ΑΛΕΞΑΝΔΡΕΙΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ
ΙΔΡΥΜΑ ΘΕΣΣΑΛΟΝΙΚΗΣ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ – WEB INTELLIGENCE

Συστήματα Συστάσεων με χρήση Μεθόδων Μηχανικής Μάθησης

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΓΕΩΡΓΙΟΥ Δ. ΠΑΠΑΔΟΠΟΥΛΟΥ

Επιβλέπων : Κωνσταντίνος Διαμαντάρας
Καθηγητής, Α.Τ.Ε.Ι.Θ.

Θεσσαλονίκη, Ιανουάριος 2017

Η σελίδα αυτή είναι σκόπιμα λευκή.

ΑΛΕΞΑΝΔΡΕΙΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΘΕΣΣΑΛΟΝΙΚΗΣ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ – WEB INTELLIGENCE

Συστήματα Συστάσεων με χρήση Μεθόδων Μηχανικής Μάθησης

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΓΕΩΡΓΙΟΥ Δ. ΠΑΠΑΔΟΠΟΥΛΟΥ

Επιβλέπων : Κωνσταντίνος Διαμαντάρας
Καθηγητής Α.Τ.Ε.Ι.Θ.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή στις 18 Φεβρουαρίου 2017.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Κωνσταντίνος Διαμαντάρας	Δημήτριος Δέρβος	Κωνσταντίνος Γουλιάνας
Καθηγητής Α.Τ.Ε.Ι.Θ.	Καθηγητής Α.Τ.Ε.Ι.Θ.	Επίκουρος Καθηγητής Α.Τ.Ε.Ι.Θ.

Θεσσαλονίκη, Φεβρουάριος 2017

(Υπογραφή)

.....

Γεώργιος Δ. Παπαδόπουλος

Μηχανικός Πληροφορικής Α.Τ.Ε.Ι.Θ.

Περίληψη

Το πρόβλημα της Πληροφοριακής Υπερφόρτωσης, είναι εντονότερο στις μέρες μας όσο ποτέ άλλοτε. Τα Συστήματα Συστάσεων αποτελούν ένα ταχέως αναπτυσσόμενο υποπεδίο της Ανάκτησης Πληροφορίας και προσπαθούν να δώσουν μια λύση στο πρόβλημα, παράγοντας συστάσεις νέων αντικειμένων προς τους χρήστες τους. Η δημοφιλέστερη τεχνική για συστήματα συστάσεων είναι το Συνεργατικό Φιλτράρισμα, για την οποία έχει προταθεί μεγάλο πλήθος αλγορίθμων μηχανικής μάθησης. Παρόλα αυτά, δεν υπάρχει κάποια ευρέως αποδεκτή μεθοδολογία αξιολόγησης της απόδοσης των αλγορίθμων Συνεργατικού Φιλτραρίσματος, συνεπώς τα πειραματικά αποτελέσματα που παρουσιάζονται στην βιβλιογραφία δεν είναι εύκολα συγκρίσιμα μεταξύ τους.

Στο πλαίσιο της διπλωματικής εργασίας γίνεται μια επισκόπηση των βασικών εννοιών και των δημοφιλέστερων προσεγγίσεων για συστήματα συστάσεων, καθώς και των κυριότερων μεθόδων μηχανικής μάθησης που χρησιμοποιούνται σε αυτά. Στη συνέχεια αναλύονται δημοφιλείς αλγόριθμοι Συνεργατικού Φιλτραρίσματος, αναπτύσσεται μια μεθοδολογία αξιολόγησης τέτοιων αλγορίθμων και γίνεται μια συγκριτική μελέτη διαφόρων τεχνικών για πρόβλεψη αξιολογήσεων χρηστών για αντικείμενα, όσον αφορά την ακρίβεια πρόβλεψης τους. Για την πειραματική διαδικασία χρησιμοποιούνται κυρίως βιβλιοθήκες ανοιχτού κώδικα για συστήματα συστάσεων και η αξιολόγηση γίνεται σε τέσσερις γνωστές συλλογές δεδομένων. Τα αποτελέσματα έδειξαν την ανωτερότητα των μεθόδων βασισμένων σε μοντέλα έναντι των μεθόδων βασισμένων στη μνήμη, όσον αφορά την ακρίβεια των προβλέψεων και την δυνατότητα κλιμάκωσης, ειδικά σε πολύ αραιά σύνολα δεδομένων.

Λέξεις Κλειδιά: Συστήματα Συστάσεων, Συνεργατικό Φιλτράρισμα, Πρόβλεψη αξιολογήσεων, Μηχανική μάθηση, Παραγοντοποίηση πινάκων, Αξιολόγηση απόδοσης

Η σελίδα αυτή είναι σκόπιμα λευκή.

Abstract

Nowadays, the problem of Information Overload is becoming more and more intense than ever before. Recommender Systems are a fast growing subfield of Information Retrieval which aims at giving a solution to the problem by recommending new items to its users. The most popular technique for recommender systems construction is that of Collaborative Filtering, for which a large number of Machine Learning algorithms has been proposed. However, there is no universally accepted methodology for Collaborative filtering algorithms performance evaluation, therefore the experimental results presented in the literature are not easily comparable to each other.

This particular research provides an overview of the basic concepts, the most popular approaches, as well as the most widespread machine learning methods used in recommender systems. Furthermore, it analyzes popular collaborative filtering algorithms, it develops a methodology concerning the evaluation of such algorithms and it also conducts a comparative study of various techniques regarding the rating prediction task, in terms of their predictive accuracy. The experimental study is carried out by using mainly open source frameworks for recommender systems and the evaluation is done on four well known datasets. The results confirmed the superiority of model based methods over memory based methods in terms of both predictive accuracy and scalability, especially in very sparse datasets.

Keywords: Recommender Systems, Collaborative filtering, Rating prediction, Machine learning, Matrix factorization, Performance evaluation

Στην οικογένεια μου….

Πίνακας περιεχομένων

1

Εισαγωγή

1.1 Πληροφοριακή υπερφόρτωση και λήψη αποφάσεων

Η ανθρώπινη καθημερινότητα είναι γεμάτη με πιθανές επιλογές (Ekstrand, 2011). Σε καθημερινή βάση ο καθένας μας βρίσκεται αντιμέτωπος με την ανάγκη λήψης αποφάσεων. Οι αποφάσεις αυτές μπορεί να αφορούν ποικίλες πτυχές της ζωής μας, όπως τα ρούχα που θα φορέσουμε, ποια ταινία θα επιλέξουμε να παρακολουθήσουμε, τις μετοχές που θα αγοράσουμε, το εστιατόριο που θα επιλέξουμε, τα βιβλία που θα διαβάσουμε, τις σελίδες που θα επισκεφτούμε στο διαδίκτυο. Ο αριθμός των δυνατών επιλογών είναι πολύ συχνά αποτρεπτικά μεγάλος – εκατοντάδες μάρκες ρούχων, δεκάδες χιλιάδες ταινίες, δεκάδες χιλιάδες βιβλία και άλλα καταναλωτικά προϊόντα, εκατοντάδες εκατομμύρια άρθρα, ιστοσελίδες και βίντεο στο διαδίκτυο.

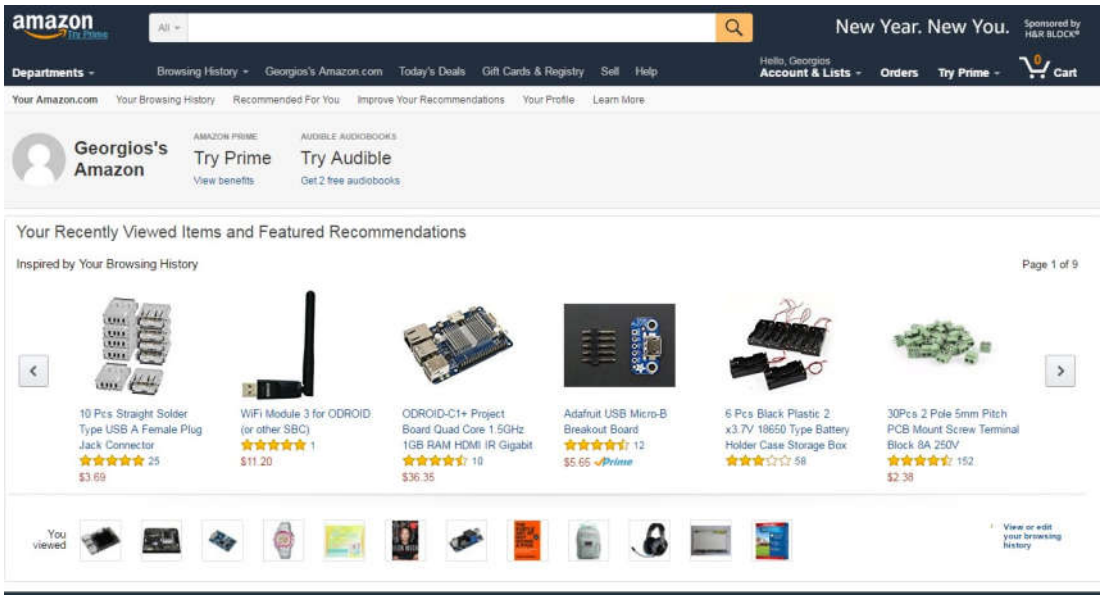
Στην κατάσταση αυτή συνέβαλλε καθοριστικά και η ψηφιακή επανάσταση των τελευταίων δεκαετιών που είχε ως αποτέλεσμα την αλματώδη αύξηση της ηλεκτρονικής πληροφορίας που διακινείται μέσω του Διαδικτύου. Έτσι, όσο ο όγκος των πληροφοριών που γίνεται διαθέσιμος μεγαλώνει ολοένα και περισσότερο στον παγκόσμιο ιστό και γενικά στην καθημερινή ζωή, τόσο δυσκολότερο γίνεται να εντοπίσει κάποιος τα αντικείμενα που τον ενδιαφέρουν και να λάβει τις αποφάσεις που απαιτούνται, δεδομένου ότι ο μέσος άνθρωπος διαθέτει περιορισμένους πόρους για αυτό το σκοπό (ο κυριότερος πόρος είναι ο χρόνος). Η κατάσταση κατά την οποία οι πληροφορίες που είναι διαθέσιμες στον χρήστη είναι τόσες

πολλές, ώστε εκείνος αδυνατεί να τις διαχειριστεί, λέγεται *πληροφοριακή υπερφόρτωση (information overload)* (Bawden and Robinson, 2008).

Η διαδικασία εντοπισμού και επιλογής των αντικειμένων που παρουσιάζουν ενδιαφέρον, εκτός από χρόνο απαιτεί και γνώση η οποία δεν είναι πάντα εύκολο να βρεθεί. Έτσι, ιστορικά οι άνθρωποι βασίζονται σε συστάσεις και αναφορές άλλων ανθρώπων ειδικών ή μη, ώστε να πάρουν αποφάσεις και να ανακαλύψουν νέα αντικείμενα που τους ενδιαφέρουν. Όμως η διαδικασία αυτή εύκολα φτάνει στα όρια της, όσο στενότερος είναι ο κύκλος των γνωστών και ειδικών στους οποίους βασιζόμαστε.

Τα Συστήματα Συστάσεων (Recommender Systems – RS) αποτελούν μια λύση που μπορεί να αντιμετωπίσει ως ένα βαθμό τα προαναφερθέντα προβλήματα. Πρόκειται για εργαλεία λογισμικού τα οποία εντάσσονται στο επιστημονικό πεδίο της Ανάκτησης Πληροφορίας (Information Retrieval) και χρησιμοποιούνται για να προτείνουν αντικείμενα που μπορούν να φανούν χρήσιμα ή ενδιαφέροντα στους χρήστες τους. Για να παράγουν τις συστάσεις, φιλτράρουν την διαθέσιμη πληροφορία σχετικά με τα αντικείμενα και λαμβάνουν υπόψη τις προτιμήσεις των άλλων χρηστών. Τα αντικείμενα που προτείνονται μπορεί να είναι οποιουδήποτε τύπου: βιβλία, ταινίες, τραγούδια, άρθρα, ιστοσελίδες, βίντεο και φυσικά καταναλωτικά προϊόντα. Οι συστάσεις βοηθούν τους χρήστες να λάβουν αποφάσεις και έτσι ήδη τα συστήματα συστάσεων παίζουν καθοριστικό ρόλο σε πολλούς τομείς όπως στην ψυχαγωγία με συστάσεις για ταινίες ή μουσικά κομμάτια, στο ηλεκτρονικό εμπόριο με συστάσεις προϊόντων όπως π.χ. υπολογιστές, βιβλία κλπ, καθώς και διαδικτυακές υπηρεσίες με συστάσεις π.χ. για ταξίδια ή ξενοδοχεία.

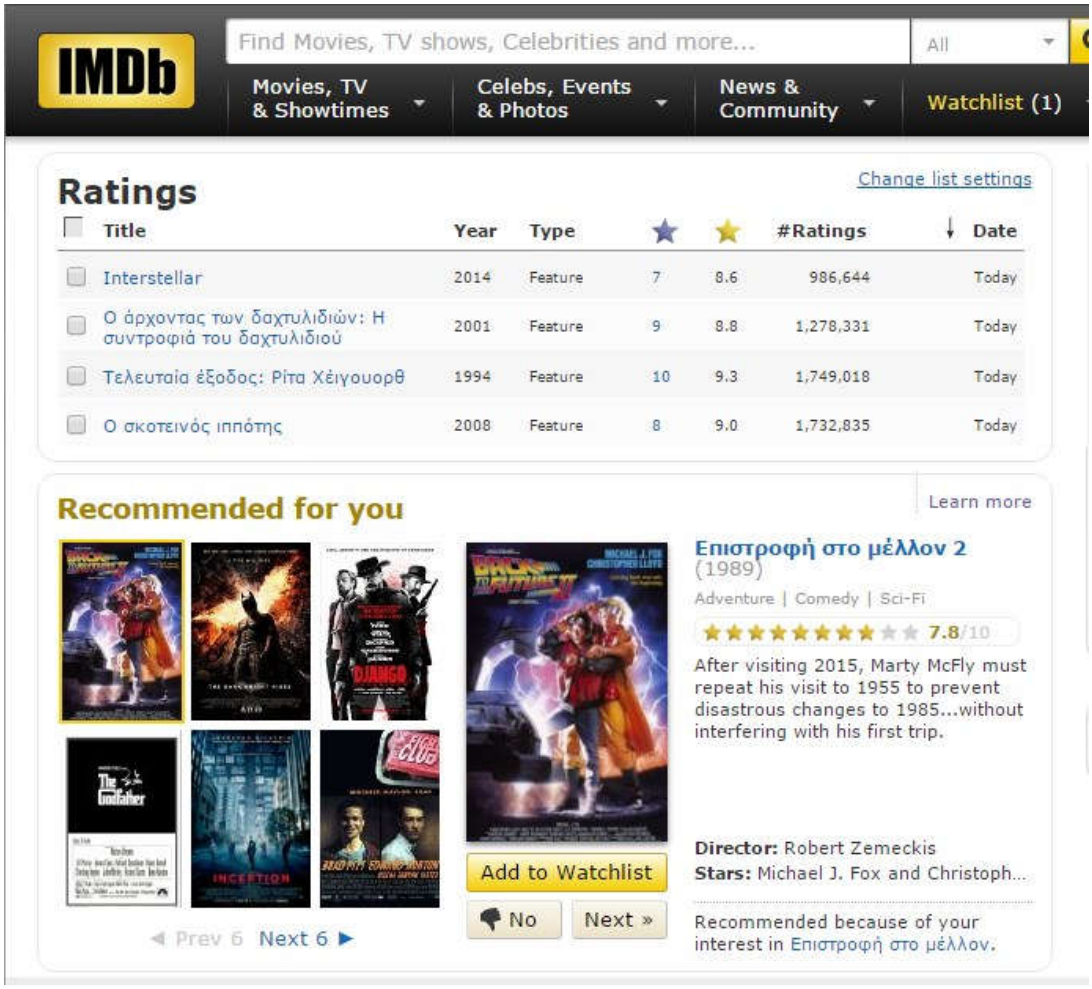
Ένα βασικό χαρακτηριστικό των συστημάτων συστάσεων είναι ότι μπορούν να δώσουν



Εικόνα 1: Προσωποποιημένες προτάσεις με βάση το ιστορικό περιήγησης στο ηλεκτρονικό κατάστημα Amazon.com

εξατομικευμένες συστάσεις (personalization), δηλαδή να προτείνουν σε κάθε χρήστη ξεχωριστά τα αντικείμενα που μπορεί να τον ενδιαφέρουν με βάση τις δικές του προτιμήσεις. Με άλλα λόγια προσπαθούν να «χτίσουν» ένα προφίλ για το κάθε χρήστη. Για τη δημιουργία του προφίλ χρήστη μπορεί να χρησιμοποιηθεί δύο ειδών πληροφορία: η άμεση (explicit) και η έμμεση (implicit).

Στην περίπτωση της έμμεσης πληροφορίας το σύστημα αποθηκεύει διάφορα στοιχεία σχετικά με την αλληλεπίδραση των χρηστών με αυτό και στη συνέχεια τα επεξεργάζεται κατάλληλα ώστε να συνάγει πληροφορίες για τις προτιμήσεις τους. Τέτοιου είδους στοιχεία μπορεί να είναι (Hu et al., 2008): το ιστορικό αγορών του χρήστη, το ιστορικό πλοήγησης ιστοσελίδων, αναζητήσεων ή ακόμα και οι κινήσεις του ποντικιού και ο χρόνος παραμονής σε κάποια ιστοσελίδα. Για παράδειγμα στην Εικόνα 1 φαίνονται οι εξατομικευμένες συστάσεις για αγορά προϊόντων που κάνει το σύστημα συστάσεων του ηλεκτρονικού καταστήματος της Amazon. Οι συστάσεις βασίζονται στο ιστορικό περιήγησης του χρήστη στο κατάστημα, δηλαδή ποια είναι τα προϊόντα τα οποία είδε πρόσφατα.



Εικόνα 2: Συστάσεις ταινιών με βάση τις αξιολογήσεις του χρήστη στον ιστότοπο IMDB

Η άμεση πληροφορία είναι αυτή που παρέχεται απευθείας από τους ίδιους τους χρήστες και αφορά την προτίμηση τους για κάποιο αντικείμενο, εκφρασμένη συνήθως με τη μορφή κάποιας κλίμακας αξιολόγησης π.χ. με αριθμούς 1 έως 5 ή 1 έως 10 ή ακόμα και δυαδικής μορφής «μου αρέσει/δεν μου αρέσει» όπως για παράδειγμα στον ιστότοπο Youtube[1]. Στην Εικόνα 2 φαίνονται οι συστάσεις ταινιών στον ιστότοπο IMDB[2] (Internet Movie DataBase). Οι συστάσεις βασίζονται στις αξιολογήσεις κάποιων ταινιών που έκανε ο χρήστης σε δεκαβάθμια (1-10) αξιολογική κλίμακα.

Η παραγωγή συστάσεων μπορεί να γίνει με πολλούς τρόπους και για το σκοπό αυτό έχουν αναπτυχθεί πολλοί αλγόριθμοι, οι οποίοι χρησιμοποιούνται ανάλογα με τις ανάγκες και τις ιδιορρυθμίες του κάθε συστήματος. Τα πιο διαδεδομένα συστήματα συστάσεων είναι τα *συνεργατικά,* τα οποία αποτελούν και αντικείμενο μελέτης της παρούσας εργασίας. Ο όρος συνεργατικά προέρχεται από το ότι τα συστήματα αυτά αξιοποιούν τα δεδομένα που παρέχουν οι ίδιοι οι χρήστες για να παράγουν τις συστάσεις. Η προσέγγιση αυτή προέκυψε από την τάση των ίδιων των ανθρώπων να ζητούν την βοήθεια/γνώμη των ειδικών ή/και φίλων τους όταν έχουν κάποιο πρόβλημα ή σκοπεύουν να πάρουν κάποια απόφαση. Με βάση αυτή την παρατήρηση το πρώτο σύστημα συστάσεων κατασκευάστηκε το 1992 και ονομαζόταν Tapestry (Goldberg et al., 1992). Το Tapestry προσπαθούσε να ομαδοποιήσει του χρήστες με βάση τα ενδιαφέροντα τους και να προτείνει λίστες ηλεκτρονικού ταχυδρομείου που να τους ενδιαφέρουν.

1.2 Αντικείμενο διπλωματικής

Η παρούσα διπλωματική εργασία αποτελεί μια συγκριτική μελέτη αλγορίθμων για Συστήματα Συστάσεων, όσον αφορά την απόδοση τους σε δημόσια διαθέσιμες συλλογές δεδομένων.

Οι δύο κύριες προσεγγίσεις στην κατασκευή Συστημάτων Συστάσεων είναι η το Φιλτράρισμα με βάση το περιεχόμενο (Content-based filtering) και το Συνεργατικό Φιλτράρισμα (Collaborative filtering). Στην πρώτη περίπτωση χρησιμοποιούνται χαρακτηριστικά (features) που προέρχονται από το γνωστικό πεδίο του τομέα εφαρμογής του συστήματος. Τα χαρακτηριστικά αυτά μπορεί να αφορούν το χρήστη (ηλικία, φύλο, επάγγελμα κλπ) ή τα αντικείμενα (είδος, σκηνοθέτης, ηθοποιοί κλπ αν π.χ. πρόκειται για ταινίες).

[1] https://www.youtube.com/

[2] http://www.imdb.com/

Στην περίπτωση του Συνεργατικού Φιλτραρίσματος δεν χρειάζεται καμία πληροφορία σχετικά με τους χρήστες ή τα αντικείμενα, παρά μόνο ένα σύνολο δεδομένων αξιολόγησης που δηλώνει τις προτιμήσεις κάθε χρήστη για κάποια αντικείμενα. Συνήθως όμως, κάθε χρήστης δηλώνει τις προτιμήσεις του μόνο για ένα μικρό ποσοστό των διαθέσιμων αντικειμένων, με αποτέλεσμα το σύνολο δεδομένων των αξιολογήσεων να είναι εξαιρετικά αραιό (sparse).

Η συγκριτική μελέτη που περιγράφεται στην παρούσα εργασία, επικεντρώνεται στην προσέγγιση του Συνεργατικού Φιλτραρίσματος. Για την επιλογή αυτή συνετέλεσαν αρκετοί παράγοντες. Καταρχήν και για τις δύο προσεγγίσεις υπάρχει μια μεγάλη πληθώρα διαθέσιμων μεθόδων, οπότε μια συγκριτική μελέτη αλγορίθμων και για τις δύο προσεγγίσεις θα ξέφευγε από τα όρια μιας διπλωματικής εργασίας. Επιπλέον, καθώς οι μέθοδοι με βάση το περιεχόμενο βασίζονται σε γνώση του χώρου εφαρμογής, κάθε προσέγγιση αυτού του είδους δεν μπορεί εύκολα να μεταφερθεί σε άλλο χώρο εφαρμογής. Αντίθετα, οι μέθοδοι Συνεργατικού Φιλτραρίσματος μπορούν να εφαρμοστούν σε δεδομένα αξιολογήσεων ανεξαρτήτως του χώρου από τον οποίο προέρχονται.

Ένας άλλος λόγος που συνετέλεσε στην επιλογή του Συνεργατικού Φιλτραρίσματος για την συγκριτική μελέτη είναι ότι είναι πολύ διαδεδομένη προσέγγιση και υπάρχουν αρκετές δημόσια διαθέσιμες συλλογές με δεδομένα αξιολογήσεων. Αυτό δεν ισχύει για τις μεθόδους με βάση το περιεχόμενο, καθώς αφενός η συγκέντρωση των χαρακτηριστικών όλων των αντικειμένων είναι δύσκολη και χρονοβόρα εργασία και αφετέρου δεν είναι εύκολο να πείσεις τους χρήστες να μοιραστούν προσωπικά δεδομένα.

