, May 1996.

[9] Nikolakopoulos, Athanasios N. and John D. Garofalakis. *NCDawareRank: a novel ranking method that exploits the decomposable structure of the web*. In *Proceedings of the sixth ACM international conference on Web search and data mining*, WSDM '13, pages 143–152, New York, NY, USA, 2013. ACM, ISBN 978-1-4503-1869-3. http://doi.acm.org/10.1145/2433396.2433415.

[10] Nikolakopoulos, Athanasios N., Marianna A. Kouneli, and John D. Garofalakis. *Hierarchical itemspace rank: Exploiting hierarchy to alleviate sparsity in ranking-based recommendation*. Neurocomputing, 163(1):126 – 136, 2015, ISSN 0925-2312. http://www.sciencedirect.com/science/article/pii/S0925231215002180, Recent Advancements in Hybrid Artificial Intelligence Systems and its Application to Real-World ProblemsProgress in Intelligent SystemsMining Humanistic DataSelected papers from the 7th International Conference on Hybrid Artificial Intelligence Systems (HAIS 2012)Selected papers from the 2nd Brazilian Conference on Intelligent Systems (BRACIS 2013)Selected papers from the 2nd Mining Humanistic Data Workshop at {EANN} 2013.

[11] Ricci, Francesco, Lior Rokach, Bracha Shapira, and Paul B. Kantor. *Recommender systems handbook*. Springer, 2011.

[12] Sinha, Rashmi R. and Kirsten Swearingen. *Comparing recommendations made by online systems and friends*. In *DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries*, 2001. http://dblp.uni-trier.de/db/conf/delos/delos2001.html#SinhaS01.

[13] Takács, Gábor, István Pilászy, Bottyán Németh, and Domonkos Tikk. *Major components of the gravity recommendation system*. SIGKDD Explor. Newsl., 9(2):80–83, December 2007, ISSN 1931-0145. http://doi.acm.org/10.1145/1345448.1345466.