Τα συστήματα συστάσεων Συνεργατικού Φιλτραρίσματος έχουν τυπικά δύο εφαρμογές (tasks). Η πρώτη είναι η *πρόβλεψη αξιολογήσεων (rating prediction)*, δηλαδή η πρόβλεψη της αξιολόγησης που θα έδινε κάποιος χρήστης σε ένα αντικείμενο που δεν γνωρίζει ή δεν έχει καταναλώσει. Η δεύτερη εφαρμογή είναι η *σύσταση των Ν καλύτερων αντικειμένων (top-N recommendation)* που το σύστημα θεωρεί ότι θα ταιριάζουν με τα ενδιαφέροντα του χρήστη.

Στην συγκεκριμένη συγκριτική μελέτη αξιολογείται η απόδοση μεθόδων Συνεργατικού Φιλτραρίσματος για πρόβλεψη αξιολογήσεων. Η πρόβλεψη αξιολογήσεων επιλέχθηκε ως η πιο γενική εφαρμογή, καθώς από τις προβλέψεις αξιολογήσεων μπορεί εύκολα να κατασκευαστεί μια λίστα Ν καλύτερων αντικειμένων για σύσταση, ως επιπρόσθετο βήμα. Βέβαια, αλγόριθμοι που είναι κατασκευασμένοι ειδικά για top-N recommendation, πιθανόν να πετυχαίνουν καλύτερη ακρίβεια σε αυτή την εφαρμογή, αλλά μια συγκριτική μελέτη τέτοιων αλγορίθμων μπορεί να θεωρηθεί ως μελλοντική επέκταση της παρούσας εργασίας.

Ένα κίνητρο για την διεξαγωγή μιας συγκριτικής μελέτης διαφόρων υπαρχόντων μεθόδων Συνεργατικού Φιλτραρίσματος είναι ότι δεν υπάρχει κάποιο καθορισμένο πλαίσιο και μεθοδολογία αξιολόγησης συστημάτων συστάσεων, με αποτέλεσμα πολλές ερευνητικές

εργασίες οι οποίες προτείνουν νέες μεθόδους να παρουσιάζουν πειραματικά αποτελέσματα τα οποία δεν είναι συγκρίσιμα με τα αποτελέσματα άλλων εργασιών. Αυτό οφείλεται σε διάφορους παράγοντες όπως διαφορετικό τρόπο προεπεξεργασίας των δεδομένων, διαφορετικούς τρόπους δειγματοληψίας των δεδομένων, μη παροχή κάποιων λεπτομερειών για την υλοποίηση των αλγορίθμων, χρήση διαφορετικών μετρικών απόδοσης κλπ. Όλα αυτά καθιστούν δύσκολη την *αναπαραγωγιμότητα (reproducibility)* των πειραματικών αποτελεσμάτων και κατά συνέπεια την σύγκριση της απόδοσης μεταξύ των αλγορίθμων.

Ωστόσο, σχετικές εργασίες που έχουν ως αντικείμενο την συγκριτική μελέτη απόδοσης αλγορίθμων συνεργατικού φιλτραρίσματος είναι σχετικά δυσεύρετες. Επιπλέον, οι περισσότερες από αυτές δεν είναι αρκετά εκτενείς, δηλαδή συγκρίνουν λίγες μεθόδους, χρησιμοποιώντας επίσης περιορισμένο αριθμό συλλογών δεδομένων.

1.2.1 Συνεισφορά της διπλωματικής εργασίας

Στην παρούσα διπλωματική εργασία αναπτύχθηκε μια μεθοδολογία αξιολόγησης και σύγκρισης μεθόδων Συνεργατικού Φιλτραρίσματος για πρόβλεψη αξιολογήσεων χρηστών σε αντικείμενα. Συγκεκριμένα, η συνεισφορά της διπλωματικής συνοψίζεται ως εξής:

1. Αναπτύξαμε μια μεθοδολογία αξιολόγησης συστημάτων για πρόβλεψη αξιολογήσεων με βάση τις δημοφιλέστερες μεθοδολογίες και μετρικές που προτείνονται στη βιβλιογραφία.
2. Δόθηκε ιδιαίτερη σημασία στην αναπαραγωγιμότητα των αποτελεσμάτων, ορίζοντας συγκεκριμένες παραμέτρους όπου χρειαζόταν και εκτελώντας τα πειράματα με συγκεκριμένα random seeds.
3. Συγκρίναμε συνολικά 16 μεθόδους πρόβλεψης αξιολογήσεων, απλές αλλά και state-of-the-art. Συγκεκριμένα αξιολογήθηκαν 4 βασικοί εκτιμητές, 4 προσεγγίσεις βασισμένες στη μνήμη και 4 παραλλαγές αυτών και 4 μέθοδοι βασισμένες σε μοντέλο.
4. Η αξιολόγηση έγινε σε 4 δημόσια διαθέσιμες συλλογές δεδομένων, προερχόμενες από 3 διαφορετικούς τομείς εφαρμογής και με διαφορετικούς βαθμούς αραιότητας.

1.3 Οργάνωση κειμένου

Στο Κεφάλαιο 2 γίνεται μια επισκόπηση μεθόδων μηχανικής μάθησης που χρησιμοποιούνται στα πλαίσια των συστημάτων συστάσεων, περιγράφονται οι βασικές έννοιες, οι κατηγορίες συστημάτων συστάσεων καθώς και τα βασικότερα προβλήματα και προκλήσεις που

παρουσιάζονται κατά την κατασκευή τους. Γίνεται ιδιαίτερη αναφορά στις μεθόδους μείωσης διαστατικότητας καθώς αποτελούν την βάση κάποιων state-of-the-art μεθόδων παραγωγής συστάσεων. Στο Κεφάλαιο 3 παρουσιάζεται η μοντελοποίηση και ο ορισμός του προβλήματος της πρόβλεψης αξιολογήσεων και στη συνέχεια αναλύονται οι που συμμετέχουν στην συγκριτική μελέτη. Το Κεφάλαιο 4 αναπτύσσει μια μεθοδολογία αξιολόγησης συστημάτων πρόβλεψης αξιολογήσεων, τον τρόπο οργάνωσης των πειραμάτων και τέλος παρουσιάζει τα αποτελέσματα των πειραμάτων. Στο Κεφάλαιο 5 αναφέρονται τεχνικές λεπτομέρειες σχετικά με την διεξαγωγή των πειραμάτων, καθώς και τα προγραμματιστικά εργαλεία που χρησιμοποιήθηκαν στην υλοποίηση. Τέλος, το Κεφάλαιο 6 συνοψίζει τα αποτελέσματα και παρουσιάζει τα συμπεράσματα και κάποιες μελλοντικές επεκτάσεις της συγκριτικής μελέτης.

2

Θεωρητικό υπόβαθρο και σχετικές εργασίες

2.1 Μηχανική Μάθηση

Η Μηχανική Μάθηση είναι ένας κλάδος της Επιστήμης Υπολογιστών που διερευνά τη μελέτη και κατασκευή αλγορίθμων που μπορούν να «μαθαίνουν» από τα δεδομένα και στη συνέχεια να κάνουν προβλέψεις σχετικά με αυτά. Ένας πιο επίσημος ορισμός προτάθηκε από τον Tom M. Mitchell (Mitchell, 1997): «Ένα πρόγραμμα υπολογιστή λέγεται ότι μαθαίνει από εμπειρία E ως προς μια κλάση εργασιών T και ένα μέτρο απόδοσης P, αν η απόδοση του σε εργασίες της κλάσης T, όπως αποτιμάται από το μέτρο P, βελτιώνεται με την εμπειρία E» (σελ. 2).

2.1.1 Κατηγορίες προβλημάτων

Ανάλογα με την φύση του προβλήματος και την προσέγγιση που ακολουθείται, οι αλγόριθμοι μηχανικής μάθησης ταξινομούνται σε τρεις μεγάλες κατηγορίες:

- *Μάθηση με επίβλεψη (Supervised Learning)*: Στον αλγόριθμο δίνονται τόσο τα δεδομένα εισόδου (υποδείγματα) με τα οποία θα εκπαιδευτεί, όσο και οι αντίστοιχες

επιθυμητές έξοδοι. Ο αλγόριθμος προσπαθεί να μάθει ένα γενικό κανόνα ώστε να μπορεί να αντιστοιχήσει τις εισόδους με τις εξόδους.

- *Μάθηση χωρίς επίβλεψη (Unsupervised Learning)*: Ο αλγόριθμος μάθησης δέχεται μόνο τα δεδομένα εισόδου και έχει ως στόχο να ανακαλύψει αν τα δεδομένα έχουν κάποια δομή, κρυφά χαρακτηριστικά ή εμφανίζουν κάποια μοτίβα.
- *Ενισχυτική μάθηση (Reinforcement Learning)*: Το σύστημα μαθαίνει δια μέσου της αλληλεπίδρασης του με το περιβάλλον, προσπαθώντας να βελτιστοποιήσει την απόδοση του με αυτόματο τρόπο.

Μια ακόμα κατηγορία, η οποία όμως αποτελεί συνδυασμό της μάθησης με επίβλεψη και της μάθησης χωρίς επίβλεψη, είναι η *μάθηση με ημι-επίβλεψη (Semi-supervised Learning)*. Στην περίπτωση αυτή, κάποια από τα δεδομένα εισόδου διαθέτουν τις αντίστοιχες επιθυμητές εξόδους, ενώ τα υπόλοιπα, που συνήθως είναι και περισσότερα, δεν διαθέτουν εξόδους.

Μια άλλη κατηγοριοποίηση των προβλημάτων με τα οποία ασχολείται η μηχανική μάθηση, μπορεί να γίνει ανάλογα με το αποτέλεσμα του αλγορίθμου μηχανικής μάθησης (Goodfellow et al., 2016):

Ταξινόμηση (Classification): Τα δεδομένα εισόδου χωρίζονται σε δύο τουλάχιστον κλάσεις (ομάδες) και ο αλγόριθμος κατασκευάζει ένα μοντέλο με βάση αυτά τα δεδομένα, το οποίο προσπαθεί να εντάξει κάποια δεδομένα σε μία από τις κλάσεις. Η ταξινόμηση συνήθως αποτελεί πρόβλημα μάθησης με επίβλεψη. Ένα παράδειγμα ταξινόμησης είναι τα φίλτρα Spam, όπου ο αλγόριθμος επεξεργάζεται e-mails και προσπαθεί να τα διαχωρίσει σε δύο κλάσεις: “spam” και “όχι spam”. Επίσης, η ταξινόμηση έχει εφαρμογή στην αναγνώριση προτύπων (pattern recognition), όπως για παράδειγμα στην αναγνώριση αντικειμένων σε εικόνες, αναγνώριση κειμένου κλπ.

Παλινδρόμηση (Regression): Η παλινδρόμηση μοιάζει με την ταξινόμηση, με την διαφορά ότι ο αλγόριθμος προσπαθεί να προβλέψει κάποια αριθμητική τιμή αντί για κλάση. Ουσιαστικά στην παλινδρόμηση το μοντέλο μαθαίνει μια συνάρτηση την οποία στη συνέχεια χρησιμοποιεί για να προβλέψει το αποτέλεσμα για κάποια δεδομένα εισόδου.

Συσταδοποίηση (Clustering): Κατά την συσταδοποίηση ο αλγόριθμος προσπαθεί να ομαδοποιήσει τα δεδομένα εισόδου σε συστάδες (clusters) με βάση κάποιο κριτήριο ομοιότητας. Ο αριθμός και το μέγεθος των συστάδων δεν είναι γνωστός. Η συσταδοποίηση αποτελεί μια τυπική εφαρμογή μάθησης χωρίς επίβλεψη.

Μείωση διαστατικότητας (Dimensionality Reduction): Τα πολυδιάστατα δεδομένα παρουσιάζουν δυσκολίες στην επεξεργασία τους. Οι αλγόριθμοι μείωσης διαστατικότητας προσπαθούν να απλοποιήσουν τα πολυδιάστατα δεδομένα αντιστοιχίζοντας τα σε χώρο λιγότερων διαστάσεων, με τέτοιο τρόπο ώστε να διατηρείται όσο το δυνατόν περισσότερη από την αρχική πληροφορία.

Συμπλήρωση Ελλειπουσών τιμών (Missing Data Imputation): Σε αυτή την περίπτωση, σε ένα αλγόριθμο δίνονται δεδομένα στα οποία οι τιμές κάποιων χαρακτηριστικών λείπουν, δηλαδή είναι άγνωστες. Σκοπός του αλγορίθμου είναι να προβλέψει τις τιμές που λείπουν, βάσει των δοθέντων δεδομένων.

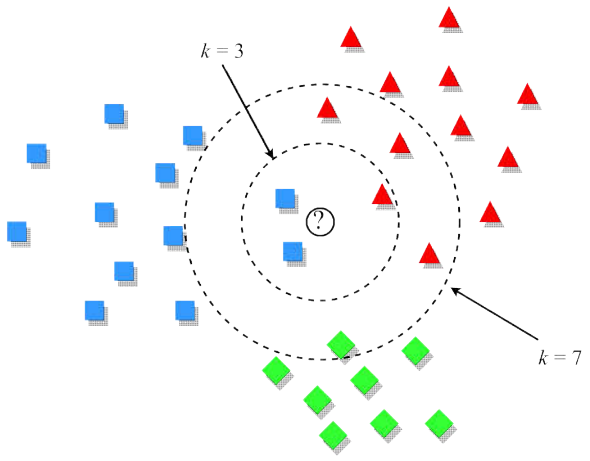
Εκτίμηση πυκνότητας ή κατανομής πιθανότητας (Density or Probability Estimation): Ο αλγόριθμος προσπαθεί να μάθει την δομή των δεδομένων εισόδου, με σκοπό να δώσει μια εκτίμηση της στατιστικής κατανομής από την οποία προήλθαν τα δεδομένα. Η εκτίμηση της κατανομής από την οποία προήλθαν τα δεδομένα μπορεί να βοηθήσει σημαντικά και σε άλλες εργασίες όπως στην Συμπλήρωση Ελλειπουσών τιμών που αναφέρθηκε παραπάνω.

2.1.2 Κατηγορίες αλγορίθμων

2.1.2.1 k - Πλησιέστεροι Γείτονες (k – Nearest Neighbors – kNN)

Η μέθοδος των πλησιέστερων γειτόνων αποτελεί μια από τις απλούστερες τεχνικές που μπορούν να χρησιμοποιηθούν για ταξινόμηση και παλινδρόμηση. Ουσιαστικά ο αλγόριθμος δεν «μαθαίνει» από τα δεδομένα, αλλά περιμένει ένα νέο πρότυπο προς ταξινόμηση και εντοπίζει τα k πιο όμοια πρότυπα εκπαίδευσης που έχει στη διάθεση του με βάση κάποια μετρική ομοιότητας. Στη συνέχεια, αν το πρότυπο πρέπει να ταξινομηθεί, τότε αντιστοιχείται στην κλάση στην οποία ανήκει η πλειοψηφία των κοντινότερων γειτόνων του. Αν πρόκειται για παλινδρόμηση, το αποτέλεσμα είναι ο μέσος όρος των τιμών των πλησιέστερων γειτόνων.

Μια συνηθισμένη μετρική ομοιότητας είναι η Ευκλείδεια απόσταση, αλλά επίσης χρησιμοποιούνται και άλλες όπως η απόσταση Hamming, καθώς και άλλες μετρικές όπως ο συντελεστής συσχέτισης του Pearson.

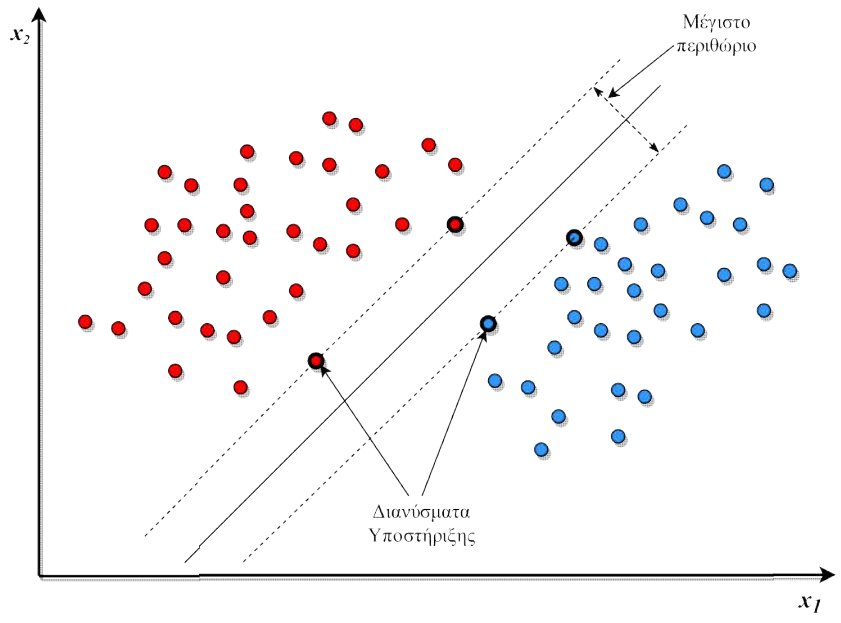


Σχήμα 1. Παράδειγμα ταξινόμησης k-NN. Το νέο πρότυπο (κύκλος) θα ταξινομηθεί σε μία κλάση με βάση τους κοντινότερους γείτονες του

Πέρα από την επιλογή της μετρικής ομοιότητας, σημαντική είναι η επιλογή της παραμέτρου k, δηλαδή του μεγέθους της γειτονιάς. Η βέλτιστη τιμή k εξαρτάται από τα δεδομένα και συνήθως επιλέγεται με ευρετικές τεχνικές, όπως τεχνικές *βελτιστοποίησης υπερπαραμέτρων (hyperparameter optimization)*. Παρόλα αυτά, η γενική ιδέα είναι το k να μην είναι πολύ μικρό γιατί τότε η τεχνική είναι ευαίσθητη στα δεδομένα που αποτελούν θόρυβο, αλλά ούτε και πολύ μεγάλο γιατί η γειτονιά μπορεί να συμπεριλάβει πρότυπα από γειτονικές κλάσεις. Για παράδειγμα στο Σχήμα 1 φαίνονται τρεις κλάσεις και το νέο πρότυπο (κύκλος) θα ταξινομηθεί σε μία από τις τρεις. Ανάλογα με την επιλεγμένη τιμή *k* μπορεί να ανήκει στην κλάση των τετραγώνων (*k* = 3) ή στην κλάση των τριγώνων (*k* = 7).

2.1.2.2 Μηχανές Διανυσμάτων Υποστήριξης (Support Vector Machines)

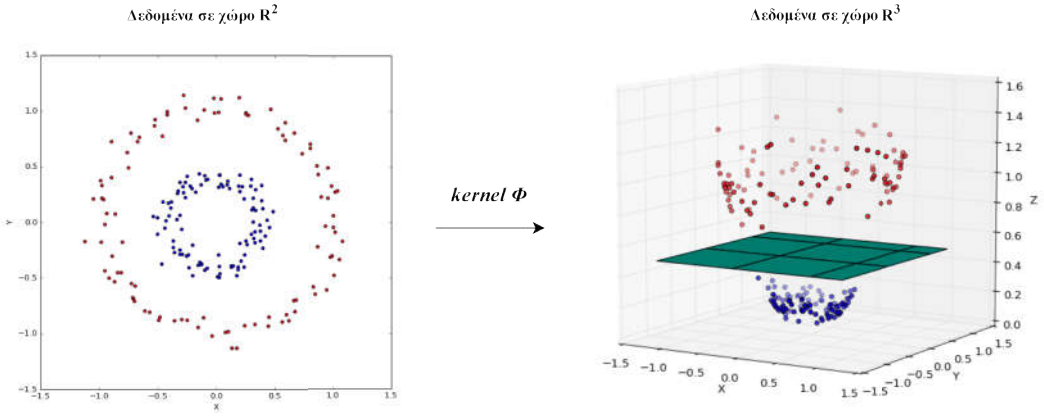
Οι Μηχανές Διανυσμάτων Υποστήριξης είναι μοντέλα μάθησης με επίβλεψη που μπορούν να χρησιμοποιηθούν για ταξινόμηση και παλινδρόμηση. Στην περίπτωση της ταξινόμησης, ο αλγόριθμος μάθησης του μοντέλου προσπαθεί να βρει το βέλτιστο υπερεπίπεδο που διαχωρίζει τα δεδομένα σε κλάσεις. Η εύρεση του βέλτιστου υπερεπιπέδου σημαίνει ότι αφήνει το μεγαλύτερο δυνατό περιθώριο ανάμεσα στις διαχωριζόμενες κλάσεις και αυτό είναι σημαντικό για την ελαχιστοποίηση της πιθανότητας εσφαλμένης ταξινόμησης νέων δειγμάτων, δηλαδή ενισχύει την ικανότητα γενίκευσης του μοντέλου. Διανύσματα Υποστήριξης ονομάζονται τα δεδομένα που βρίσκονται πιο κοντά στο διαχωριστικό υπερεπίπεδο, δηλαδή είναι αυτά που είναι πιο δύσκολο να ταξινομηθούν σωστά (Cortes and Vapnik, 1995).



Σχήμα 2. Οι Μηχανές Διανυσμάτων Υποστήριξης βρίσκουν το υπερεπίπεδο (εδώ ευθεία) που διαχωρίζει τις κλάσεις μεγιστοποιώντας το περιθώριο ανάμεσα τους

Στην περίπτωση που τα δεδομένα δεν είναι γραμμικά διαχωρίσιμα, μπορούν να χρησιμοποιηθούν επιπλέον *μεταβλητές χαλαρότητας (slack variables)*, οι οποίες προσδίδουν κάποια ανοχή στα όρια του διαχωριστικού υπερεπιπέδου.

Μια άλλη προσέγγιση για την αντιμετώπιση μη-γραμμικά διαχωρίσιμων δεδομένων είναι να γίνει προβολή των αρχικών δεδομένων σε ένα χώρο υψηλότερων διαστάσεων στον οποίο μπορεί να είναι πλέον γραμμικά διαχωρίσιμα (Σχήμα 3). Ο μετασχηματισμός αυτός είναι γνωστός ως *«τέχνασμα πυρήνα» (kernel trick)* και πραγματοποιείται με την βοήθεια των *συναρτήσεων πυρήνα (kernel functions)*, εκ των οποίων η γνωστότερη είναι η *συνάρτηση ακτινικής βάσης (Radial Basis Function – RBF)*. Επίσης μπορεί να χρησιμοποιηθεί και ο πολυωνυμικός πυρήνας που βασίζεται σε πολυωνυμικές συναρτήσεις $2^{ου}$ ή μεγαλύτερου βαθμού.



Σχήμα 3. Μετασχηματισμός μη-γραμμικά διαχωρίσιμων δεδομένων σε χώρο υψηλότερης διάστασης με την βοήθεια πολυωνυμικού πυρήνα 2ου βαθμού

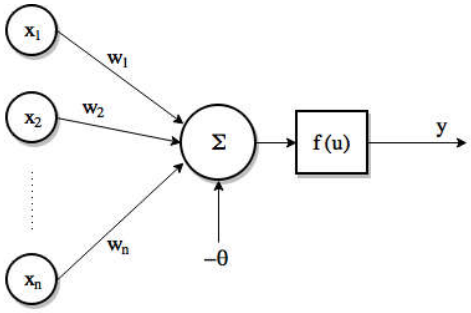
Οι Μηχανές Διανυσμάτων Υποστήριξης κατασκευάζουν μοντέλα με μεγάλη ακρίβεια ταξινόμησης και μεγάλη δύναμη γενίκευσης. Από την άλλη όμως, η διαδικασία μάθησης είναι υπολογιστικά δαπανηρή σε μεγάλα σύνολα δεδομένων, ιδιαίτερα με τη χρήση μη-γραμμικών πυρήνων.

2.1.2.3 Τεχνητά Νευρωνικά Δίκτυα

Τα Τεχνητά Νευρωνικά Δίκτυα - ΤΝΔ (Artificial Neural Networks - ANN) είναι δίκτυα που αποτελούνται από απλούς υπολογιστικούς κόμβους (νευρώνες), διασυνδεδεμένους μεταξύ τους. Οι κόμβοι του ονομάζονται νευρώνες σε αντιστοιχία με τους βιολογικούς νευρώνες και η δομή και λειτουργία τους μοιάζει με την λειτουργία του βιολογικού εγκεφάλου.

Κάθε νευρώνας δέχεται ένα σύνολο αριθμητικών εισόδων, επιτελεί έναν υπολογισμό με βάση αυτές τις εισόδους και παράγει μία έξοδο. Η έξοδος μπορεί να τροφοδοτηθεί ως είσοδος σε άλλους νευρώνες του δικτύου ή να κατευθυνθεί στο περιβάλλον.

Η πιο απλή περίπτωση ΤΝΔ είναι το Perceptron (Rosenblatt, 1958), το οποίο αποτελείται από ένα μόνο νευρώνα. Ο νευρώνας (Σχήμα 4) αποτελείται από τρία στοιχεία: ένα σύνολο από n συνάψεις (που αντιστοιχούν στις εισόδους), ένα αθροιστή και μια συνάρτηση ενεργοποίησης.



Σχήμα 4. Μοντέλο τεχνητού νευρώνα του Perceptron

Κάθε σύναψη δέχεται μια είσοδο x και χαρακτηρίζεται από μια τιμή βάρους w. Έτσι αν ο νευρώνας αποτελείται από n συνάψεις, δέχεται ένα διάνυσμα εισόδων $\boldsymbol{x} = [x_1, x_2, \ldots . x_n]$ και τα συναπτικά βάρη μπορούν να αναπαρασταθούν από ένα διάνυσμα $\boldsymbol{w} = [w_1, w_2, \ldots . w_n]$. Ο αθροιστής αθροίζει τις τιμές εισόδου πολλαπλασιασμένες με τα αντίστοιχα βάρη των συνάψεων. Επίσης, ορίζεται μια παράμετρος κατωφλίου θ, η οποία λέγεται κατώφλι ενεργοποίησης και μια συνάρτηση ενεργοποίησης $f(.)$ η οποία δίνει την έξοδο του νευρώνα. Η συνάρτηση ενεργοποίησης $f(.)$ είναι η βηματική συνάρτηση 0/1 (Εξ. (2.1)) ή η διπολική βηματική συνάρτηση -1/1 (Εξ. (2.2)).

$$f(u) = \begin{cases} 1, & \alpha\nu\ u > 0 \\ 0, & \alpha\nu\ u \leq 0 \end{cases} \tag{2.1}$$

$$f(u) = \begin{cases} 1, & \alpha\nu\ u > 0 \\ -1, & \alpha\nu\ u \leq 0 \end{cases} \tag{2.2}$$

Η τελική έξοδος του νευρώνα δίνεται από τη σχέση:

$$y = f\left(\sum_{i=1}^{n} w_i x_i - \theta\right) \tag{2.3}$$

Το απλό perceptron μπορεί να εκπαιδευτεί ώστε να διαχωρίζει δεδομένα που ανήκουν σε γραμμικά διαχωρίσιμες κλάσεις.

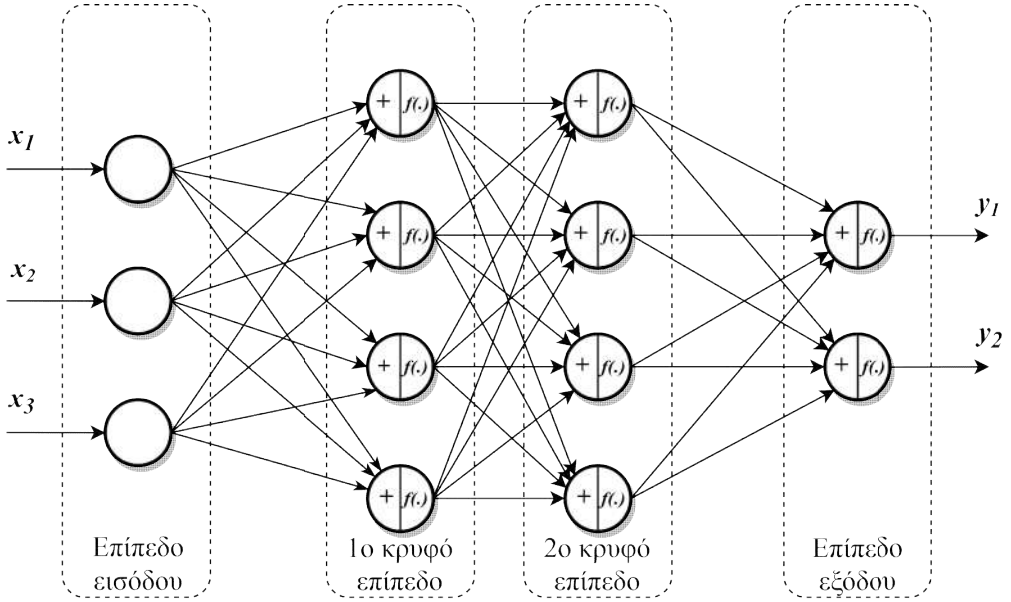
Σε ένα δίκτυο μπορούν να χρησιμοποιηθούν περισσότεροι νευρώνες, τοποθετημένοι σε διαφορετικά επίπεδα και κατάλληλα συνδεδεμένοι μεταξύ τους. Ανάλογα με την αρχιτεκτονική που ακολουθείται, τα επίπεδα που περιέχουν νευρώνες διακρίνονται σε τρεις κατηγορίες: το επίπεδο εισόδου, στα κρυφά επίπεδα και στο επίπεδο εξόδου. Οι νευρώνες του επιπέδου εισόδου λαμβάνουν τις εισόδους του δικτύου χωρίς να κάνουν κάποια επεξεργασία. Οι νευρώνες των κρυφών επιπέδων λαμβάνουν τις εισόδους από το προηγούμενο επίπεδο,

επιτελούν την επεξεργασία που περιγράφηκε για τον μεμονωμένο νευρώνα του Perceptron και τροφοδοτούν με την έξοδο τους, τους νευρώνες του επόμενου επιπέδου. Οι νευρώνες του επιπέδου εξόδου συνδυάζουν τα αποτελέσματα του προηγούμενου επιπέδου και παράγουν την τελική έξοδο του δικτύου. Σε τέτοιου είδους αρχιτεκτονικές για λόγους εκπαίδευσης του δικτύου, ως συνάρτηση ενεργοποίησης προτιμάται η *σιγμοειδής συνάρτηση*:

$$f(u) = \frac{1}{1 + e^{-u}}$$

ή η *συνάρτηση υπερβολικής εφαπτομένης*:

$$\tanh(u) = \frac{e^{u} - e^{-u}}{e^{u} + e^{-u}}$$



Σχήμα 5. Αρχιτεκτονική νευρωνικού δικτύου MLP με δύο κρυφά επίπεδα

Αυτή η αρχιτεκτονική νευρωνικών δικτύων που περιγράφηκε ονομάζεται Δίκτυο Perceptron Πολλών Στρωμάτων (Multi-Layer Perceptron – MLP) και έχει την δυνατότητα να προσεγγίζει οποιαδήποτε ομαλή συνάρτηση, δηλαδή μπορεί να διαχωρίσει δεδομένα που ανήκουν σε μη-γραμμικά διαχωρίσιμες κλάσεις. Το Σχήμα 5 δείχνει ένα παράδειγμα δικτύου Perceptron Πολλών Στρωμάτων με τρεις εισόδους, δύο κρυφά επίπεδα με τέσσερις νευρώνες στο καθένα και δύο εξόδους. Σημειώνεται ότι τα δίκτυα MLP με περισσότερα από ένα κρυφά επίπεδα αποτελούν μια αρχιτεκτονική δικτύων που ονομάζεται Βαθιά Νευρωνικά Δίκτυα (Deep Neural Networks) και απαιτούν ειδικές τεχνικές εκπαίδευσης (Goodfellow et al., 2016).

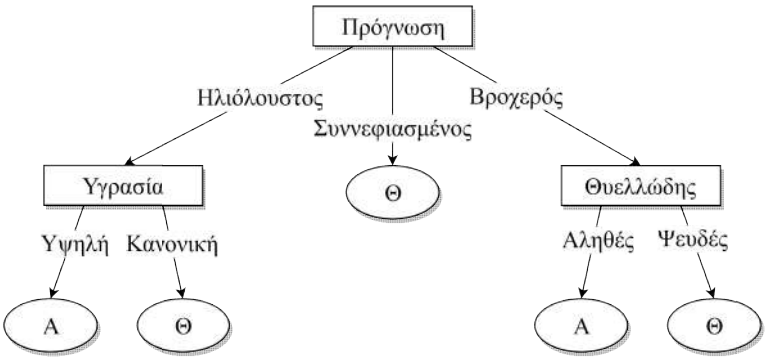
2.1.2.4 Δέντρα αποφάσεων

Τα Δέντρα Αποφάσεων (Decision Trees) ανήκουν στην κατηγορία των ταξινομητών, αλλά μπορούν να χρησιμοποιηθούν και για παλινδρόμηση (Quinlan, 1986). Χρησιμοποιώντας τα διαθέσιμα δεδομένα κατασκευάζεται μια δενδρική δομή, με βάση την οποία μπορεί να αποφασιστεί σε ποια κλάση θα ανήκει ένα νέο αντικείμενο, ακολουθώντας μια διαδρομή μέσα στο δέντρο. Οι κόμβοι του εν λόγω δέντρου μπορεί να είναι είτε κόμβοι απόφασης (όπου ελέγχεται κάποια ιδιότητα του αντικειμένου για να καθοριστεί ποιο κλαδί του δέντρου θα επιλεγεί για το επόμενο βήμα) ή κόμβοι-φύλλα (που καθορίζουν την κλάση στην οποία ανήκει το αντικείμενο). Για την κατασκευή δέντρων αποφάσεων χρησιμοποιούνται διάφοροι αλγόριθμοι, με γνωστότερους τους ID3 (Quinlan, 1986) και C4.5 (Quinlan, 1993).

Πίνακας 1. Ένα σύνολο εκπαίδευσης καιρικών συνθηκών

Πρότυπο	Πρόγνωση	Θερμοκρασία	Υγρασία	Θυελλώδης	Κλάση
1	Ηλιόλουστος	Ζεστός	Υψηλή	Ψευδές	Α
2	Ηλιόλουστος	Ζεστός	Υψηλή	Αληθές	Α
3	Συννεφιασμένος	Ζεστός	Υψηλή	Ψευδές	Θ
4	Βροχερός	Ήπιος	Υψηλή	Ψευδές	Θ
5	Βροχερός	Δροσερός	Κανονική	Ψευδές	Θ
6	Βροχερός	Δροσερός	Κανονική	Αληθές	Α
7	Συννεφιασμένος	Δροσερός	Κανονική	Αληθές	Θ
8	Ηλιόλουστος	Ήπιος	Υψηλή	Ψευδές	Α
9	Ηλιόλουστος	Δροσερός	Κανονική	Ψευδές	Θ
10	Βροχερός	Ήπιος	Κανονική	Ψευδές	Θ
11	Ηλιόλουστος	Ήπιος	Κανονική	Αληθές	Θ
12	Συννεφιασμένος	Ήπιος	Υψηλή	Αληθές	Θ
13	Συννεφιασμένος	Ζεστός	Κανονική	Ψευδές	Θ
14	Βροχερός	Ζεστός	Υψηλή	Αληθές	Α

Ένα παράδειγμα κατασκευής δέντρου απόφασης φαίνεται στο Σχήμα 6. Ο Πίνακας 1 περιέχει τα δεδομένα εκπαίδευσης που χρησιμοποιήθηκαν για την κατασκευή του δέντρου και αναφέρονται σε καιρικές συνθήκες (Quinlan, 1986). Τα διαθέσιμα χαρακτηριστικά του καιρού που λαμβάνονται υπόψη είναι η Πρόγνωση, η Υγρασία, η Θερμοκρασία και αν ο άνεμος είναι Θυελλώδης. Με βάση τις τιμές αυτών των χαρακτηριστικών κάθε πρότυπο αντιστοιχεί σε μία από τις δύο κλάσεις απόφασης διεξαγωγής κάποιας δραστηριότητας: Α – Αρνητική ή Θ – Θετική.



Σχήμα 6. Απλό δέντρο απόφασης για την διεξαγωγή κάποιας δραστηριότητας με βάση τις καιρικές συνθήκες. Οι κλάση Α αντιστοιχεί στην Αρνητική απόφαση, ενώ η Θ στην Θετική απόφαση

Τα πλεονεκτήματα των δέντρων απόφασης είναι η ευκολία κατασκευής τους και η ταχύτητα ταξινόμησης τους. Παρόλα αυτά, η απόδοση τους ως προς την ακρίβεια πρόβλεψης δεν είναι πολύ υψηλή και έχουν την τάση υπερεκπαίδευσης (overfitting) στα δεδομένα εκπαίδευσης, γι' αυτό στην πράξη χρησιμοποιούνται σε συνδυαστικά μοντέλα (ensemble models) στα οποία κατασκευάζονται πολλά δέντρα και οι προβλέψεις τους συνδυάζονται σε μια τελική πρόβλεψη. Ένα από τα γνωστότερα συνδυαστικά μοντέλα με δέντρα απόφασης είναι τα Random Forests (Breiman, 2001).

2.2 Αλγόριθμοι βελτιστοποίησης

Σε πολλά προβλήματα μηχανικής μάθησης είναι απαραίτητο να γίνει βελτιστοποίηση κάποιας συνάρτησης κόστους J, η οποία συνδέεται με την ποιότητα του εκπαιδευόμενου μοντέλου που χρησιμοποιείται. Συνήθως η συνάρτηση κόστους έχει σχέση με την απόκλιση των προβλέψεων του μοντέλου μάθησης από τις πραγματικές τιμές. Μια τυπική συνάρτηση κόστους είναι το *Μέσο Τετραγωνικό Σφάλμα (Mean Squared Error – MSE)*:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(\widehat{\boldsymbol{Y}}_i - \boldsymbol{Y}_i\right)^2 \qquad (2.4)$$

όπου $\widehat{\boldsymbol{Y}}$ είναι ένα διάνυσμα n προβλέψεων και $\boldsymbol{Y}$ είναι το διάνυσμα των πραγματικών τιμών που αντιστοιχούν στις εισόδους της συνάρτησης που παρήγαγε τις προβλέψεις. Ανάλογα με το πρόβλημα και τον αλγόριθμο βελτιστοποίησης που χρησιμοποιείται, μπορεί να χρησιμοποιούνται παραλλαγές του MSE ή και τελείως διαφορετικές συναρτήσεις κόστους. Στη συνέχεια παρουσιάζονται μερικοί δημοφιλείς αλγόριθμοι βελτιστοποίησης.

2.2.1 Κατάβαση δυναμικού (Gradient Descent)

Μια συχνά χρησιμοποιούμενη μέθοδος βελτιστοποίησης παραμέτρων είναι η *κατάβαση δυναμικού (Gradient Descent)*, η οποία μπορεί να βρει την ελάχιστη τιμή μιας συνάρτησης κόστους n μεταβλητών $J(\boldsymbol{w})$ όπου $\boldsymbol{w} = [w_1, w_2, \dots w_n]$. Ο στόχος της μεθόδου είναι να βρει τιμές των μεταβλητών του διανύσματος $\boldsymbol{w}$, τέτοιες ώστε η τιμή της J να είναι η ελάχιστη δυνατή:

$$J_{min} = J(w_1^*, w_2^*, \dots w_n^*) = \min_{w_1, w_2, \dots w_n} J(w_1, w_2, \dots . w_n) \tag{2.5}$$

Ο αλγόριθμος ξεκινάει με τυχαίες αρχικές τιμές των μεταβλητών και στη συνέχεια λειτουργώντας επαναληπτικά, προσπαθεί να βρει το σημείο του ελαχίστου υπολογίζοντας την κλίση της συνάρτησης σε κάθε σημείο και προχωρώντας «καθοδικά», δηλαδή αντίθετα από την κλίση. Η κλίση (gradient) $\boldsymbol{g}$ υπολογίζεται ως η μερική παράγωγος της συνάρτησης κόστους ως προς κάθε μεταβλητή, δηλαδή:

$$\boldsymbol{g} = \left[\frac{\partial J}{\partial w_1}, \frac{\partial J}{\partial w_2}, \dots ., \frac{\partial J}{\partial w_n}\right]$$

Σε κάθε επανάληψη, ο αλγόριθμος μετακινείται κατά $\boldsymbol{\Delta w}$**,** που ορίζεται ως η αρνητική κλίση $\boldsymbol{g}$ πολλαπλασιασμένη με ένα *ρυθμό μάθησης (learning rate)* η:

$$\boldsymbol{\Delta w} = -\eta \boldsymbol{g}$$

Έτσι, τελικά ο κανόνας κατάβασης δυναμικού δίνεται από τη σχέση:

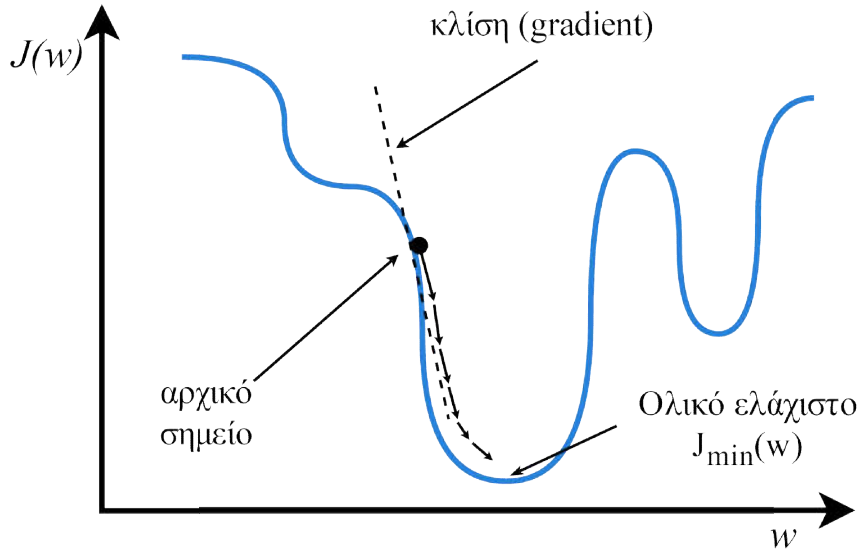
$$\boldsymbol{w'} = \boldsymbol{w} - \eta \boldsymbol{g} \tag{2.6}$$

ή εναλλακτικά η i-οστή μεταβλητή μεταβάλλεται με βάση τη σχέση:

$$w_i' = w_i - \eta \frac{\partial J(w)}{\partial w_i} \tag{2.7}$$

Το $\boldsymbol{w'}$ και w_i' στις σχέσεις (2.6) και (2.7) αντίστοιχα, συμβολίζουν την νέα τιμή της μεταβλητής η οποία θα πάρει τη θέση της παλιάς τιμής στην επόμενη επανάληψη του αλγορίθμου.

Στο Σχήμα 7 απεικονίζεται ένα απλό παράδειγμα κατάβασης δυναμικού σε συνάρτηση κόστους $J(w)$ μιας μεταβλητής. Η γραφική παράσταση της συνάρτησης μοιάζει με την μορφολογία του εδάφους, δηλαδή τα υψηλά σημεία της ονομάζονται *λόφοι* και τα χαμηλά *κοιλάδες*. Παρατηρούμε λοιπόν ότι ο αλγόριθμος οδηγεί στην πλησιέστερη κοιλάδα και τελικά θα συγκλίνει στον πάτο της.



Σχήμα 7. Παράδειγμα εύρεσης ελαχίστου συνάρτησης με κατάβαση δυναμικού (gradient descent)

2.2.2 Μαζική και Στοχαστική κατάβαση δυναμικού

Υπάρχουν δύο τρόποι υπολογισμού των παραμέτρων *w* της συνάρτησης κόστους. Ο πρώτος τρόπος είναι ο αλγόριθμος να ανατρέξει σε όλα τα πρότυπα εκπαίδευσης, πριν κάνει ένα βήμα διόρθωσης των τιμών των παραμέτρων μέσω της συνάρτησης κόστους. Αυτή η τεχνική ονομάζεται *μαζική κατάβαση δυναμικού (Batch Gradient Descent).*

Στη δεύτερη περίπτωση, οι παράμετροι *w* διορθώνονται για κάθε πρότυπο εκπαίδευσης που συναντά ο αλγόριθμος, δηλαδή η διόρθωση γίνεται αυξητικά. Η τεχνική αυτή ονομάζεται *Στοχαστική κατάβαση δυναμικού (Stochastic Gradient Descent – SGD)*, αλλιώς γνωστή και ως *Αυξητική κατάβαση δυναμικού (Incremental Gradient Descent)* και αποτελεί μια στοχαστική προσέγγιση της Μαζικής κατάβασης δυναμικού.

Η Στοχαστική κατάβαση δυναμικού έχει το πλεονέκτημα ότι συγκλίνει γρηγορότερα σε ελάχιστο, αλλά όπως και η Μαζική κατάβαση δυναμικού δεν εγγυάται ότι θα βρει το ολικό ελάχιστο. Παρόλα αυτά, προτιμάται ως μέθοδος βελτιστοποίησης όταν υπάρχει μεγάλος όγκος δεδομένων εκπαίδευσης.

Μια ενδιάμεση προσέγγιση μεταξύ των δύο μεθόδων είναι η mini-batch gradient descent, στην οποία σε κάθε επανάληψη χρησιμοποιείται ένα «πακέτο» δεδομένων εκπαίδευσης. Αυτό έχει το πλεονέκτημα της ομαλότερης σύγκλισης του αλγορίθμου και της καλύτερης απόδοσης από υπολογιστική άποψη, καθώς μπορούν να χρησιμοποιηθούν τεχνικές *διανυσματοποίησης (vectorization)* για επιτάχυνση των υπολογισμών.

2.2.3 Αποφυγή της υπερπροσαρμογής (Overfitting)

Η βελτιστοποίηση της συνάρτησης κόστους ενός προβλήματος μηχανικής μάθησης, γίνεται με βάση ένα πεπερασμένο σύνολο δεδομένων εκπαίδευσης. Αν το σύνολο δεδομένων δεν περιέχει ικανό αριθμό προτύπων, υπάρχει ο κίνδυνος να προσαρμοστούν οι παράμετροι του μοντέλου πολύ καλά στα συγκεκριμένα δεδομένα. Αυτό έχει ως αποτέλεσμα, ο αλγόριθμος να παρουσιάζει χαμηλή ακρίβεια πρόβλεψης σε νέα δεδομένα που δεν υπάρχουν στο σύνολο εκπαίδευσης. Το φαινόμενο αυτό ονομάζεται *υπερπροσαρμογή (overfitting)* και μειώνει την ικανότητα γενίκευσης του μοντέλου.

Για την αντιμετώπιση του προβλήματος της υπερπροσαρμογής, μια δημοφιλής μέθοδος είναι η *ομαλοποίηση (regularization)*. Η μέθοδος regularization επιβάλλει περιορισμούς στο εύρος των παραμέτρων και έτσι εμποδίζει το μοντέλο να γίνει υπερβολικά περίπλοκο κατά την εκπαίδευση του.

Συνήθως αρκεί να προστεθεί ένας επιπλέον όρος στην συνάρτηση κόστους, ο οποίος επιβάλλει ποινή στις μεγάλες τιμές παραμέτρων. Μια συνήθης τεχνική είναι η L_2- Regularization, που προσθέτει το άθροισμα των τετραγώνων όλων των παραμέτρων πολλαπλασιασμένο με ένα συντελεστή regularization λ.

Αν $J(\boldsymbol{w})$ είναι η αρχική συνάρτηση κόστους n παραμέτρων, η εκδοχή της συνάρτησης κόστους με L_2- Regularization δίνεται από την παρακάτω σχέση:

$$J_{L_2}(\boldsymbol{w}) = J(\boldsymbol{w}) + \frac{\lambda}{n}\sum_{\boldsymbol{w}} \boldsymbol{w}^2 \tag{2.8}$$

όπου $\lambda > 0$. Όσο μεγαλύτερη η τιμή του λ τόσο περιορίζονται οι τιμές των παραμέτρων και το τελικό μοντέλο είναι απλούστερο. Η βέλτιστη τιμή του λ εξαρτάται από το σύνολο δεδομένων και συνήθως επιλέγεται με ευρετικές μεθόδους.

2.3 Συστήματα συστάσεων

Τα συστήματα συστάσεων είναι συστήματα λογισμικού τα οποία έχουν τη δυνατότητα να παρέχουν στους χρήστες τους συστάσεις για αντικείμενα/προϊόντα που εμπίπτουν στα ενδιαφέροντα τους και είναι πιθανό να έχουν διαφύγει της προσοχής τους.

2.3.1 Βασικές έννοιες και ορισμοί

Τα δεδομένα που διαχειρίζονται τα συστήματα συστάσεων αφορούν τρία είδη οντοτήτων: τους *χρήστες (users)*, τα *αντικείμενα (items)* και τις *συναλλαγές (transactions)*, που γενικά είναι διάφορες συσχετίσεις μεταξύ χρηστών και αντικειμένων (Ricci et al., 2015). Στην

περίπτωση που το σύστημα προέρχεται από το χώρο του ηλεκτρονικού εμπορίου, οι χρήστες και τα αντικείμενα μπορεί να εμφανίζονται και ως *πελάτης (customer)* και *προϊόν (product)* αντίστοιχα.

Ο *χρήστης* είναι ένα άτομο που χρησιμοποιεί το σύστημα συστάσεων και που αφενός το τροφοδοτεί με δεδομένα προτίμησης σχετικά με κάποια αντικείμενα, αφετέρου επιθυμεί να λάβει συστάσεις σχετικά με καινούρια αντικείμενα τα οποία ενδεχομένως να τον ενδιαφέρουν. Ανάλογα με το σύστημα μπορεί να αξιοποιούνται και πρόσθετες πληροφορίες για τον κάθε χρήστη, όπως ηλικία, φύλο και επάγγελμα. Χρησιμοποιώντας τις πληροφορίες που διαθέτει το σύστημα για τον κάθε χρήστη, κατασκευάζεται ένα *μοντέλο χρήστη (user model)* ή αλλιώς *προφίλ χρήστη (user profile)*, δηλαδή μια αναπαράσταση του χρήστη που ενσωματώνει τις προτιμήσεις και τα χαρακτηριστικά του.

Το *αντικείμενο* μπορεί να είναι ένα οποιοδήποτε προϊόν, υπηρεσία ή οποιασδήποτε μορφής πληροφορία (π.χ. φωτογραφίες, μουσική, ειδήσεις) που μπορεί να ενδιαφέρει τους χρήστες. Συνήθως, το κάθε σύστημα συστάσεων είναι προσανατολισμένο σε ένα συγκεκριμένο είδος αντικειμένων όπως για παράδειγμα η υπηρεσία συστάσεων video του Youtube.com (Davidson et al., 2010), όμως υπάρχουν και συστήματα, ειδικά στο χώρο του ηλεκτρονικού εμπορίου, που παρέχουν συστάσεις για προϊόντα διαφόρων ειδών, όπως για παράδειγμα στα Amazon.com και Ebay.com (Schafer et al., 1999).

Με τον όρο *συναλλαγή* εννοείται οποιαδήποτε αλληλεπίδραση μεταξύ ενός χρήστη και του συστήματος συστάσεων που μπορεί να καταγραφεί. Τέτοιου είδους δεδομένα μπορεί να περιλαμβάνουν τα αντικείμενα που είδε ή αγόρασε ο χρήστης και τυχόν ανατροφοδότηση (feedback) που έδωσε για κάποια αντικείμενα.

2.3.1.1 Άμεση και έμμεση πληροφορία αξιολογήσεων

Η συνηθέστερη μορφή δεδομένων που εμπίπτουν στην κατηγορία των συναλλαγών είναι οι *αξιολογήσεις* ή *βαθμολογίες (ratings)*. Οι αξιολογήσεις συλλέγονται είτε άμεσα (explicit ratings) είτε έμμεσα (implicit ratings).

Κατά την άμεση αξιολόγηση ζητείται από τον χρήστη να εκφράσει την άποψη του για ένα αντικείμενο, μέσω μιας αξιολογικής κλίμακας. Σύμφωνα με τους (Schafer et al., 2007) οι άμεσες αξιολογήσεις μπορούν να έχουν διάφορες μορφές όπως αριθμητικές τιμές (για παράδειγμα από 1 έως 5 αστέρια), δυαδικές τιμές όπως συμφωνώ/διαφωνώ ή μου αρέσει/δε μου αρέσει.

Κατά την έμμεση αξιολόγηση το σύστημα προσπαθεί να συνάγει τις προτιμήσεις του χρήστη, αναλύοντας δεδομένα που συλλέγει από την συμπεριφορά του. Η συμπεριφορά του χρήστη περιγράφεται από το ιστορικό αγορών του, από τις επισκέψεις σε ιστοσελίδες, τις λέξεις-

κλειδιά που χρησιμοποίησε για αναζήτηση και γενικά οποιαδήποτε πληροφορία μπορεί να υπονοήσει την προτίμηση του χρήστη για κάποιο αντικείμενο.

2.3.1.2 Το πρόβλημα της παραγωγής συστάσεων

Ο στόχος των συστημάτων συστάσεων είναι η παραγωγή συστάσεων σχετικά με αντικείμενα που μπορεί να είναι χρήσιμα για τον χρήστη. Για να το επιτευχθεί αυτό, πρέπει το σύστημα να καταφέρει να προβλέψει την χρησιμότητα κάποιων αντικειμένων και να αποφασίσει ποια από αυτά να συστήσει ή να παρουσιάσει ένα κατάλογο αντικειμένων ταξινομημένο κατά φθίνουσα χρησιμότητα.

Σύμφωνα με τους (Adomavicius and Tuzhilin, 2005) το πρόβλημα της παροχής συστάσεων ορίζεται τυπικά ως εξής: Έστω C το σύνολο των χρηστών και S το σύνολο των αντικειμένων που μπορούν να συσταθούν. Επίσης ορίζεται ως u μια συνάρτηση χρησιμότητας (utility function) μέσω της οποίας υπολογίζεται η χρησιμότητα ενός αντικειμένου s για κάποιο χρήστη c. Στην συνέχεια για κάθε χρήστη $c \in C$, θέλουμε να επιλέξουμε εκείνο το αντικείμενο $s' \in S$ το οποίο μεγιστοποιεί την συνάρτηση χρησιμότητας για τον χρήστη c. Δηλαδή:

$$\forall\, c \in C, \quad s'_c = arg\max_{s \in S}\, u(c, s) \tag{2.9}$$

Σύμφωνα με τον παραπάνω ορισμό, ο χρήστης $c_a \in C$ στον οποίο αναφερόμαστε αποκαλείται *ενεργός χρήστης (active user)*. Επίσης ορίζεται το $NR \subseteq S$, ως το υποσύνολο των αντικειμένων που δεν έχουν βαθμολογηθεί από τον ενεργό χρήστη, και για τα οποία το σύστημα συστάσεων μπορεί να παράγει προβλέψεις. Αντίστοιχα, *ενεργό αντικείμενο (active item)*, s_a, ονομάζουμε κάποιο αντικείμενο που δεν έχει βαθμολογηθεί από τον ενεργό χρήστη, δηλαδή $s_a \in NR$ και για το οποίο το σύστημα συστάσεων θα παράγει κάποια πρόβλεψη.

2.3.1.3 Τύποι εργασιών σύστασης (Recommendation tasks)

Η *πρόβλεψη (prediction)* είναι μια αριθμητική τιμή r, η οποία εκφράζει την εκτιμώμενη αξιολόγηση του ενεργού χρήστη c_a για το ενεργό αντικείμενο s_a, που παρήγαγε το σύστημα συστάσεων.

Η *σύσταση (recommendation)* αποτελεί μια λίστα N αντικειμένων, τα οποία σύμφωνα με το σύστημα συστάσεων, αποτελούν τα N αντικείμενα με τις υψηλότερες αξιολογήσεις για τον ενεργό χρήστη. Η διαδικασία αυτή είναι επίσης γνωστή και ως top-N recommendation.

2.3.2 Αναπαράσταση δεδομένων αξιολόγησης

Οι αξιολογήσεις (ratings) των χρηστών αναπαριστώνται με τη βοήθεια ενός πίνακα χρηστών-αντικειμένων και χρησιμοποιούνται κυρίως από τους αλγορίθμους συνεργατικού φιλτραρίσματος που θα περιγραφούν σε επόμενη παράγραφο.

Πίνακας 2. Πίνακας αξιολογήσεων χρηστών-αντικειμένων

	I_1	I_2	I_3	I_4	I_5
U_1	4	5	2	-	3
U_2	4	5	-	-	2
U_3	1	-	3	-	-
U_4	-	-	2	-	4

Ο Πίνακας 2 αναπαριστά ένα υποθετικό πίνακα αξιολογήσεων, όπου οι γραμμές αναπαριστούν τους χρήστες και οι στήλες τα αντικείμενα, οπότε το στοιχείο *(i, j)* του πίνακα εκφράζει την τιμή αξιολόγησης του χρήστη *i* για το αντικείμενο *j*. Οι χρήστες χρησιμοποιούν μια αξιολογική κλίμακα από 1 έως 5, όπου το 1 δηλώνει την αρνητικότερη γνώμη τους και το 5 την θετικότερη γνώμη τους. Τα μη συμπληρωμένα στοιχεία του πίνακα δηλώνουν ότι οι αντίστοιχοι χρήστες δεν αξιολόγησαν τα συγκεκριμένα αντικείμενα. Σε πραγματικά συστήματα συστάσεων συνήθως τα αντικείμενα είναι πάρα πολλά, οπότε ο κάθε χρήστης τυπικά αξιολογεί μόνο λίγα από αυτά. Αυτό έχει ως αποτέλεσμα ο πίνακας δεδομένων να είναι *αραιός (sparse)* δηλαδή οι γνωστές τιμές αξιολόγησης αποτελούν ένα μικρό ποσοστό του.

2.3.3 Τύποι συστημάτων συστάσεων

Υπάρχουν διαφορετικοί τύποι Συστημάτων Συστάσεων, ανάλογα με τον τομέα εφαρμογής τους, τις πληροφορίες που αξιοποιούν και τον αλγόριθμο που χρησιμοποιούν. Ο (Burke, 2007) ταξινομεί τα Συστήματα Συστάσεων σε πέντε κατηγορίες.

2.3.3.1 Συνεργατικό Φιλτράρισμα (Collaborative Filtering)

Τα συστήματα αυτού του τύπου χρησιμοποιούν τις αξιολογήσεις πολλών χρηστών σχετικά με αντικείμενα, προσπαθώντας να εντοπίσουν μοτίβα (ομοιότητες ή ανομοιότητες) στις αξιολογήσεις τους. Στη συνέχεια χρησιμοποιούν αυτά τα μοτίβα για να συστήσουν στον

ενεργό χρήστη αντικείμενα, τα οποία αξιολογήθηκαν θετικά από παρόμοιους με αυτόν χρήστες.

Η κατηγορία αυτή είναι η πιο διαδεδομένη τεχνική παραγωγής συστάσεων και χρησιμοποιείται ευρέως στο διαδίκτυο. Το βασικό της πλεονέκτημα είναι η απλότητα στην υλοποίηση, καθώς δεν είναι απαραίτητο να εξαχθούν χαρακτηριστικά των αντικειμένων και μάλιστα λειτουργεί με τον ίδιο τρόπο για οποιοδήποτε είδος αντικειμένου.

Παρόλα αυτά, η τεχνική αυτή επηρεάζεται από δύο σημαντικά προβλήματα που αναλύονται σε επόμενη παράγραφο: το πρόβλημα της αραιότητας των αξιολογήσεων (rating sparsity) και το πρόβλημα της ψυχρής εκκίνησης (cold-start), τα οποία αναλύονται σε επόμενη παράγραφο. Τέλος, το συνεργατικό φιλτράρισμα είναι ευάλωτο σε επιθέσεις κακόβουλων χρηστών (shilling attacks), οι οποίοι δημιουργώντας πλαστά προφίλ μπορεί να αλλοιώσουν τις αξιολογήσεις κάποιων αντικειμένων, για παράδειγμα να δώσουν πολλές θετικές αξιολογήσεις σε δικά τους αντικείμενα και πολλές αρνητικές σε αντικείμενα ανταγωνιστών.

2.3.3.2 Φιλτράρισμα με βάση το περιεχόμενο (Content-based Filtering)

Αυτός ο τύπος συστήματος συστήνει αντικείμενα που είναι παρόμοια με τα αντικείμενα που άρεσαν στον χρήστη παλαιότερα. Ο υπολογισμός της ομοιότητας *(similarity)* γίνεται με βάση τα χαρακτηριστικά των αντικειμένων. Τα χαρακτηριστικά συνήθως αποτελούνται από *λέξεις-κλειδιά (keywords)* που εξάγονται από την περιγραφή των αντικειμένων και συνθέτουν το *προφίλ του αντικειμένου (item profile)*. Έτσι, το σύστημα επιλέγει αντικείμενα με βάση τη συσχέτιση μεταξύ του περιεχομένου των αντικειμένων και των προτιμήσεων των χρηστών σε αντίθεση με το συνεργατικό φιλτράρισμα που επιλέγει αντικείμενα με βάση τη συσχέτιση μεταξύ χρηστών. Για παράδειγμα ένα σύστημα συστάσεων για μουσικά κομμάτια μπορεί να λαμβάνει υπόψη πληροφορίες για το είδος της μουσικής, το άλμπουμ και τους καλλιτέχνες που συμμετέχουν, συνθέτοντας έτσι το προφίλ κάθε μουσικού κομματιού. Στη συνέχεια με βάση το προφίλ των κομματικών που αρέσουν σε ένα χρήστη, του προτείνει άλλα παρόμοια κομμάτια του ίδιου καλλιτέχνη ή συγκροτήματος.

Το κύριο μειονέκτημα της μεθόδου αυτής είναι ότι απαιτεί την ύπαρξη συγκεκριμένων πληροφοριών ανάλογα με το είδος του αντικειμένου, πράγμα που δεν είναι πάντα εύκολο. Επίσης, μπορεί να παρατηρηθεί το φαινόμενο της υπερεξειδίκευσης, δηλαδή το σύστημα να παράγει σχεδόν πανομοιότυπες συστάσεις χωρίς κανένα στοιχεία καινοτομίας.

2.3.3.3 Με βάση την γνώση (Knowledge based)

Τα συστήματα αυτού του είδους στηρίζονται σε συγκεκριμένη γνώση η οποία καθορίζει κατά πόσο τα χαρακτηριστικά ενός αντικειμένου ανταποκρίνονται στις ανάγκες και τα ενδιαφέροντα του χρήστη, δηλαδή αν το αντικείμενο θα είναι χρήσιμο στον χρήστη ή όχι. Η

γνώση προέρχεται από ειδικούς, οπότε η ποιότητα των συστάσεων είναι υψηλού επιπέδου. Τα συστήματα αυτού του είδους μπορούν να χρησιμοποιηθούν όταν δεν είναι δυνατή η χρήση συνεργατικών μεθόδων ή μεθόδων με βάση το περιεχόμενο. Παρόλα αυτά η απόκτηση της γνώσης που απαιτείται για την αποδοτική λειτουργία του συστήματος συχνά είναι δύσκολη ή δαπανηρή.

2.3.3.4 Δημογραφικά Συστήματα (Demographic systems)

Τα Δημογραφικά Συστήματα Συστάσεων παράγουν συστάσεις για αντικείμενα βάσει του δημογραφικού profile του χρήστη. Πολλές ιστοσελίδες εφαρμόζουν απλές εξατομικευμένες λύσεις με βάση τις δημογραφικές πληροφορίες, όπως για παράδειγμα κάποιες φορές οι χρήστες οδηγούνται σε συγκεκριμένες σελίδες σύμφωνα με την χώρα και την γλώσσα τους ή οι συστάσεις προσαρμόζονται στην ηλικία και στο φύλο του χρήστη. Τα συστήματα αυτά χρησιμοποιούνται κυρίως στο πεδίο του marketing, καθώς δεν είναι αποδοτικά για συστάσεις αντικειμένων, δεδομένου ότι τα μέλη μιας δημογραφικής ομάδας μπορεί να έχουν πολύ διαφορετικά ενδιαφέροντα ή προτιμήσεις σχετικά με αντικείμενα ή π.χ. είδη μουσικής. Παρόλα αυτά οι δημογραφικές πληροφορίες μπορούν να συνδυαστούν με επιτυχία με άλλες προσεγγίσεις.

2.3.3.5 Υβριδικό Φιλτράρισμα (Hybrid Filtering)

Η κατηγορία αυτή συστημάτων χρησιμοποιεί ένα συνδυασμό των τεχνικών που αναφέραμε παραπάνω, εκμεταλλευόμενα τα προτερήματα της μίας τεχνικής για να καλύψουν τα μειονεκτήματα της άλλης. Υπάρχουν πολλοί διαφορετικοί τρόποι με τους οποίους συνδυάζονται δύο ή και περισσότεροι τύποι συστημάτων συστάσεων για να δημιουργηθεί ένα υβριδικό σύστημα (Burke, 2007).

Μερικοί συνήθεις τρόποι δημιουργίας υβριδικών συστημάτων είναι οι παρακάτω:

- Να εφαρμοστούν ξεχωριστά η μέθοδος συνεργατικού φιλτραρίσματος και η μέθοδος με βάση το περιεχόμενο και στη συνέχεια να συνδυαστούν τα αποτελέσματα τους (Christakou et al., 2007).
- Στην μέθοδο συνεργατικού φιλτραρίσματος να ενσωματωθούν κάποια από τα χαρακτηριστικά του περιεχομένου των αντικειμένων.
- Στην μέθοδο με βάση το περιεχόμενο να χρησιμοποιηθούν και χαρακτηριστικά συνεργατικής προσέγγισης (Debnath et al., 2008).
- Να κατασκευαστεί ένα ενοποιημένο σύστημα το οποίο ενσωματώνει χαρακτηριστικά και των δύο μεθόδων (Cantador et al., 2008).

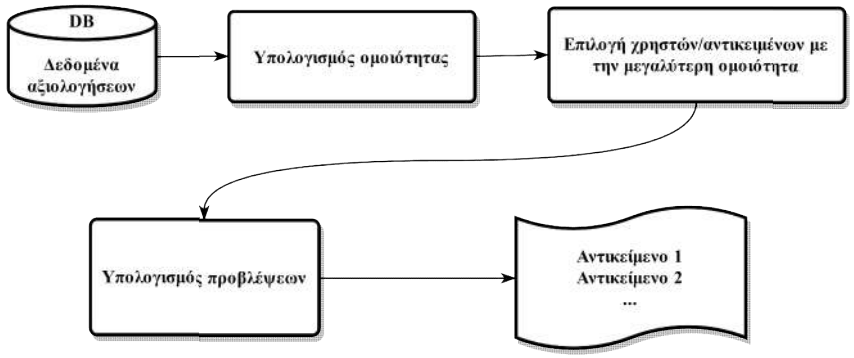
2.3.4 Τρόποι λειτουργίας συστημάτων Συνεργατικού Φιλτραρίσματος

Η παρούσα εργασία εστιάζεται στην κατηγορία των συστημάτων Συνεργατικού Φιλτραρίσματος. Στην ενότητα αυτή θα γίνει μια ταξινόμηση των συστημάτων αυτών με βάση τον τρόπο λειτουργίας τους.

2.3.4.1 Συστήματα βασισμένα στη μνήμη (Memory-based)

Τα συστήματα που βασίζονται στη μνήμη χρησιμοποιούν στατιστικές τεχνικές για να βρουν ομάδες χρηστών, που συχνά αποκαλούνται *γειτονιές (neighborhoods)*, όπου οι χρήστες μοιάζουν με τον ενεργό χρήστη. Τα συστήματα αυτά βασίζονται κυρίως σε υπολογισμούς ομοιότητας μεταξύ χρηστών ή αντικειμένων χρησιμοποιώντας όλες τις αξιολογήσεις που περιέχει το σύστημα, δηλαδή στη μνήμη τους, από όπου πήραν και το όνομα τους.

Η λειτουργία των συστημάτων που βασίζονται στη μνήμη αποτελείται από 3 φάσεις: την φάση υπολογισμού ομοιότητας, την φάση επιλογής των χρηστών/αντικειμένων με την μεγαλύτερη ομοιότητα και τέλος την φάση υπολογισμού των προβλέψεων. Η διαδικασία φαίνεται σχηματικά στο Σχήμα 8.



Σχήμα 8. Λειτουργία συστημάτων συνεργατικού φιλτραρίσματος βασισμένα στη μνήμη

Στην φάση υπολογισμού της ομοιότητας, υπάρχουν διάφοροι τρόποι που μπορούν να εφαρμοστούν. Ο συνήθης τρόπος είναι να θεωρηθούν οι αξιολογήσεις ως σημεία ενός πολυδιάστατου χώρου και στη συνέχεια να υπολογιστεί η μεταξύ τους ομοιότητα ως συνάρτηση της απόστασης μεταξύ των σημείων.

Η απλούστερη και πιο συνηθισμένη μετρική απόστασης είναι η *Ευκλείδεια απόσταση* (ή *νόρμα* L_2). Έτσι για να υπολογιστεί η ομοιότητα του ενεργού χρήστη *a* με έναν άλλο χρήστη *u* χρησιμοποιείται η Εξίσωση (2.10): (Amatriain and Pujol, 2015).

$$d_{a,u} = \sqrt{\sum_{i \in I_{au}} \left(r_{a,i} - r_{u,i}\right)^2} \qquad (2.10)$$

όπου $I_{\alpha u}$ είναι το σύνολο των αντικειμένων που έχουν αξιολογήσει και οι δύο χρήστες και $r_{a,i}$, $r_{u,i}$ είναι η αξιολόγηση που έδωσαν στο αντικείμενο i αντίστοιχα.

Μια γενικευμένη μετρική απόστασης με βάση την Ευκλείδεια είναι η απόσταση Minkowski:

$$d_{a,u} = \left(\sum_{i \in I_{au}} \left| r_{a,i} - r_{u,i} \right|^r \right)^{\frac{1}{r}} \tag{2.11}$$

όπου r είναι ο βαθμός της απόστασης. Ανάλογα με την τιμή του r ο γενικός τύπος παίρνει συγκεκριμένες μορφές. Για $r = 2$ προκύπτει ο τύπος της Ευκλείδειας απόστασης (Εξ. (2.10)), ενώ για $r = 1$ προκύπτει ο τύπος της απόστασης *Manhattan* (ή *νόρμα* L_1).

Μια άλλη προσέγγιση είναι να θεωρηθούν οι αξιολογήσεις ενός χρήστη ή ενός αντικειμένου ως διανύσματα ενός πολυδιάστατου χώρου και στη συνέχεια να υπολογιστεί η μεταξύ τους ομοιότητα ως συνάρτηση του συνημιτόνου της γωνίας που σχηματίζουν τα δύο διανύσματα. Η μετρική αυτή είναι γνωστή ως *ομοιότητα συνημιτόνου (cosine similarity)* και υπολογίζεται από τον τύπο:

$$\cos(r_a, r_u) = \frac{\overrightarrow{r_a} \cdot \overrightarrow{r_u}}{\|\overrightarrow{r_a}\|_2 \times \|\overrightarrow{r_u}\|_2} = \frac{\sum_{i \in I_{au}} r_{a,i}\, r_{u,i}}{\sqrt{\sum_{i \in I_{au}} r_{a,i}^2} \sqrt{\sum_{i \in I_{au}} r_{u,i}^2}} \tag{2.12}$$

όπου $\overrightarrow{r_a}$ και $\overrightarrow{r_u}$ τα δύο διανύσματα, το σύμβολο $\cdot$ είναι το εσωτερικό γινόμενο των διανυσμάτων και $\|\overrightarrow{r_a}\|_2$ είναι η νόρμα του διανύσματος $\overrightarrow{r_a}$. Η ελάχιστη τιμή της μετρικής είναι -1 που δηλώνει ότι τα διανύσματα είναι αντίθετα και η μέγιστη 1, δηλαδή ότι τα διανύσματα ταυτίζονται. Όταν τα διανύσματα είναι κάθετα μεταξύ τους (γωνία 90^0) το συνημίτονο είναι 0 υποδηλώνοντας ότι τα διανύσματα είναι ασυσχέτιστα (Su and Khoshgoftaar, 2009).

Η ομοιότητα μπορεί επίσης να υπολογιστεί με την χρήση συντελεστών συσχέτισης που χρησιμοποιούνται στην Στατιστική και περιγράφουν την σχέση μεταξύ δύο μεταβλητών. Υπάρχουν πολλοί συντελεστές συσχέτισης, όμως αυτός που χρησιμοποιείται πιο συχνά είναι ο *Συντελεστής Συσχέτισης του Pearson (Pearson's Correlation Coefficient)*(Amatriain and Pujol, 2015). Ο συντελεστής συσχέτισης του Pearson μεταξύ δύο μεταβλητών μετράει το βαθμό γραμμικής συσχέτισης τους και ορίζεται ως το πηλίκο της διαίρεσης της συνδιακύμανσης των δύο μεταβλητών με το γινόμενο των τυπικών αποκλίσεων τους. Έτσι, στην περίπτωση των αξιολογήσεων των δύο χρηστών a και s, ο συντελεστής συσχέτισης Pearson δίνεται από τον τύπο:

$$Pearson_{a,u} = \frac{\Sigma(I_a, I_u)}{\sigma_{I_\alpha} \times \sigma_{I_u}} = \frac{\sum_{i \in I_{au}} (r_{a,i} - \overline{r_a})\,(r_{u,i} - \overline{r_u})}{\sqrt{\sum_{i \in I_{au}} (r_{a,i} - \overline{r_a})^2 \sum_{i \in I_{au}} (r_{a,i} - \overline{r_u})^2}} \quad (2.13)$$

όπου $\Sigma(I_a, I_u)$ είναι η συνδιακύμανση των αξιολογήσεων των χρηστών a και s, σ_{I_α}, σ_{I_u} είναι οι αντίστοιχες τυπικές αποκλίσεις και $\overline{r_a}$, $\overline{r_u}$ είναι η μέση τιμή αξιολογήσεων των αντικειμένων που έχουν αξιολογήσει και οι δύο χρήστες. Η μέγιστη τιμή είναι 1 όταν οι δύο χρήστες συσχετίζονται τέλεια και η ελάχιστη -1 όταν υπάρχει αντισυσχέτιση. Οι ενδιάμεσες τιμές δείχνουν το βαθμό της γραμμικής εξάρτησης μεταξύ τους. Η τιμή 0 δείχνει ότι δεν υπάρχει συσχέτιση μεταξύ τους.

Εκτός από τις προαναφερθείσες, υπάρχουν και άλλες μετρικές ομοιότητας όπως *Προσαρμοσμένη Ομοιότητα Συνημιτόνου (Adjusted Cosine Similarity)*, *Spearman's Correlation*, *Kendall's τ correlation* (Herlocker et al., 1999; Su and Khoshgoftaar, 2009).

Σημειώνεται ότι ο υπολογισμός ομοιότητας δεν γίνεται μόνο με βάση τους χρήστες *(user-based)*, αλλά μπορεί να υπολογιστεί και με βάση τα αντικείμενα *(item-based)*. Στην περίπτωση αυτή όλοι οι τύποι για τις μετρικές που αναφέρθηκαν υπολογίζονται χρησιμοποιώντας το σύνολο των κοινών αξιολογήσεων που έχουν λάβει δύο αντικείμενα, αντί για το σύνολο των κοινών αξιολογήσεων που έχουν δώσει δύο χρήστες (Sarwar et al., 2001). Η τεχνική item-based μπορεί να εφαρμοστεί όταν ο αριθμός των χρηστών ενός συστήματος είναι αρκετά μεγαλύτερος από τον αριθμό των αντικειμένων, οπότε η user-based προσέγγιση είναι πιο ακριβή υπολογιστικά.

Στην επόμενη φάση της διαδικασίας, γίνεται η επιλογή χρηστών/αντικειμένων με την μεγαλύτερη ομοιότητα με τον ενεργό χρήστη ή το ενεργό αντικείμενο αντίστοιχα. Για την επιλογή συνήθως χρησιμοποιείται η προσέγγιση των *Κ κοντινότερων γειτόνων (K Nearest Neighbors)*, κατά την οποία οι χρήστες ταξινομούνται κατά φθίνουσα σειρά της μετρικής ομοιότητας που χρησιμοποιήθηκε και κρατούνται οι Κ χρήστες με την μεγαλύτερη τιμή. Μια εναλλακτική προσέγγιση είναι με την χρήση *κατωφλίου (threshold)*, όπου κρατούνται όλοι οι χρήστες των οποίων η τιμή της μετρικής ομοιότητας υπερβαίνει μια προκαθορισμένη τιμή κατωφλίου (Kim and Yang, 2007). Η διαδικασία γίνεται με αντίστοιχο τρόπο τόσο στην user-based προσέγγιση, όσο και στην item-based.

Στην τελευταία φάση του υπολογισμού των προβλέψεων, ουσιαστικά γίνεται η επεξεργασία των αξιολογήσεων των όμοιων χρηστών (ή αντικειμένων στην περίπτωση item-based) που προέκυψαν από την προηγούμενη φάση, ώστε να παραχθούν προβλέψεις για τον ενεργό χρήστη (ή αντικείμενο). Η πιο απλή προσέγγιση είναι ο υπολογισμός του μέσου όρου των αξιολογήσεων των γειτόνων (Εξ. (2.14)). Μια πιο αποτελεσματική προσέγγιση είναι να υπολογιστεί ο σταθμισμένος μέσος όρος των αξιολογήσεων των γειτόνων χρησιμοποιώντας

τους βαθμούς ομοιότητας ως βάρη (Εξ. (2.15)). Με αυτόν τον τρόπο οι χρήστες με μεγαλύτερο βαθμό ομοιότητας συνεισφέρουν περισσότερο.

$$\hat{r}_{a,i} = \frac{1}{|U_{a,i}|} \sum_{u \in U_{a,i}} r_{u,i} \tag{2.14}$$

$$\hat{r}_{a,i} = \frac{\sum_{u \in U_{a,i}} w_{a,u}\, r_{u,i}}{\sum_{u \in U_{a,i}} w_{a,u}} \tag{2.15}$$

όπου $\hat{r}_{a,i}$ είναι η πρόβλεψη της αξιολόγησης που θα έκανε ο ενεργός χρήστης a για ένα αντικείμενο i, $U_{a,i}$ είναι το σύνολο των όμοιων ως προς τον α χρηστών, που έχουν αξιολογήσει το αντικείμενο i και έχουν προκύψει από την προηγούμενη φάση επιλογής. Τέλος, $w_{a,u}$ είναι η τιμή της μετρικής ομοιότητας του χρήστη α ως προς το χρήστη u.

Μια παραλλαγή του σταθμισμένου μέσου όρου που χρησιμοποιεί την απόκλιση των αξιολογήσεων από τον μέσο όρο αξιολογήσεων του εκάστοτε χρήστη, προτάθηκε από τους (Resnick et al., 1994) και υπολογίζεται από την Εξ. (2.16).

$$\hat{r}_{a,i} = \overline{r_a} + \frac{\sum_{u \in U_{a,i}} w_{a,u}\, (r_{u,i} - \overline{r_u})}{\sum_{u \in U_{a,i}} |w_{a,u}|} \tag{2.16}$$

όπου $\overline{r_a}$, $\overline{r_u}$ ο μέσος όρος των αξιολογήσεων του χρήστη a και u αντίστοιχα. Αυτή η προσέγγιση προσπαθεί να αντιμετωπίσει το γεγονός ότι δεν αξιολογούν όλοι οι χρήστες με τον ίδιο τρόπο, αλλά κάποιοι τείνουν να δίνουν γενικά μεγάλες βαθμολογίες, ενώ κάποιοι άλλοι να είναι εξαιρετικά κριτικοί.

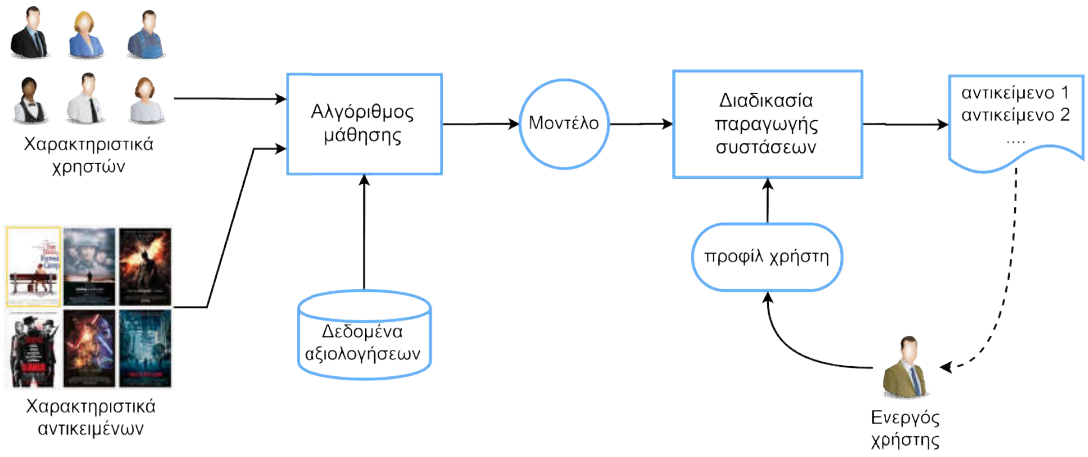
Το κύριο πλεονέκτημα των συστημάτων με βάση τη μνήμη είναι η απλότητα τους και η εύκολη υλοποίηση τους. Επίσης, η ανανέωση των προβλέψεων κατά την προσθήκη νέων δεδομένων στο σύστημα είναι άμεση. Παρόλα αυτά, ακριβώς επειδή επεξεργάζονται όλα τα διαθέσιμα δεδομένα, παρουσιάζουν προβλήματα σε μεγάλα συστήματα λόγω του αυξανόμενου υπολογιστικού κόστους. Τέλος, δεν μπορούν να παραγάγουν συστάσεις για χρήστες που δεν έχουν κοινές αξιολογήσεις με άλλους χρήστες και επίσης δεν μπορούν να προτείνουν αντικείμενα τα δεν έχουν αξιολογηθεί από κανέναν.

2.3.4.2 Συστήματα βασισμένα σε μοντέλο (Model-based)

Τα Συστήματα που βασίζονται σε μοντέλο χρησιμοποιούν αλγορίθμους Μηχανικής Μάθησης και με βάση τα δεδομένα αξιολογήσεων του συστήματος εκπαιδεύουν μοντέλα, προσπαθώντας να ανακαλύψουν μοτίβα στις υπάρχουσες αλληλεπιδράσεις μεταξύ χρηστών και αντικειμένων. Επιπλέον, μπορούν να χρησιμοποιηθούν και χαρακτηριστικά των χρηστών ή/και αντικειμένων για βελτίωση των συστάσεων. Στη συνέχεια τα μοντέλα αυτά

χρησιμοποιούνται για να παράγουν συστάσεις με βάση το προφίλ του ενεργού χρήστη (Σχήμα 9).

Τα μοντέλα που κατασκευάζονται στα πλαίσια συστημάτων συστάσεων μπορεί να είναι είτε μοντέλα ταξινόμησης (classification models), είτε μοντέλα παλινδρόμησης (regression models). Τα πρώτα προσπαθούν να ταξινομήσουν τα αντικείμενα που δεν έχει αξιολογήσει ο χρήστης σε κλάσεις ανάλογα με το πόσο μπορεί να του αρέσουν. Αντίθετα, τα μοντέλα παλινδρόμησης προσπαθούν να προβλέψουν την τιμή αξιολόγησης που θα έδινε ένας χρήστης σε αντικείμενα που δεν έχει αξιολογήσει ακόμα.



Σχήμα 9. Λειτουργία συστημάτων συνεργατικού φιλτραρίσματος βασισμένα σε μοντέλο

Οι προσεγγίσεις που βασίζονται σε μοντέλο είναι πολυάριθμες. Ενδεικτικά, κάποιες από αυτές είναι τα Μπεϋζιανά Δίκτυα (Bayesian Networks), οι τεχνικές συσταδοποίησης (Clustering), τα νευρωνικά δίκτυα (Neural Networks), οι Περιορισμένες Μηχανές Boltzmann (Restricted Boltzmann Machines) και οι Μηχανές Διανυσμάτων Υποστήριξης (Support Vector Machines) (Su and Khoshgoftaar, 2009). Πέρα από τις παραπάνω, οι προσεγγίσεις βασισμένες σε μοντέλο που θεωρούνται ως state-of-the-art είναι τα μοντέλα Λανθανόντων Παραγόντων (Latent Factor Models) και Παραγοντοποίησης Πινάκων (Matrix Factorization) (Koren et al., 2009).

Τα πλεονεκτήματα των συστημάτων που βασίζονται σε μοντέλο είναι η αποτελεσματικότητα τους σχετικά με την αναπαράσταση των προτιμήσεων των χρηστών, η ταχύτητα παραγωγής των συστάσεων (εφόσον είναι ήδη εκπαιδευμένα) και ότι συνήθως είναι πολύ μικρότερα σε μέγεθος από τα δεδομένα τα οποία χρειάστηκαν για να εκπαιδευτούν. Από την άλλη, ένα σοβαρό μειονέκτημα είναι ότι συνήθως η διαδικασία εκπαίδευσης τους έχει μεγάλο υπολογιστικό κόστος και ότι δεν είναι πάντα εύκολο να ενημερωθούν με νέα δεδομένα.

2.3.5 Προβλήματα και προκλήσεις

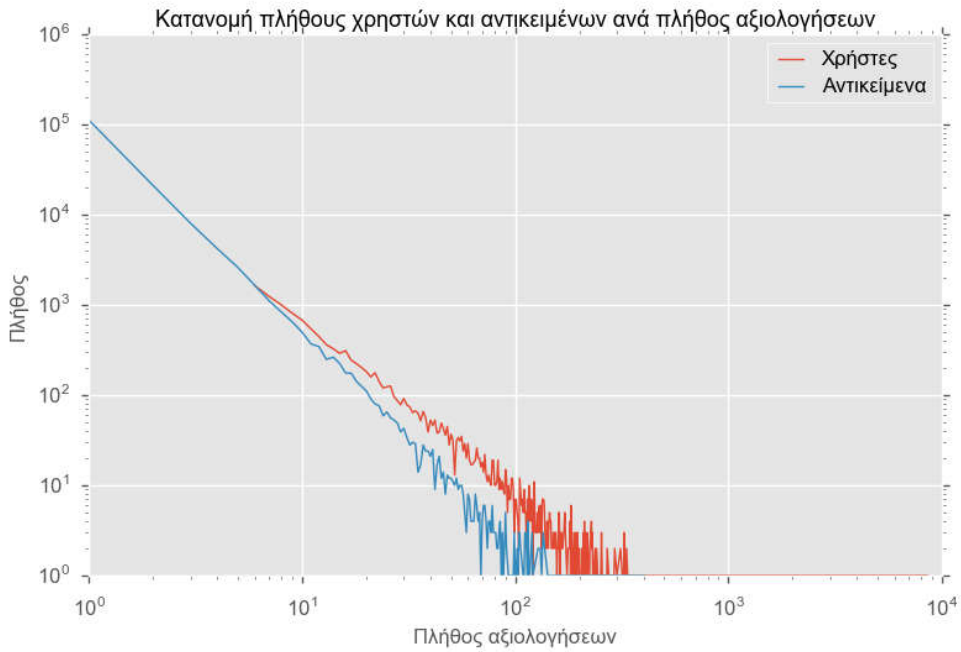
Τα Συστήματα Συστάσεων προσπαθούν να αντιμετωπίσουν διάφορα προβλήματα και προκλήσεις και ανάλογα με την κατηγορία τους μπορεί να τα αντιμετωπίζουν σε διαφορετικό βαθμό. Στη συνέχεια αναφέρονται τα σημαντικότερα από αυτά (Adomavicius and Tuzhilin, 2005).

2.3.5.1 Αραιότητα των αξιολογήσεων (Rating sparsity)

Στα συστήματα Συνεργατικού Φιλτραρίσματος, συνήθως υπάρχει ένας αρκετά μεγάλος αριθμός χρηστών οι οποίοι εκφράζουν τις προτιμήσεις τους μέσω αξιολογήσεων για ένα επίσης μεγάλο αριθμό αντικειμένων. Όμως, ο κάθε χρήστης συνήθως αξιολογεί μόνο ένα μικρό ποσοστό των αντικειμένων, άρα ο συνολικός αριθμός των «γνωστών» αξιολογήσεων είναι, στις περισσότερες περιπτώσεις, πολύ μικρότερος από τον αριθμό των αξιολογήσεων που το σύστημα καλείται να προβλέψει. Η κατάσταση αυτή είναι γνωστή ως πρόβλημα αραιότητας των αξιολογήσεων και η πρόκληση είναι να καταφέρει το σύστημα να προβλέψει αξιολογήσεις βασιζόμενο σε σχετικά λίγα δεδομένα.

Η αραιότητα μπορεί να έχει διάφορες επιπτώσεις στη λειτουργία του συστήματος. Συνήθως τα αραιά δεδομένα αξιολογήσεων χαρακτηρίζονται και από ανισότητα στην κατανομής τους, δηλαδή μια μικρή ομάδα δημοφιλών αντικειμένων λαμβάνει πολλές αξιολογήσεις, ενώ αντίθετα τα υπόλοιπα αντικείμενα έχουν πολύ λίγες έως ελάχιστες αξιολογήσεις. Έτσι για παράδειγμα τα αντικείμενα που έχουν αξιολογηθεί από πολύ λίγους χρήστες, ακόμα και να αξιολογήθηκαν πολύ υψηλά, το σύστημα θα τα προτείνει σε άλλους χρήστες πολύ σπάνια. Αντίστοιχα, συνήθως υπάρχει μια μικρή ομάδα «φανατικών» χρηστών, οι οποίοι έχουν αξιολογήσει πάρα πολλά αντικείμενα, ενώ η πλειοψηφία των χρηστών έχει πολύ λίγες αξιολογήσεις. Το Σχήμα 10 δείχνει την κατανομή του πλήθους των χρηστών σε σχέση με το πλήθος των αξιολογήσεων που έχουν δώσει και την κατανομή του πλήθους των αντικειμένων σε σχέση με το πλήθος αξιολογήσεων που έχουν λάβει, στην συλλογή δεδομένων Book Crossing (Ziegler et al., 2005). Η κατανομή ακολουθεί τον νόμο του Zipf, δηλαδή είναι εκθετική, αλλά η αναπαράσταση έγινε σε λογαριθμικούς άξονες (log-log) για μεγαλύτερη ευκρίνεια, όπου η κατανομή φαίνεται να είναι γραμμική.

Σε γενικές γραμμές η σημαντικότερη συνέπεια της αραιότητας των αξιολογήσεων είναι η χαμηλή αλληλοκάλυψη, δηλαδή η μικρή πιθανότητα να βρεθούν χρήστες οι οποίοι να έχουν αξιολογήσει τα ίδια αντικείμενα. Για το λόγο αυτό, ένα σύστημα που βασίζεται στην αναζήτηση ομοιότητας μεταξύ των χρηστών είναι πολύ πιθανό να καταλήξει σε εσφαλμένα αποτελέσματα, όταν η αλληλοκάλυψη είναι χαμηλή. Για παράδειγμα αν κάποιος χρήστης έχει ασυνήθιστα ενδιαφέροντα σε σχέση με την πλειοψηφία των υπολοίπων, το σύστημα δεν θα μπορέσει να βρει παρόμοιους χρήστες ώστε να παράγει ικανοποιητικές συστάσεις γι' αυτόν.



Σχήμα 10. Κατανομή πλήθους χρηστών και αντικειμένων ανά πλήθος αξιολογήσεων στην συλλογή δεδομένων Book Crossing (Ziegler et al., 2005), με ποσοστό αραιότητας 99,997%

Η αντιμετώπιση του προβλήματος δεν είναι εύκολη, συνήθως όμως γίνεται με τη βοήθεια επιπλέον τεχνικών ή αξιοποιώντας πρόσθετα δεδομένα που μπορεί να είναι διαθέσιμα για τον χρήστη. Αν είναι διαθέσιμα δεδομένα για το προφίλ του χρήστη όπως για παράδειγμα φύλο, ηλικία, τόπος κατοικίας, εκπαίδευση και επάγγελμα, αυτά μπορούν να χρησιμοποιηθούν κατά την εύρεση παρόμοιων χρηστών, δηλαδή αν δεν μπορεί να βρεθεί παρόμοιος χρήστης με βάση τις αξιολογήσεις να αναζητηθούν χρήστες με παρόμοια δημογραφικά χαρακτηριστικά. Η υποβοήθηση της τεχνικής του Συνεργατικού Φιλτραρίσματος με αυτόν τον τρόπο ονομάζεται επίσης και Δημογραφικό Φιλτράρισμα (demographic filtering). Παρόλα αυτά, η λύση αυτή δεν είναι πάντα εφικτή, καθώς η συλλογή δημογραφικών δεδομένων δεν είναι εύκολη, συνήθως επειδή οι χρήστες για διάφορους λόγους μπορεί να μην δέχονται να παρέχουν τέτοια δεδομένα, είτε πλήρη είτε έστω ελλιπή (Pazzani, 1999).

Ένας άλλος τρόπος αντιμετώπισης του προβλήματος της αραιότητας των αξιολογήσεων είναι η χρήση τεχνικών μείωσης διαστατικότητας, όπως για παράδειγμα η Παραγοντοποίηση Ιδιαζουσών τιμών (Singular Value Decomposition – SVD) (Sarwar et al., 2000). Η τεχνική αυτή αναλύεται σε επόμενη παράγραφο της εργασίας.

2.3.5.2 Το πρόβλημα της ψυχρής-εκκίνησης (Cold-start problem)

Κατά την λειτουργία ενός Συστήματος Συστάσεων εκτός από αξιολογήσεις, προστίθενται συνεχώς και νέοι χρήστες και νέα αντικείμενα. Υπάρχουν λοιπόν δύο όψεις στο πρόβλημα της ψυχρής-εκκίνησης, η εγγραφή ενός νέου χρήστη και η εισαγωγή ενός νέου αντικειμένου στο σύστημα.

Στην περίπτωση του νέου χρήστη, το πρόβλημα είναι ότι το σύστημα δεν διαθέτει καμία πληροφορία σχετικά με τις προτιμήσεις του. Κατά συνέπεια, με Συνεργατικό Φιλτράρισμα το σύστημα δεν μπορεί να βρει χρήστες με παρόμοιες προτιμήσεις, ενώ με Φιλτράρισμα με βάση το περιεχόμενο δεν μπορεί να βρει παρόμοια αντικείμενα. Επιπλέον, ανάλογα με το χρησιμοποιούμενο αλγόριθμο, υπάρχει κάποιος ελάχιστος αριθμός απαιτούμενων αξιολογήσεων που πρέπει να παρέχει ο χρήστης, πριν το σύστημα μπορέσει να αρχίσει να παράγει αξιόπιστες προβλέψεις, π.χ. πρέπει ένας νέος χρήστης να αξιολογήσει τουλάχιστον δέκα αντικείμενα. Το πρόβλημα του νέου χρήστη αντιμετωπίζεται συνήθως με την χρήση υβριδικών προσεγγίσεων, συνδυάζοντας δηλαδή τεχνικές Συνεργατικού Φιλτραρίσματος και Φιλτραρίσματος με βάση το περιεχόμενο. Μια άλλη προσέγγιση είναι να προτείνονται στους νέους χρήστες κάποια δημοφιλή αντικείμενα, έως ότου συμπληρωθεί ο ελάχιστος απαιτούμενος αριθμός αξιολογήσεων (Adomavicius and Tuzhilin, 2005).

Αντίστοιχα στην περίπτωση του νέου αντικειμένου που εισάγεται σε ένα σύστημα Συνεργατικού Φιλτραρίσματος, αν το αντικείμενο δεν συγκεντρώσει ένα ελάχιστο αριθμό αξιολογήσεων, το σύστημα δεν μπορεί να το προτείνει σε κανένα χρήστη. Και αυτή η περίπτωση αντιμετωπίζεται με την χρήση υβριδικών τεχνικών.

2.4 Το βραβείο Netflix

Το βραβείο Netflix ήταν ένας ανοιχτός διαγωνισμός που ξεκίνησε το 2006 με σκοπό την ανάδειξη του καλύτερου αλγορίθμου συνεργατικού φιλτραρίσματος για πρόβλεψη αξιολογήσεων χρηστών για ταινίες, με χρηματικό έπαθλο ενός εκατομμυρίου δολαρίων. Το διαγωνισμό πραγματοποίησε η εταιρία Netflix, μια υπηρεσία παροχής video streaming μέσω διαδικτύου. Στο διάστημα διενέργειας του διαγωνισμού ήταν μια διαδικτυακή υπηρεσία ενοικίασης ταινιών DVD.

2.4.1 Περιγραφή του διαγωνισμού

Η εταιρία δημοσίευσε μια πραγματικά μεγάλη, για την εποχή εκείνη, συλλογή δεδομένων, που περιελάμβανε περίπου 100 εκατομμύρια αξιολογήσεις από 480.189 χρήστες για 17.770 ταινίες. Κάθε εγγραφή των δεδομένων αποτελούνταν από μια τετράδα της μορφής < χρήστης, ταινία, ημερομηνία αξιολόγησης, βαθμός αξιολόγησης >. Τα πεδία του χρήστη και της ταινίας ήταν ακέραιοι αναγνωριστικοί αριθμοί (id), ενώ οι βαθμοί αξιολόγησης ήταν ακέραιοι αριθμοί εύρους 1-5 (αστέρια αξιολόγησης) (Bennett and Lanning, 2007). Τα δεδομένα αυτά αποτελούσαν τα δεδομένα εκπαίδευσης (*training set*) για τους αλγορίθμους. Επιπλέον, η εταιρία παρείχε και ένα σύνολο δεδομένων αποτελούμενο από περίπου 2,8 εκατομμύρια εγγραφές, γνωστό ως *qualifying set*, σε μορφή τριάδων της μορφής

< χρήστης, ταινία, ημερομηνία αξιολόγησης >, αλλά οι τιμές αξιολογήσεων ήταν γνωστές μόνο στην επιτροπή αξιολόγησης του διαγωνισμού. Ο σκοπός του διαγωνισμού ήταν να παραχθούν προβλέψεις αξιολογήσεων για τις άγνωστες τιμές αξιολογήσεων του qualifying set.

Οι διαγωνιζόμενοι μπορούσαν να υποβάλλουν τις προβλέψεις τους στην ιστοσελίδα[3] του διαγωνισμού και αυτόματα να δουν την ακρίβεια τους με βάση την μετρική RMSE (Root Mean Squared Error) (παραγρ. 3.5.2.2) για περίπου τα μισά δεδομένα του qualifying set (γνωστό ως *quiz set*). Το αποτέλεσμα των προβλέψεων για τα υπόλοιπα δεδομένα (γνωστά ως *test set*) δεν ανακοινωνόταν, αλλά κρατούνταν κρυφό μέχρι το τέλος του διαγωνισμού και με βάση αυτά θα έβγαινε ο τελικός νικητής (Bennett and Lanning, 2007).

Η Netflix χρησιμοποιούσε ένα δικό της αλγόριθμο παραγωγής συστάσεων που ονομαζόταν *Cinematch*. Ο Cinematch μπορούσε να πετύχει RMSE 0,9514 στο quiz set και 0,9525 στο test set. Για να κερδίσει κάποιος το διαγωνισμό έπρεπε ο αλγόριθμος του να πετύχει τουλάχιστον κατά 10 % καλύτερο αποτέλεσμα RMSE από τον Cinematch. Επειδή ο αριθμός των ημερήσιων υποβολών που δεχόταν η σελίδα του διαγωνισμού ήταν περιορισμένος, η εταιρία επισήμανε ένα κομμάτι του training set, γνωστό ως *probe set* που είχε τις ίδιες στατιστικές ιδιότητες με το qualifying set, ώστε οι διαγωνιζόμενοι να μπορούν να κάνουν offline αξιολόγηση των αλγορίθμων τους.

Η σπουδαιότητα του διαγωνισμού του Netflix οφείλεται σε δύο παράγοντες. Ο πρώτος παράγοντας ήταν το ύψος του βραβείου. Όποιος κατάφερνε να ξεπεράσει σε ακρίβεια τον αλγόριθμο Cinematch τουλάχιστον κατά 10 %, θα κέρδιζε 1 εκατομμύριο δολάρια. Αυτό ήταν αρκετό ώστε να προσελκύσει πάνω από 51.000 διαγωνιζόμενους από 186 χώρες του κόσμου. Ο δεύτερος παράγοντας ήταν ότι ο νικητής, προκειμένου να πάρει το χρηματικό έπαθλο, έπρεπε να δημοσιεύσει λεπτομερώς τον αλγόριθμο του ώστε όλοι να ωφεληθούν από τις γνώσεις και τις τεχνικές που χρειάζονταν για την επίτευξη της βελτίωσης. Επίσης, κάθε χρόνο από την έναρξη του διαγωνισμού, ο καλύτερος έως τότε αλγόριθμος θα κέρδιζε ένα ενδιάμεσο έπαθλο των 50.000 δολαρίων, με την προϋπόθεση ότι θα πετύχαινε βελτίωση τουλάχιστον κατά 1 % από το καλύτερο αποτέλεσμα της προηγούμενης χρονιάς ή σε σχέση με τον Cinematch ειδικά για τον πρώτο χρόνο του διαγωνισμού. Αντίστοιχα, οι νικητές των ενδιάμεσων επάθλων ήταν επίσης υποχρεωμένοι να δημοσιεύσουν την προσέγγιση τους προκειμένου να λάβουν τα χρήματα.

[3] http://www.netflixprize.com/

2.4.2 Αποτελέσματα του διαγωνισμού

Το πρώτο ενδιάμεσο έπαθλο του 2007 κέρδισαν οι Yehuda Koren, Robert Bell και Chris Volinsky της ομάδας *BellCor*, πετυχαίνοντας αποτέλεσμα RMSE 0,8712, δηλαδή βελτίωση κατά 8,43 % σε σχέση με το Cinematch, δημοσιεύοντας ταυτόχρονα την προσέγγιση τους (Bell et al., 2007).

Το δεύτερο ενδιάμεσο έπαθλο κέρδισε το 2008, η ομάδα *"BellKor in Chaos"* η οποία προέκυψε από την συνένωση των ομάδων *BellKor* και *BigChaos*. Η βελτίωση άγγιξε το 9,44 % (RMSE 0,8616) και τα μέλη της ομάδας δημοσίευσαν τις μεθόδους τους (Bell et al., 2008; Töscher and Jahrer, 2008).

Το τέλος του διαγωνισμού ήταν τον Ιούλιο του 2009, όταν η ομάδα "*BellKor's Pragmatic Chaos*" κατάφερε να επιτύχει βελτίωση 10,05 % σε σχέση με τον Cinematch, με τελικό αποτέλεσμα στο test set RMSE 0,8567, κερδίζοντας έτσι το έπαθλο του ενός εκατομμυρίου δολαρίων. Η νικήτρια ομάδα προήλθε από την συνένωση των ομάδων *"BellKor in Chaos"* και *"Pragmatic Theory"* και τα μέλη της δημοσίευσαν τις μεθόδους τους (Koren, 2009; Piotte and Chabbert, 2009; Töscher et al., 2009). Αξίζει να σημειωθεί ότι η ομάδα *"The Ensemble"* που κατέκτησε την δεύτερη θέση, πέτυχε το ίδιο αποτέλεσμα RMSE στο test set, αλλά η υποβολή των προβλέψεων έγινε μόλις 20 λεπτά αργότερα σε σχέση με την νικήτρια ομάδα.

2.4.3 Σημασία του διαγωνισμού και μετέπειτα εξελίξεις

Όπως προαναφέρθηκε, ο διαγωνισμός για το βραβείο Netflix έγινε εξαιρετικά δημοφιλής και πυροδότησε την έρευνα στον τομέα της πρόβλεψης αξιολογήσεων με μεθόδους Συνεργατικού Φιλτραρίσματος. Η σχετική βιβλιογραφία στο διάστημα 2007- 2010 είναι εξαιρετικά πλούσια σε μεθόδους και προσεγγίσεις που χρησιμοποιήθηκαν κατά την διάρκεια του διαγωνισμού. Αξίζει να σημειωθεί ότι κατά τη διάρκεια του διαγωνισμού Netflix, εδραιώθηκαν οι τεχνικές Παραγοντοποίησης Πινάκων (Matrix Factorization) ως state-of-the-art για το πρόβλημα της πρόβλεψης αξιολογήσεων. Επιπλέον, η συλλογή δεδομένων Netflix χρησιμοποιήθηκε σε πολλές δημοσιεύσεις για την πειραματική επαλήθευση μεθόδων για συστήματα συστάσεων.

Αν και στη συλλογή δεδομένων Netflix δεν υπήρχαν καθόλου στοιχεία σχετικά με τους χρήστες και είχε κατασκευαστεί με τρόπο που να διατηρείται το απόρρητο των πελατών της εταιρίας, υπήρξαν κατηγορίες σχετικά την προστασία των δεδομένων. Το Νοέμβριο του 2007 δημοσιεύθηκε εργασία ερευνητών του Πανεπιστημίου του Τέξας που έδειξε ότι μπορούσαν να αναγνωρίσουν μεμονωμένους χρήστες της συλλογής δεδομένων Netflix, χρησιμοποιώντας στατιστικές μεθόδους και επιπλέον δεδομένα αξιολογήσεων από τον ιστοχώρο του Internet

Movie Database (IMDB)[4] και κατά συνέπεια να ανακαλύψουν πιθανώς ευαίσθητα προσωπικά δεδομένα (Narayanan and Shmatikov, 2007). Αυτό σε συνδυασμό με αγωγή που κατέθεσαν κάποιοι χρήστες του Netflix εναντίον της εταιρίας, είχε ως αποτέλεσμα την απόσυρση της συλλογής δεδομένων Netflix καθιστώντας την πλέον μη διαθέσιμη για χρήση στην έρευνα. Επιπλέον, μετά την λήξη του διαγωνισμού είχε ανακοινωθεί ότι θα ακολουθήσει νέος διαγωνισμός, αλλά τελικά ματαιώθηκε λόγω των νομικών προβλημάτων που αναφέρθηκαν προηγουμένως.

2.5 Μείωση διαστατικότητας

Τα δεδομένα που προέρχονται από τον πραγματικό κόσμο, συνήθως χαρακτηρίζονται από *υψηλή διαστατικότητα (high dimensionality)*. Η ύπαρξη πολλών διαστάσεων στα δεδομένα προκαλεί ποικίλα προβλήματα στην επεξεργασία τους, όπως για παράδειγμα η μέτρηση της απόστασης μεταξύ των σημείων (Domingos, 2012). Το πρόβλημα γίνεται εντονότερο σε δεδομένα πολύ υψηλών διαστάσεων (εκατοντάδων ή και χιλιάδων), αλλιώς γνωστό και ως *«κατάρα της διαστατικότητας» ("Curse of dimensionality")*, το οποίο οδήγησε στην ανάπτυξη διαφόρων τεχνικών *μείωσης διαστατικότητας (Dimensionality Reduction – DR)*.

Μείωση διαστατικότητας είναι ο μετασχηματισμός διανυσματικών δεδομένων που ανήκουν σε χώρο υψηλού αριθμού διαστάσεων, σε μια αναπαράσταση δεδομένων μειωμένων διαστάσεων, διατηρώντας όμως τη δομή και τις ιδιότητες των δεδομένων, με τρόπο που να επιτρέπει την περαιτέρω επεξεργασία τους. Πιο συγκεκριμένα, αν διαθέτουμε ένα πίνακα $\boldsymbol{X}$ με δεδομένα που ανήκουν σε ένα χώρο υψηλής διάστασης R^n, θέλουμε να τα μετασχηματίσουμε σε ένα νέο χώρο χαμηλότερης διάστασης R^k (όπου συνήθως $k << n$). (Van Der Maaten et al., 2009)

Η μείωση διαστατικότητας είναι σημαντική σε πολλούς τομείς, καθώς υποβοηθά πολλές άλλες τεχνικές εξόρυξης δεδομένων και μηχανικής μάθησης, όπως ταξινόμηση, συσταδοποίηση κ.α. Γενικότερα, τα πλεονεκτήματα της είναι ότι ελαττώνει τον απαιτούμενο χρόνο επεξεργασίας και χώρο αποθήκευσης των δεδομένων, καθώς και ότι πολλές φορές βελτιώνει την ακρίβεια αλγορίθμων μηχανικής μάθησης επειδή αφαιρεί ως ένα βαθμό τη συσχέτιση (correlation) μεταξύ των μεταβλητών. Για το λόγο αυτό είναι σύνηθες να εφαρμόζεται ως βήμα προεπεξεργασίας των δεδομένων πριν την εφαρμογή τεχνικών όπως αυτές που αναφέρθηκαν προηγουμένως. Επιπρόσθετα, βοηθά στην οπτικοποίηση (visualization) της δομής των δεδομένων όταν η μείωση γίνεται σε 2 ή 3 διαστάσεις.

[4] http://www.imdb.com/

2.5.1 Κατηγοριοποίηση τεχνικών μείωσης διαστατικότητας

Οι τεχνικές μείωσης διαστατικότητας μπορούν να ταξινομηθούν σε κυρτές (convex) και μη-κυρτές (non-convex). Οι κυρτές προσπαθούν να βελτιστοποιήσουν μια αντικειμενική συνάρτηση (objective function) που δεν περιέχει τοπικά βέλτιστα σημεία, σε αντίθεση με τις μη-κυρτές που η αντικειμενική τους συνάρτηση περιέχει τοπικά βέλτιστα (Van Der Maaten et al., 2009). Κυρτές τεχνικές είναι η Ανάλυση Κυρίων Συνιστωσών (PCA), η Kernel-PCA, η Isomap κ.α. Παραδείγματα μη-κυρτών τεχνικών είναι οι Autoencoders, η LLC και η t-SNE (Van der Maaten and Hinton, 2008).

Ένας άλλος τρόπος κατηγοριοποίησης είναι ο διαχωρισμός μεταξύ γραμμικών (linear) και μη-γραμμικών (non-linear) τεχνικών. Γραμμικές είναι οι τεχνικές που εκτελούν τη μείωση μετασχηματίζοντας τα δεδομένα σε γραμμικό χώρο, όπως η για παράδειγμα η PCA. Αντίθετα, οι μη-γραμμικές μπορούν να συλλάβουν τη δομή πιο περίπλοκων μη-γραμμικών δεδομένων. Εκτός από την PCA, όλες οι υπόλοιπες τεχνικές που αναφέρθηκαν πιο πάνω είναι μη-γραμμικές.

Μια άλλη κατηγορία αλγορίθμων μείωσης διαστατικότητας είναι τα Μοντέλα Λανθανόντων Παραγόντων (Latent Factor Models) ή αλλιώς μέθοδοι Παραγοντοποίησης (Factorization Methods). Οι συγκεκριμένοι αλγόριθμοι πετυχαίνουν μείωση διαστατικότητας αναλύοντας τον αρχικό πίνακα δεδομένων σε γινόμενο πινάκων χαμηλότερου βαθμού (rank). Ένας από τους γνωστότερους αλγορίθμους της κατηγορίας αυτής είναι η Παραγοντοποίηση Ιδιαζουσών τιμών (Singular Value Decomposition – SVD).

2.5.2 Ανάλυση Κυρίων Συνιστωσών (PCA)

Μια από τις δημοφιλέστερες τεχνικές μείωσης διαστατικότητας είναι η Ανάλυση Κυρίων Συνιστωσών (Principal Component Analysis – PCA) (Abdi and Williams, 2010). Ο στόχος της PCA είναι να εξάγει την πιο σημαντική πληροφορία από τα αρχικά δεδομένα, να συμπιέσει το μέγεθος των δεδομένων κρατώντας μόνο την προαναφερθείσα πιο σημαντική πληροφορία και να αναλύσει την δομή των δεδομένων.

Αναλυτικότερα, η PCA κατασκευάζει μια αναπαράσταση δεδομένων σε γραμμικό χώρο χαμηλών διαστάσεων η οποία διατηρεί την μέγιστη διακύμανση (variance) των αρχικών πολυδιάστατων δεδομένων. Πιο συγκεκριμένα υπολογίζει νέες μεταβλητές που ονομάζονται Κύριες Συνιστώσες, οι οποίες αποτελούν γραμμικούς συνδυασμούς των αρχικών μεταβλητών. Η πρώτη Κύρια Συνιστώσα υπολογίζεται έτσι ώστε να περιγράφει την μέγιστη δυνατή διακύμανση των αρχικών δεδομένων. Η δεύτερη Κύρια Συνιστώσα υπολογίζεται υπό τον περιορισμό να είναι ορθογώνια σε σχέση με την πρώτη και επίσης να έχει την αμέσως

επόμενη μέγιστη δυνατή διακύμανση. Οι υπόλοιπες συνιστώσες υπολογίζονται με ανάλογο τρόπο.

Η PCA χρησιμοποιήθηκε στο παρελθόν σε συστήματα συστάσεων (Goldberg et al., 2001), όμως πλέον προτιμούνται περισσότερο οι Μέθοδοι Παραγοντοποίησης (Amatriain and Pujol, 2015).

2.5.3 Μέθοδοι παραγοντοποίησης (Factorization Methods)

Οι Μέθοδοι παραγοντοποίησης προσπαθούν να αντιμετωπίσουν το πρόβλημα της αραιότητας των αξιολογήσεων μετασχηματίζοντας το αρχικό χώρο των δεδομένων σε ένα λανθάνων χώρο μειωμένων διαστάσεων στον οποίο αποτυπώνονται τα βασικά χαρακτηριστικά των χρηστών και των αντικειμένων. Αν οι χρήστες και τα αντικείμενα συγκριθούν μεταξύ τους σε αυτόν τον πυκνότερο χώρο, είναι πιθανόν να ανακαλυφθούν σχέσεις που δεν είναι δυνατόν να βρεθούν στον αρχικό χώρο.

Στη συνέχεια παρουσιάζεται η γνωστότερη προσέγγιση στα πλαίσια των συστημάτων συστάσεων: Η Παραγοντοποίηση Ιδιαζουσών τιμών (Singular Value Decomposition - SVD). Στην μέθοδο SVD βασίζεται μια οικογένεια μεθόδων με την γενική ονομασία Μέθοδοι Παραγοντοποίησης Πινάκων (Matrix Factorization). Μερικές δημοφιλείς μέθοδοι Παραγοντοποίησης Πινάκων παρουσιάζονται στο Κεφάλαιο 3 της παρούσας εργασίας.

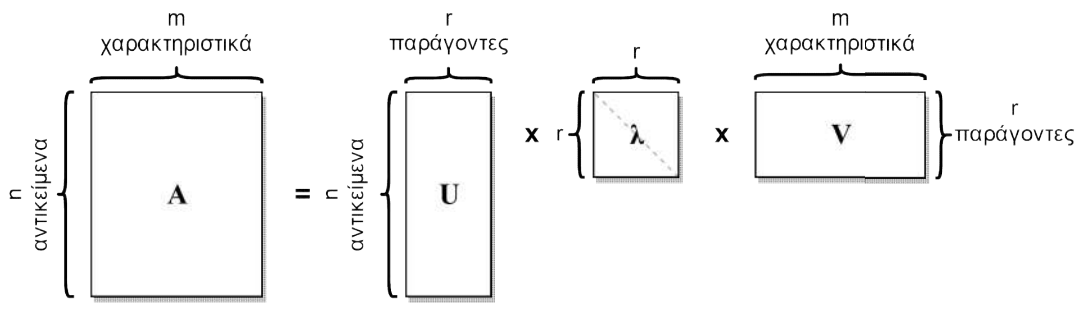
2.5.3.1 Παραγοντοποίηση Ιδιαζουσών τιμών (Singular Value Decomposition - SVD)

Η Παραγοντοποίηση Ιδιαζουσών τιμών (Singular Value Decomposition - SVD) (Golub and Reinsch, 1970) είναι μια μαθηματική μέθοδος που έχει ως σκοπό να βρει ένα χώρο χαμηλότερων διαστάσεων, στον οποίο τα χαρακτηριστικά θα αναπαριστούν έννοιες (concepts) ή λανθάνοντες παράγοντες (latent factors).

Η τεχνική βασίζεται στο εξής θεώρημα: Κάθε πίνακας A μπορεί να αναλυθεί σε ένα γινόμενο πινάκων της μορφής:

$$A = U\,\lambda\,V^T$$

όπου A είναι ο δεδομένος πίνακας διαστάσεων $n \times m$, δηλαδή n αντικείμενα και m χαρακτηριστικά, U ένας πίνακας $n \times r$ (n αντικείμενα, r παράγοντες), λ ένας διαγώνιος πίνακας $r \times r$ (βαρύτητα κάθε παράγοντα) και V ένας πίνακας με m χαρακτηριστικά και r παράγοντες. Η ανάλυση παρουσιάζεται σχηματικά στο Σχήμα 11.



Σχήμα 11. Σχηματική αναπαράσταση της Παραγοντοποίησης Ιδιαζουσών τιμών (SVD). Ένας πίνακας n x m αναλύεται σε γινόμενο τριών πινάκων: n x r, r x r και r x m

Ο διαγώνιος πίνακας λ περιέχει τις *ιδιάζουσες τιμές (singular values)* του πίνακα A, που είναι πάντα θετικές και εμφανίζονται σε φθίνουσα σειρά. Ο πίνακας U μπορεί να θεωρηθεί ως πίνακας ομοιότητας αντικειμένων-παραγόντων και αντίστοιχα ο πίνακας V ως πίνακας ομοιότητας χαρακτηριστικών-παραγόντων.

Όπως αναφέρθηκε, οι r ιδιάζουσες τιμές του πίνακα λ είναι κατά φθίνουσα σειρά. Έτσι, αν διατηρηθούν μόνο οι k μεγαλύτερες τιμές του λ, μπορεί να κατασκευαστεί μια περικομμένη (truncated) SVD εκδοχή του αρχικού πίνακα A. Η παραλλαγή truncatedSVD δημιουργεί λοιπόν μια προσέγγιση k βαθμού του πίνακα A, ώστε να ισχύει:

$$A_k = U_k \lambda_k V_k^T$$

Έτσι, κατασκευάζεται μια αναπαράσταση που συλλαμβάνει την λανθάνουσα πληροφορία, μειώνει τον θόρυβο των χαρακτηριστικών και βρίσκεται σε μειωμένο χώρο k διαστάσεων, όπου συνήθως $k \ll r$ (Amatriain and Pujol, 2015).

2.5.3.2 Παραγοντοποίηση πινάκων (Matrix Factorization)

Η εφαρμογή του της μεθόδου SVD σε δεδομένα αξιολογήσεων χρηστών παρουσιάζει δυσκολίες λόγω της αραιότητας των δεδομένων, καθώς η μαθηματική λύσης της δεν ορίζεται όταν υπάρχουν άγνωστες τιμές. Έχουν προταθεί τεχνικές συμπλήρωσης των άγνωστων τιμών, π.χ. με την μέση τιμή των αξιολογήσεων ώστε να εξαλειφθεί η αραιότητα, όμως μια τέτοια προσέγγιση μπορεί να είναι υπολογιστικά ακριβή, ειδικά σε μεγάλα σύνολα δεδομένων. Επιπλέον ενέχει ο κίνδυνος αλλοίωσης των αρχικών δεδομένων, οδηγώντας σε ανακριβή αποτελέσματα (Koren and Bell, 2015).

Για τους λόγους που αναφέρθηκαν παραπάνω, έχουν προταθεί άλλες τεχνικές που βασίζονται στην SVD, που έχουν δηλαδή ως αποτέλεσμα την ανάλυση του αρχικού πίνακα αξιολογήσεων σε δύο πίνακες, όπου ο ένας περιέχει τα χαρακτηριστικά των χρηστών και ο άλλος τα χαρακτηριστικά των αντικειμένων. Είναι γνωστές με την γενική ονομασία Παραγοντοποίηση Πινάκων (Matrix Factorization – MF) και μπορούν να χειριστούν αραιούς πίνακες δεδομένων, καθώς λαμβάνουν υπ' όψη τους μόνο τις γνωστές αξιολογήσεις. Πολλές

από τις τεχνικές MF χρησιμοποιούν τον αλγόριθμο Alternating Least Squares (ALS) ή Stochastic Gradient Descent (SGD) για να ανακαλύψουν το ζητούμενο χαμηλοδιάστατο χώρο χαρακτηριστικών (Koren et al., 2009). Παραδείγματα γνωστών τεχνικών MF είναι οι Regularized SVD, Non-negative Matrix Factorization (NMF), FunkSVD, SVD++ και RISMF.

2.6 Άλλες σχετικές εργασίες

2.6.1 Συνεργατικό Φιλτράρισμα

Η γνωστότερη ίσως βιβλιογραφική επισκόπηση σχετικά με Συνεργατικό Φιλτράρισμα αλλά και για μεθόδους Συστημάτων Συστάσεων είναι η (Adomavicius and Tuzhilin, 2005). Αποτελεί μια εκτενή αναφορά σε μεθόδους αλλά και τις προκλήσεις και προβλήματα που αφορούν τον τομέα των συστημάτων συστάσεων.

Μια βιβλιογραφική επισκόπηση ειδικά για μεθόδους συνεργατικού φιλτραρίσματος είναι η (Su and Khoshgoftaar, 2009). Αναφέρεται στις τρεις βασικές κατηγορίες τεχνικών συνεργατικού φιλτραρίσματος: βασισμένες στη μνήμη, βασισμένες σε μοντέλα και υβριδικές, αναφέροντας παραδείγματα αλγορίθμων για κάθε κατηγορία, καθώς και πλεονεκτήματα και μειονεκτήματα κάθε προσέγγισης. Δεν περιλαμβάνει πειραματική αξιολόγηση των αλγορίθμων.

Μια ενδιαφέρουσα προσέγγιση συνεργατικού φιλτραρίσματος προτείνεται από τους (Braida et al., 2015). Πρόκειται για ένα framework για τον μετασχηματισμό ενός πίνακα αξιολογήσεων σε μορφή κατάλληλη για εκπαίδευση μεθόδων μηχανικής μάθησης με επίβλεψη. Η μέθοδος τους προσπαθεί να εξάγει χαρακτηριστικά χρηστών και αντικειμένων από τον πίνακα αξιολογήσεων, χρησιμοποιώντας μέθοδο SVD. Στη συνέχεια κατασκευάζει ένα νέο χώρο χαρακτηριστικών, όπου κάθε αξιολόγηση αναπαρίσταται από τα χαρακτηριστικά που εξήχθησαν προηγουμένως. Τέλος, εκπαιδεύουν αλγορίθμους μηχανικής μάθησης όπως νευρωνικά δίκτυα και Random Forests για να προβλέψουν τις αξιολογήσεις των χρηστών.

2.6.2 Πειραματική σύγκριση απόδοσης αλγορίθμων

Οι εργασίες με θέμα την σύγκριση της απόδοσης αλγορίθμων συνεργατικού φιλτραρίσματος είναι δυσεύρετες, δεν ακολουθούν συγκεκριμένες μεθοδολογίες αξιολόγησης της απόδοσης και χρησιμοποιούν διαφορετικές συλλογές δεδομένων. Έτσι η σύγκριση των αποτελεσμάτων μεταξύ των εργασιών δεν είναι εύκολη.

Μια πρώιμη σύγκριση μεταξύ αλγορίθμων βασισμένων στη μνήμη γίνεται στην (Herlocker et al., 1999), τους οποίους ονομάζει αλγορίθμους βασισμένους στη γειτονιά (neighborhood-based). Αξιολογούνται διάφορες μετρικές ομοιότητας πάνω σε δεδομένα αξιολογήσεων για ταινίες και ως μετρική ακρίβειας προβλέψεων χρησιμοποιείται η Mean Absolute Error (MAE).

Οι (Fisher et al., 2000) παρουσίασαν το SWAMI, ένα framework σε Java, το οποίο επιτρέπει την αξιολόγηση της απόδοσης αλγορίθμων συνεργατικού φιλτραρίσματος. Συγκρίνουν τρεις μεθόδους: ένα εκτιμητή με βάση τη συσχέτιση Pearson, ένα εκτιμητή που βασίζεται σε μηχανή διανυσμάτων υποστήριξης και ένα εκτιμητή με βάση τη συσχέτιση Pearson και συσταδοποίηση των χρηστών με k-means. Οι μέθοδοι εφαρμόζονται σε δεδομένα αξιολογήσεων για ταινίες και ως μετρική ακρίβειας προβλέψεων χρησιμοποιείται η MAE.

Στην εργασία (González-Caro et al., 2002) γίνεται η πρώτη εκτενής πειραματική σύγκριση 5 αλγορίθμων συνεργατικού φιλτραρίσματος σε δύο συλλογές δεδομένων από διαφορετικά πεδία. Οι αλγόριθμοι είναι: μέθοδοι βασισμένοι στη μνήμη, Dependency Networks, Aspect Model, Online Learning και Support Vector Machines και χρησιμοποιούνται οι μετρικές ακρίβειας MAE και ROC. Όμως τα δεδομένα αξιολογήσεων μετατράπηκαν σε δυαδικά δεδομένα (αρέσει – δεν αρέσει), άρα η σύγκριση είναι μεταξύ ταξινομητών.

Η εργασία (Calderón-Benavides et al., 2004) συγκρίνει τους 4 από τους 5 αλγορίθμους που αξιολόγησε η προηγούμενη εργασία, αλλά με βάση τις αρχικές τιμές των κλιμάκων αξιολόγησης και όχι δυαδικές. Επίσης, χρησιμοποιούνται τρεις συλλογές δεδομένων με διαφορετικό βαθμό αραιότητας και τέσσερις μετρικές ποιότητας των αλγορίθμων.

Η πρώτη εκτενής εργασία σύγκρισης αλγορίθμων στην οποία περιλαμβάνονται μέθοδοι μείωσης διαστατικότητας είναι η (Huang et al., 2007). Συνολικά συγκρίνονται έξι αλγόριθμοι συνεργατικού φιλτραρίσματος, όμως επειδή η εργασία απευθύνεται στον τομέα του e-commerce, η σύγκριση αφορά σύσταση N-αντικειμένων και όχι πρόβλεψη αξιολογήσεων. Χρησιμοποιούνται μετρικές κατάταξης όπως Precision, Recall κ.α. και η αξιολόγηση γίνεται σε τρεις συλλογές δεδομένων σχετικές με e-commerce: ενδυμασία, βιβλία και ταινίες, από τις οποίες οι δύο δεν είναι ελεύθερα διαθέσιμες.

Η πιο εκτενής και λεπτομερής μέχρι στιγμής εργασία σύγκρισης αλγορίθμων συνεργατικού φιλτραρίσματος είναι πιθανόν η (Cacheda et al., 2011). Αξιολογούνται τρεις προσεγγίσεις βασισμένες στη μνήμη (User-based, Item-based και Similarity-Fusion) και έξι προσεγγίσεις βασισμένες σε μοντέλα (Regression-based, Slope one, SVD, Regularized SVD, Integrated neighbor-based – SVD model και Cluster-based smoothing. Χρησιμοποιούνται δύο συλλογές με αξιολογήσεις ταινιών, εκ των οποίων η μία είναι η συλλογή Netflix. Επίσης χρησιμοποιούνται αρκετές μετρικές συμπεριλαμβανομένων και MAE, RMSE, ROC.

Μια επίσης εκτενής σύγκριση παρουσιάζεται στην εργασία (Lee et al., 2012) στην οποία συγκρίνονται συνολικά 12 προσεγγίσεις συνεργατικού φιλτραρίσματος: 4 μέθοδοι βασισμένες στη μνήμη και 8 μέθοδοι βασισμένες σε μοντέλα (εκ των οποίων 5 μέθοδοι βασίζονται σε παραγοντοποίηση πινάκων). Η συλλογή δεδομένων που χρησιμοποιείται είναι η Netflix και η αξιολόγηση γίνεται με γνώμονα την πυκνότητα των αξιολογήσεων, δηλαδή γίνεται προεπεξεργασία των δεδομένων ώστε να δοκιμαστούν περιπτώσεις με διαφορετικό αριθμό αξιολογήσεων, χρηστών και αντικειμένων. Οι βασικές μετρικές ακρίβειας είναι η MAE και η RMSE.

Οι υλοποιήσεις αλγορίθμων παραγωγής συστάσεων μπορεί να διαφέρουν ανάλογα με τη βιβλιοθήκη ή framework στο οποίο περιέχονται. Αντίστοιχα και οι μεθοδολογίες αξιολόγησης των αλγορίθμων μπορεί να έχουν σημαντικές διαφορές, καθιστώντας την σύγκριση της απόδοσης μεταξύ αλγορίθμων και την αναπαραγωγή των συγκεκριμένων πειραμάτων μια κοπιώδη διαδικασία. Στην εργασία (Said and Bellogín, 2014) παρουσιάζεται ένα framework σχεδιασμένο για την αξιολόγηση συστημάτων συστάσεων, που ονομάζεται RiVal και συγκρίνονται αλγόριθμοι User-based, Item-based και Matrix Factorization που παρέχονται από 3 frameworks συνεργατικού φιλτραρίσματος: *Apache Mahout* (Anil et al., 2010), *LensKit* (Ekstrand et al., 2011) και *MyMediaLite* (Gantner et al., 2011). Χρησιμοποιούνται 3 συλλογές δεδομένων και η αξιολόγηση της ακρίβειας προβλέψεων γίνεται με την μετρική RMSE.

3

Πρόβλεψη αξιολογήσεων με Συνεργατικό Φιλτράρισμα

Η παρούσα εργασία ασχολείται με το πρόβλημα του Συνεργατικού Φιλτραρίσματος και συγκεκριμένα με την πρόβλεψη αξιολογήσεων που μπορεί να δώσουν οι χρήστες σε αντικείμενα που δεν έχουν αξιολογήσει ακόμα.

3.1 Μοντελοποίηση εννοιών

Ο ορισμός του προβλήματος του Συνεργατικού Φιλτραρίσματος βασίζεται στην μοντελοποίηση των παρακάτω εννοιών (Takács et al., 2009):

Ορίζεται μια τριάδα τυχαίων μεταβλητών (U, I, R), όπου:

- Η τυχαία μεταβλητή U παίρνει τιμές από το διάστημα $\{1, \dots, N\}$ (N είναι ο αριθμός των χρηστών) και ονομάζεται *αναγνωριστικό χρήστη (userID)*.
- Η τυχαία μεταβλητή I παίρνει τιμές από το διάστημα $\{1, \dots, M\}$ (M είναι ο αριθμός των αντικειμένων) και ονομάζεται *αναγνωριστικό αντικειμένου (itemID)*.
- Η τυχαία μεταβλητή R είναι η τιμή αξιολόγησης και παίρνει τιμές από ένα σύνολο $X \subset \mathbb{R}$. Συνήθεις τιμές αξιολόγησης μπορεί να είναι δυαδικές ($X = \{0,1\}$), ακέραιοι αριθμοί ενός συγκεκριμένου εύρους (π.χ. $X = \{1, 2, 3, 4, 5\}$) ή ακόμα και πραγματικοί αριθμοί που ανήκουν σε ένα κλειστό διάστημα (π.χ. $X = [-10, 10]$).

Για να αναφερθούμε σε συγκεκριμένες τιμές της τριάδας (U, I, R), χρησιμοποιούμε τον συμβολισμό (u, i, r) που σημαίνει ότι ο χρήστης u αξιολόγησε το αντικείμενο i με την τιμή αξιολόγησης r.

3.2 Ορισμός του προβλήματος

Ο σκοπός της πρόβλεψης αξιολογήσεων είναι να προβλέψουμε το R με βάση τα (U, I) έτσι ώστε η απόκλιση των προβλέψεων από τις πραγματικές τιμές να είναι η ελάχιστη δυνατή. Μια μετρική ακρίβειας προβλέψεων που χρησιμοποιείται συχνά είναι η Ρίζα του Μέσου Τετραγωνικού Σφάλματος (RMSE) (Παρ. 3.5.2.2, σελ. 56):

$$RMSE = \sqrt{\boldsymbol{E}\left\{\left(\hat{R} - R\right)^2\right\}} \tag{3.1}$$

όπου $\hat{R}$ είναι η πρόβλεψη του R και ο συμβολισμός $E\{\}$ είναι η μέση τιμή. Σημειώνεται ότι μπορούν να χρησιμοποιηθούν και άλλες μετρικές ακρίβειας (Παρ. 3.5.2).

Στην πράξη, η κατανομή της τριάδας (U, I, R) δεν είναι γνωστή, αλλά διαθέτουμε μόνο ένα πεπερασμένο δείγμα δεδομένων $\mathcal{R}'$ μήκους t που προήλθε από την κατανομή αυτή:

$$\mathcal{R}' = \{\, (u_1, i_1, r_1), (u_2, i_2, r_2), \ldots, (u_t, i_t, r_t) \,\}$$

Το δείγμα $\mathcal{R}'$ μπορεί να χρησιμοποιηθεί για την εκπαίδευση εκτιμητών. Το σύνολο των δυάδων *(userID, itemID)* μπορεί να συμβολιστεί ως $\mathcal{R} = \{(u, i) : \exists r : (u, i, r) \in R'\}$.

Το δείγμα $\mathcal{R}'$ μπορεί να αναπαρασταθεί ως ένας μερικώς ορισμένος πίνακας $\boldsymbol{R} \in \mathbb{R}^{N \times M}$, που τα στοιχεία του είναι γνωστά στις θέσεις $(u, i) \in \mathcal{R}$, αλλά άγνωστα στις θέσεις $(u, i) \notin \mathcal{R}$. Η τιμή κάθε γνωστού στοιχείου του πίνακα $\boldsymbol{R}$ στην θέση $(u, i) \in \mathcal{R}$, συμβολίζεται με $r_{u,i}$ και περιέχει την αξιολόγηση του χρήστη u για το αντικείμενο i.

Όπως αναφέρθηκε παραπάνω, ο σκοπός του Συνεργατικού Φιλτραρίσματος είναι η εκπαίδευση εκτιμητών που ελαχιστοποιούν το σφάλμα πρόβλεψης (Εξ. (3.1)). Στην πράξη όμως αυτό το σφάλμα δεν μπορεί να μετρηθεί, καθώς η κατανομή των τριάδων (U, I, R) είναι άγνωστη. Αντ' αυτού, μπορεί να εκτιμηθεί το σφάλμα σε ένα σύνολο επικύρωσης το οποίο μπορεί να προέλθει με τυχαίο καταμερισμό του δείγματος $\mathcal{R}'$ σε ένα νέο μικρότερο σύνολο εκπαίδευσης και στο εν λόγω σύνολο επικύρωσης. Αν ορίσουμε το σύνολο επικύρωσης ως V' και τις δυάδες *(userID, itemID)* του συνόλου V' ως $V = \{(u, i) : \exists r : (u, i, r) \in V'\}$, τότε η εκτίμηση του σφάλματος RMSE στο σύνολο V' μπορεί να υπολογιστεί ως εξής:

$$\widehat{RMSE} = \sqrt{\frac{1}{|V'|} \sum_{(u,i,r) \in V'} \left(\hat{r}_{u,i} - r_{u,i}\right)^2} \tag{3.2}$$

3.3 Αλγόριθμοι Συνεργατικού Φιλτραρίσματος

3.3.1 Βασικοί Εκτιμητές (Baseline Predictors)

Το Συνεργατικό Φιλτράρισμα προσπαθεί να μοντελοποιήσει τις αλληλεπιδράσεις μεταξύ χρηστών και αντικειμένων οι οποίες έχουν ως αποτέλεσμα τις εκάστοτε τιμές αξιολόγησης. Παρόλα αυτά μπορεί να υπάρχουν και άλλοι παράγοντες που επηρεάζουν τις αξιολογήσεις που δίνουν οι χρήστες, οι οποίοι δεν εξαρτώνται από τις αλληλεπιδράσεις.

Ένα φαινόμενο το οποίο επηρεάζει τις αξιολογήσεις είναι η *πόλωση (bias)*, δηλαδή για παράδειγμα κάποιοι χρήστες μπορεί να έχουν την τάση να αξιολογούν υψηλά σε γενικές γραμμές, ενώ αντίθετα κάποιοι άλλοι να είναι ιδιαίτερα φειδωλοί στις αξιολογήσεις τους. Αντίστοιχα, κάποια αντικείμενα μπορεί να λαμβάνουν γενικά υψηλότερες αξιολογήσεις από άλλα, γιατί για παράδειγμα είναι πολύ δημοφιλή.

Επειδή οι πολώσεις μπορεί να παίζουν μεγάλο ρόλο στην διαμόρφωση των αξιολογήσεων, έχει νόημα να μοντελοποιηθούν ξεχωριστά, ώστε η επίδραση τους να διαχωριστεί από τις πραγματικές αλληλεπιδράσεις μεταξύ χρηστών και αντικειμένων. Τα μοντέλα αυτά ονομάζονται *Βασικοί Εκτιμητές (Baseline Predictors)* ή πιο απλά *πολώσεις (biases)* (Koren, 2008).

Ο απλούστερος βασικός εκτιμητής είναι η μέση τιμή όλων των διαθέσιμων αξιολογήσεων *(Global Average Baseline)*. Επίσης, ένας άλλο βασικός εκτιμητής είναι η μέση τιμή των αξιολογήσεων του κάθε χρήστη *(User Average Baseline)* και αντίστοιχα μπορεί να μοντελοποιηθεί και ο βασικός εκτιμητής κάθε αντικειμένου, ως η μέση τιμή όλων των αξιολογήσεων που έλαβε *(Item Average Baseline)*.

Μια άλλη προσέγγιση μοντελοποίησης των biases είναι να συνδυαστούν όλα τα παραπάνω σε ένα ενιαίο μοντέλο βασικού εκτιμητή χρηστών-αντικειμένων. Για το σκοπό αυτό υπολογίζουμε τις αποκλίσεις του User Baseline και Item Baseline από το Global Baseline και το τελικό μοντέλο User- Item Baseline που τα συνδυάζει δίνεται από την παρακάτω εξίσωση:

$$b_{u,i} = \mu + b_u + b_i \qquad (3.3)$$

όπου $b_{u,i}$ είναι η βασική πρόβλεψη για μια άγνωστη αξιολόγηση $r_{u,i}$, μ είναι η μέση τιμή όλων των γνωστών αξιολογήσεων και οι παράμετροι b_u και b_i είναι αποκλίσεις των γνωστών αξιολογήσεων του χρήστη u και του αντικειμένου i από το μ, αντίστοιχα.

Το μοντέλο μπορεί να χρησιμοποιηθεί ως εξής: Ας υποτεθεί ότι σε ένα σύνολο δεδομένων με αξιολογήσεις χρηστών για ταινίες και κλίμακα αξιολόγησης από 1 έως 5 αστέρια, το μ είναι 3,7 αστέρια και θέλουμε να βρούμε την βασική πρόβλεψη του χρήστη X για την ταινία «Τιτανικός». Επειδή ο Τιτανικός είναι «καλή» ταινία αξιολογήθηκε κατά μέσο όρο 0,5

αστέρια περισσότερο από το μ. Όμως, ο X είναι κριτικός χρήστης και κατά μέσο όρο βαθμολογεί $0{,}3$ αστέρια λιγότερο από το μ. Έτσι η βασική πρόβλεψη για την αξιολόγηση του «Τιτανικού» από τον χρήστη X υπολογίζεται ως εξής: $3{,}7 + 0{,}5 - 0{,}3 = 3{,}9$ αστέρια.

Η συνάρτηση κόστους που πρέπει να ελαχιστοποιηθεί για το μοντέλο του βασικού εκτιμητή User - Item Baseline δίνεται από την σχέση:

$$\min_{b*} \sum_{(u,i)\in\mathcal{R}} \left(r_{u,i} - (\mu + b_u + b_i)\right)^2 \tag{3.4}$$

Η συνάρτηση κόστους προσπαθεί να βρει τις παραμέτρους b_u και b_i και μπορεί να ελαχιστοποιηθεί είτε χρησιμοποιώντας τον αλγόριθμο κατάβασης δυναμικού (Παρ. 2.2.1), είτε με την μέθοδο Alternating Least Squares (Παρ 3.3.3.2).

Επειδή συνήθως τα δεδομένα αξιολογήσεων είναι αραιά (sparse), υπάρχει ο κίνδυνος υπερπροσαρμογής (overfitting) του μοντέλου στα δεδομένα εκπαίδευσης. Για την αποφυγή αυτού του προβλήματος, στη συνάρτηση κόστους μπορεί να προστεθεί και ένας όρος regularization για να περιορίσει το εύρος των παραμέτρων. Έτσι η τελική συνάρτηση κόστους με regularization θα πάρει την μορφή:

$$\min_{b*} \sum_{(u,i)\in\mathcal{R}} \left(r_{u,i} - (\mu + b_u + b_i)\right)^2 + \lambda\left(\sum_u b_u^2 + \sum_i b_i^2\right) \tag{3.5}$$

όπου λ είναι ο συντελεστής regularization.

Καθώς υπάρχει η περίπτωση τα πλήθη χρηστών και αντικειμένων να είναι πολύ διαφορετικά μεταξύ τους, υπάρχει η δυνατότητα να χρησιμοποιηθούν ξεχωριστοί συντελεστές για τους χρήστες και τα αντικείμενα, δίνοντας έτσι καλύτερο έλεγχο της πολυπλοκότητας του μοντέλου. Άρα η συνάρτηση κόστους με ξεχωριστούς συντελεστές regularization διαμορφώνεται ως εξής:

$$\min_{b*} \sum_{(u,i)\in\mathcal{R}} \left(r_{u,i} - (\mu + b_u + b_i)\right)^2 + \lambda_u \sum_u b_u^2 + \lambda_i \sum_i b_i^2 \tag{3.6}$$

όπου λ_u είναι ο συντελεστής regularization για τα biases των χρηστών και λ_i είναι ο συντελεστής regularization για τα biases των αντικειμένων.

Οι βασικοί εκτιμητές είναι οι απλούστερες προσεγγίσεις για το πρόβλημα της πρόβλεψης αξιολογήσεων και ως εκ τούτου η ακρίβεια τους είναι σχετικά χαμηλή σε σχέση με άλλες προσεγγίσεις. Έτσι η χρησιμότητα τους ως μεμονωμένες προσεγγίσεις έγκειται μόνο στην μεταχείριση τους ως ελάχιστο μέτρο ακρίβειας προβλέψεων για ένα συγκεκριμένο σύνολο δεδομένων. Με άλλα λόγια οποιαδήποτε άλλη μέθοδος, για να θεωρηθεί αξιοπρεπής πρέπει να έχει τουλάχιστον μεγαλύτερη ακρίβεια από τους βασικούς εκτιμητές. Παρόλα αυτά, μια

άλλη πρακτική εφαρμογή των βασικών εκτιμητών είναι η χρήση τους σε περιπτώσεις που ένας αλγόριθμος δεν μπορεί να παράγει προβλέψεις λόγω του προβλήματος της ψυχρής εκκίνησης (cold-start problem), δηλαδή όταν για κάποιο χρήστη ή αντικείμενο δεν υπάρχουν αρκετά δεδομένα ώστε να παραχθούν προβλέψεις.

3.3.2 Μοντέλα Παραγοντοποίησης Πινάκων (Matrix Factorization)

Τα μοντέλα Παραγοντοποίησης Πινάκων ανήκουν στην γενική κατηγορία των Μοντέλων Λανθανόντων Παραγόντων (Latent Factor Models), που έχουν ως σκοπό να ανακαλύψουν στα δεδομένα, τους λανθάνοντες παράγοντες (Latent Factors) που επηρεάζουν την διαμόρφωση τους. Άλλες προσεγγίσεις αυτής της κατηγορίας είναι τα Νευρωνικά Δίκτυα (Salakhutdinov et al., 2007) και η Λανθάνουσα Κατανομή Dirichlet (Latent Dirichlet Allocation) (Blei et al., 2003).

Όπως αναφέρθηκε στην Παρ. 2.5.3.2 (σελ. 38), η μέθοδος SVD δεν μπορεί να εφαρμοστεί σε δεδομένα που περιέχουν άγνωστες τιμές. Για το λόγο αυτό, στα πλαίσια του διαγωνισμού Netflix Prize, προτάθηκαν αρκετές μέθοδοι που βασίζονται στην μέθοδο SVD, αλλά χρησιμοποιούν μόνο τις γνωστές αξιολογήσεις, αποφεύγοντας παράλληλα τα φαινόμενα υπερπροσαρμογής με μεθόδους regularization.

Τα μοντέλα Παραγοντοποίησης Πινάκων, όπως και η μέθοδος SVD, προσπαθούν να προβάλλουν τους χρήστες και τα αντικείμενα σε ένα λανθάνων χώρο χαμηλότερων διαστάσεων, όπου οι αλληλεπιδράσεις μεταξύ χρηστών και αντικειμένων, αναπαριστώνται ως εσωτερικά γινόμενα μεταξύ των λανθανόντων παραγόντων των χρηστών και των αντικειμένων. Οι λανθάνοντες παράγοντες μπορεί να αναφέρονται σε διάφορες έννοιες. Για παράδειγμα, αν τα αντικείμενα είναι ταινίες, ένας παράγοντας μπορεί να αντιπροσωπεύει αν πρόκειται για ταινία δράσης ή κωμωδία, αν περιλαμβάνει βία ή όχι και γενικά οτιδήποτε θα μπορούσε να χαρακτηρίσει μια ταινία σε μια βαθμιαία κλίμακα. Παρομοίως για ένα χρήστη, ένας παράγοντας μπορεί να αναφέρεται στο αν προτιμά ταινίες δράσης ή κωμωδίες, αν του αρέσουν ταινίες που περιλαμβάνουν βία κ.ο.κ. Τελικά, η κάθε αξιολόγηση ταινίας από ένα χρήστη, αναλύεται σε ένα εσωτερικό γινόμενο των αντίστοιχων παραγόντων του χρήστη και της ταινίας. Η ανάλυση αυτή πραγματοποιείται αυτόματα κατά την εκπαίδευση του μοντέλου.

3.3.3 Παραγοντοποίηση Πινάκων τύπου SVD (SVD-based Matrix Factorization)

Η Παραγοντοποίηση Πινάκων τύπου SVD βασίζεται στην κλασική μέθοδο SVD, αλλά για να αντιμετωπίσει την αραιότητα των δεδομένων εκτελείται επαναληπτικά και χρησιμοποιεί

κάποιο αλγόριθμο βελτιστοποίησης όπως θα δούμε παρακάτω. Η προσέγγιση αυτή έχει τη γενική ονομασία *Matrix Factorization* (*MF*) και υπάρχουν διάφορες παραλλαγές της.

Αν διαθέτουμε ένα πίνακα αξιολογήσεων $\boldsymbol{R}$ με N γραμμές και M στήλες (δηλαδή N χρήστες και M αντικείμενα), μπορούμε να κατασκευάσουμε μια προσέγγιση του $\widehat{\boldsymbol{R}}$ ως γινόμενο δύο μικρότερων πινάκων:

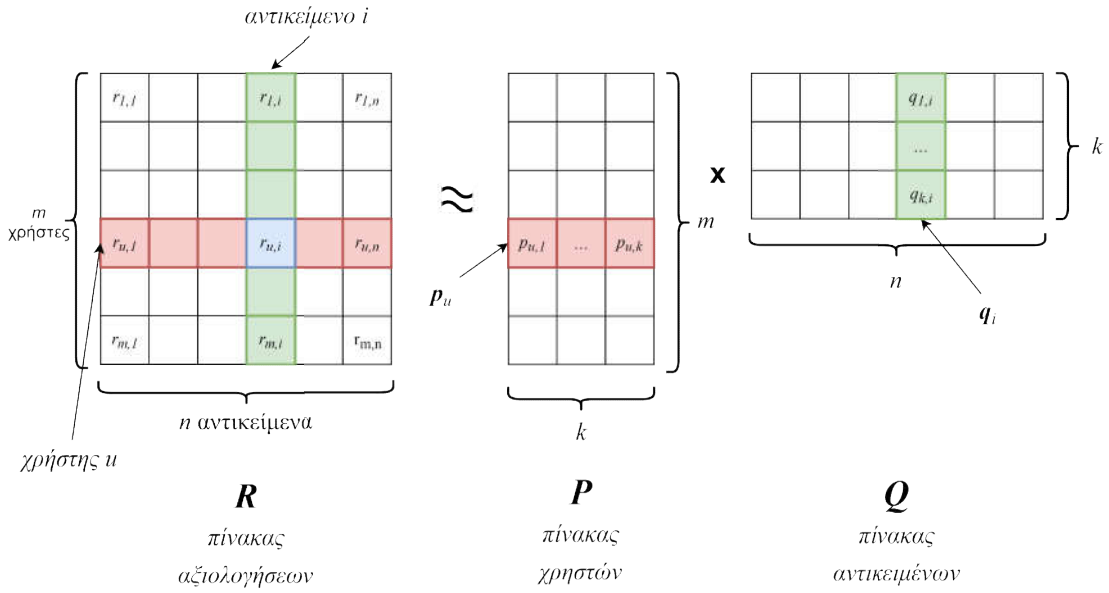
$$\boldsymbol{R} \approx \widehat{\boldsymbol{R}} = \boldsymbol{P}\boldsymbol{Q} \tag{3.7}$$

Όπου $\boldsymbol{P}$ και $\boldsymbol{Q}$ είναι πίνακες διαστάσεων $N \times K$ και $K \times M$ αντίστοιχα. Ο $\boldsymbol{P}$ είναι ο πίνακας των χαρακτηριστικών των χρηστών και ο $\boldsymbol{Q}$ είναι ο πίνακας χαρακτηριστικών των αντικειμένων. K είναι ο αριθμός των παραγόντων για την συγκεκριμένη παραγοντοποίηση και συνήθως ισχύει $K \ll N$ και $K \ll M$.

Αν συμβολίσουμε τα στοιχεία του $\boldsymbol{P}$ ως $p_{u,k}$, τα στοιχεία του $\boldsymbol{Q}$ ως $q_{k,i}$ και τα στοιχεία του $\widehat{\boldsymbol{R}}$ ως $\hat{r}_{u,i}$ τότε η εξίσωση (3.7) μπορεί να γραφτεί με την παρακάτω αναλυτική μορφή:

$$\hat{r}_{u,i} = \sum_{k=1}^{K} p_{u,k} q_{k,i} = \boldsymbol{p}_u \boldsymbol{q}_i \tag{3.8}$$

όπου $\boldsymbol{p}_u$ είναι μια γραμμή (διάνυσμα) του $\boldsymbol{P}$ και $\boldsymbol{q}_i$ είναι μια στήλη (διάνυσμα) του $\boldsymbol{Q}$. Στο Σχήμα 12 απεικονίζεται η γενική μορφή της Παραγοντοποίησης Πινάκων.



Σχήμα 12. Παραγοντοποίηση πίνακα R σε δύο μικρότερων διαστάσεων πίνακες P και Q

Για να υπολογίσουμε τους πίνακες $\boldsymbol{P}$ και $\boldsymbol{Q}$ μπορεί να χρησιμοποιηθεί κάποιος αλγόριθμος βελτιστοποίησης. Για παράδειγμα μπορούμε να αρχικοποιήσουμε τους πίνακες με τυχαίες τιμές και να υπολογίσουμε πόσο αποκλίνει το γινόμενο τους από τον αρχικό πίνακα $\boldsymbol{R}$. Στη

συνέχεια η απόκλιση (σφάλμα πρόβλεψης) μπορεί να ελαχιστοποιηθεί με την μέθοδο της Κατάβασης Δυναμικού (Παρ. 2.2.1., σελ. 17). Το σφάλμα πρόβλεψης δίνεται από τη σχέση:

$$e_{u,i} = r_{u,i} - \hat{r}_{u,i} = r_{u,i} - \boldsymbol{p}_u \boldsymbol{q}_i \tag{3.9}$$

Η συνάρτηση κόστους που πρέπει να ελαχιστοποιηθεί είναι το συνολικό τετραγωνικό σφάλμα (Sum of Squared Errors – SSE):

$$SSE = \sum_{(u,i)\in V} e_{u,i}^2 = \sum_{(u,i)\in V} \left(r_{u,i} - \boldsymbol{p}_u \boldsymbol{q}_i\right)^2 \tag{3.10}$$

Επίσης, σύμφωνα με τον ορισμό του RMSE (Εξ. (3.1), σελ. 43), ισχύει επίσης ότι:

$$RMSE = \sqrt{\frac{1}{|V|} SSE} \tag{3.11}$$

άρα η ελαχιστοποίηση του SSE, σημαίνει επίσης ότι θα ελαχιστοποιηθεί και το RMSE.

Στην πράξη, η απ' ευθείας ελαχιστοποίηση του SSE μπορεί να έχει ως αποτέλεσμα την υπερπροσαρμογή (overfitting) του μοντέλου στα δεδομένα εκπαίδευσης. Αυτό συμβαίνει επειδή το σύνολο εκπαίδευσης συνήθως είναι αραιό. Για να αποφύγουμε την υπερπροσαρμογή είναι απαραίτητο να προσθέσουμε στο μοντέλο ένα όρο regularization, οπότε η τελική συνάρτηση κόστους θα είναι η εξής:

$$L = \sum_{(u,i)\in V} \left(r_{u,i} - \boldsymbol{p}_u \boldsymbol{q}_i\right)^2 + \lambda(\|\boldsymbol{p}_u\|^2 + \|\boldsymbol{q}_i\|^2) \tag{3.12}$$

όπου $\|\boldsymbol{p}_u\|$ και $\|\boldsymbol{q}_i\|$ είναι η Ευκλείδεια νόρμα των διανυσμάτων $\boldsymbol{p}_u$ και $\boldsymbol{q}_i$ αντίστοιχα και λ είναι ένας συντελεστής regularization.

Η παραγοντοποίηση πινάκων τύπου SVD με SGD παρουσιάζεται σε αρκετές εργασίες με διαφορετικές ονομασίες και μικρές διαφορές στην υλοποίηση, όλες όμως βασίζονται στην ίδια ιδέα. Για παράδειγμα ο Paterek την αναφέρει ως Regularized SVD (Paterek, 2007) και στην (Takács et al., 2009) αναφέρεται ως Regularized Incremental Simultaneous MF (RISMF).

Για την βελτιστοποίηση της Εξίσωσης (3.12), οι πιο δημοφιλείς μέθοδοι είναι μέσω Στοχαστικής Κατάβασης Δυναμικού (Stochastic Gradient Descent – SGD) και Alternating Least Squares (ALS). Η προσέγγιση μέσω SGD προτάθηκε από τον Simon Funk κατά τη διάρκεια του διαγωνισμού Netflix Prize (Webb, 2006) και βασίστηκε σε μια προηγούμενη εργασία του (Gorrell and Webb, 2005). Η προσέγγιση μέσω Alternating Least Squares (ALS) προτάθηκε από τους (Bell and Koren, 2007).

3.3.3.1 Βελτιστοποίηση με SGD

Στην προσέγγιση βελτιστοποίησης μέσω SGD, αρχικά οι πίνακες $\boldsymbol{P}$ και $\boldsymbol{Q}$ αρχικοποιούνται με μικρές τυχαίες τιμές. Ο αλγόριθμος διατρέχει τα δεδομένα εκπαίδευσης και ενημερώνει τους πίνακες P και Q μετά από κάθε πρότυπο εκπαίδευσης. Σε κάθε εποχή εκπαίδευσης, γίνεται ένα πέρασμα όλων των αξιολογήσεων και για κάθε τιμή υπολογίζεται το σφάλμα πρόβλεψης με βάση την εξίσωση (3.9). Η ενημέρωση των στοιχείων των $\boldsymbol{P}$ και $\boldsymbol{Q}$, γίνεται σε μικρά βήματα με βάση ένα ρυθμό εκπαίδευσης και αντίθετα με την κατεύθυνση της κλίσης (gradient) της συνάρτησης κόστους. Η κλίση (gradient) δίνεται από τις παρακάτω μερικές παραγώγους της συνάρτησης κόστους (3.12):

$$\frac{\partial}{\partial \boldsymbol{p}_u} L = -2(e_{u,i}\boldsymbol{q}_i - \lambda \boldsymbol{p}_u) \tag{3.13}$$

$$\frac{\partial}{\partial \boldsymbol{q}_i} L = -2(e_{u,i}\boldsymbol{p}_u - \lambda \boldsymbol{q}_i) \tag{3.14}$$

Με βάση τις εξισώσεις (3.13) και (3.14), οι εξισώσεις ενημέρωσης των $\boldsymbol{p}_u$και $\boldsymbol{q}_i$ είναι οι εξής:

$$\boldsymbol{p}_u \leftarrow \boldsymbol{p}_u + \eta(e_{u,i}\boldsymbol{q}_i - \lambda \boldsymbol{p}_u) \tag{3.15}$$

$$\boldsymbol{q}_i \leftarrow \boldsymbol{q}_i + \eta(e_{u,i}\boldsymbol{p}_u - \lambda \boldsymbol{q}_i) \tag{3.16}$$

όπου η είναι ο ρυθμός εκπαίδευσης.

Η πλήρης περιγραφή της Παραγοντοποίησης Πινάκων με Στοχαστική Κατάβαση Δυναμικού φαίνεται στον Αλγόριθμο 3.1. Το κριτήριο τερματισμού της επαναληπτικής διαδικασίας είναι η μείωση του σφάλματος να γίνει μικρότερη από μια προκαθορισμένη τιμή κατωφλίου ϵ ή να πραγματοποιηθεί ένας συγκεκριμένος αριθμός επαναλήψεων ($epoch_{max}$).

Αλγόριθμος 3.1 Παραγοντοποίηση Πινάκων με Στοχαστική Κατάβαση Δυναμικού

Είσοδοι: Σύνολο εκπαίδευσης S, ρυθμός εκπαίδευσης η,
συντελεστής regularization λ, μέγιστος αριθμός εποχών εκπαίδευσης $epoch_{max}$
Έξοδοι: Πίνακες $\boldsymbol{P}$ και $\boldsymbol{Q}$ τέτοιοι ώστε να ισχύει $\boldsymbol{R} \approx \boldsymbol{PQ}$

1. Αρχικοποίησε τους $\boldsymbol{P}$ και $\boldsymbol{Q}$ με τυχαίες τιμές
2. $n \leftarrow 0$
3. Υπολόγισε το σφάλμα πρόβλεψης e_n για την εποχή n
4. **Επανάλαβε**
5. **Για** $(u, i, r_{u,i},) \in S$ **κάνε**
6. Υπολόγισε το σφάλμα $e_{u,i}$ (Εξ. (3.9))
7. Υπολόγισε την κλίση (gradient) (Εξ. (3.13) και (3.14))

8. Ενημέρωσε το $\boldsymbol{p}_u$ (Εξ. (3.15))
9. Ενημέρωσε το $\boldsymbol{q}_i$ (Εξ. (3.16))
10. **Τέλος Για**
11. $n \leftarrow n + 1$
12. Υπολόγισε το σφάλμα πρόβλεψης e_n για την εποχή n
13. **Μέχρις ότου** $|e_n - e_{n-1}| < \epsilon$ ή $n > epoch_{max}$

Καθώς υπάρχει η περίπτωση τα πλήθη χρηστών και αντικειμένων να είναι πολύ διαφορετικά μεταξύ τους, στη συνάρτηση κόστους (3.12) υπάρχει η δυνατότητα να χρησιμοποιηθούν ξεχωριστοί συντελεστές regularization λ_u και λ_i για τους χρήστες και τα αντικείμενα αντίστοιχα, δίνοντας έτσι καλύτερο έλεγχο της πολυπλοκότητας του μοντέλου. Στην περίπτωση αυτή η συνάρτηση κόστους είναι η παρακάτω:

$$L = \sum_{(u,i)\in V} \left(r_{u,i} - \boldsymbol{p}_u \boldsymbol{q}_i\right)^2 + \lambda_u \|\boldsymbol{p}_u\|^2 + \lambda_i \|\boldsymbol{q}_i\|^2) \qquad (3.17)$$

όπου λ_u είναι ο συντελεστής regularization για τους χρήστες και λ_i είναι ο συντελεστής regularization για τα αντικείμενα. Στην περίπτωση αυτή τροποποιούνται επίσης κατάλληλα και οι εξισώσεις (3.13) - (3.16), ώστε να περιλαμβάνουν τους δύο συντελεστές.

3.3.3.2 Βελτιστοποίηση με ALS

Μια άλλη προσέγγιση για την βελτιστοποίηση της συνάρτησης κόστους (3.12) είναι χρησιμοποιώντας την μέθοδο Alternating Least Squares (ALS). Η μέθοδος ALS εφαρμόζεται στην περίπτωση που σε μια εξίσωση υπάρχουν δύο άγνωστοι όπως για παράδειγμα στην περίπτωση της Παραγοντοποίησης Πινάκων, όπου θέλουμε να υπολογίσουμε τους πίνακες $\boldsymbol{P}$ και $\boldsymbol{Q}$. Η μέθοδος βασίζεται στην παρατήρηση ότι αν θεωρήσουμε τον ένα από τους δύο αγνώστους ως σταθερό όρο, η εξίσωση μετατρέπεται σε δευτεροβάθμια και η λύση της μπορεί να υπολογιστεί αναλυτικά (μέθοδος ελαχίστων τετραγώνων). Έτσι, ξεκινώντας με αρχικοποίηση του πίνακα $\boldsymbol{P}$ με τυχαίες τιμές και θεωρώντας τον σταθερό, μπορούμε να υπολογίσουμε τις τιμές του πίνακα $\boldsymbol{Q}$. Στη συνέχεια, θεωρούμε τον πίνακα $\boldsymbol{Q}$ σταθερό και υπολογίζουμε τις τιμές του πίνακα $\boldsymbol{P}$**.** Τα δύο βήματα υπολογισμού επαναλαμβάνονται μέχρι να ελαχιστοποιηθεί η συνάρτηση κόστους. Το όνομα της μεθόδου προέρχεται από την εναλλαγή και επανυπολογισμό μεταξύ των δύο πινάκων.

Οι εξισώσεις υπολογισμού των τιμών δίνονται με βάση την λύση του συστήματος γραμμικών εξισώσεων που προκύπτει κατά την ελαχιστοποίηση της συνάρτησης κόστους (3.12), κάθε φορά που διατηρούμε έναν από τους δύο πίνακες σταθερούς. Έτσι οι εξισώσεις υπολογισμού για κάθε διάνυσμα $\boldsymbol{p}_u$ και $\boldsymbol{q}_i$ είναι οι παρακάτω (Zhou et al., 2008):

$$\boldsymbol{p}_u = (\boldsymbol{Q}_u \cdot \boldsymbol{Q}_u^T + \lambda n_u \boldsymbol{I})^{-1} \cdot \boldsymbol{Q}_u \boldsymbol{r}_u \tag{3.18}$$

όπου $\boldsymbol{Q}_u$ είναι ένας υποπίνακας του πίνακα $\boldsymbol{Q}$ που περιέχει μόνο τα αντικείμενα που έχει αξιολογήσει ο χρήστης *u*, n_u είναι ο αριθμός των αντικειμένων που αξιολόγησε ο χρήστης *u*, $\boldsymbol{I}$ είναι ένας μοναδιαίος πίνακας διαστάσεων $k \times k$ και $\boldsymbol{r}_u$ είναι το διάνυσμα που περιέχει τις αξιολογήσεις του χρήστη *u*.

$$\boldsymbol{q}_i = (\boldsymbol{P}_i \cdot \boldsymbol{P}_i^T + \lambda n_i \boldsymbol{I})^{-1} \cdot \boldsymbol{P}_i \boldsymbol{r}_i \tag{3.19}$$

όπου $\boldsymbol{P}_i$ είναι ένας υποπίνακας του πίνακα $\boldsymbol{P}$ που περιέχει μόνο τους χρήστες που αξιολόγησαν το αντικείμενο *i*, n_i είναι ο αριθμός των χρηστών που αξιολόγησαν το αντικείμενο *i*, $\boldsymbol{I}$ είναι ένας μοναδιαίος πίνακας διαστάσεων $k \times k$ και $\boldsymbol{r}_i$ είναι το διάνυσμα που περιέχει τις αξιολογήσεις του αντικειμένου *i*.

Η πλήρης περιγραφή της Παραγοντοποίησης Πινάκων με Alternating Least Squares φαίνεται στον Αλγόριθμο 3.2. Το κριτήριο τερματισμού της επαναληπτικής διαδικασίας είναι η μείωση του σφάλματος να γίνει μικρότερη από μια προκαθορισμένη τιμή κατωφλίου ϵ ή να πραγματοποιηθεί ένας συγκεκριμένος αριθμός επαναλήψεων ($epoch_{max}$).

Αλγόριθμος 3.2 Παραγοντοποίηση Πινάκων με Alternating Least Squares

Είσοδοι: Σύνολο εκπαίδευσης S, συντελεστής regularization λ,
μέγιστος αριθμός εποχών εκπαίδευσης $epoch_{max}$
Έξοδοι: Πίνακες $\boldsymbol{P}$ και $\boldsymbol{Q}$ τέτοιοι ώστε να ισχύει $\boldsymbol{R} \approx \boldsymbol{PQ}$

1. Αρχικοποίησε τον $\boldsymbol{P}$ με τυχαίες τιμές
2. $n \leftarrow 0$
3. Υπολόγισε το σφάλμα πρόβλεψης e_n για την εποχή n
4. **Επανάλαβε**
5. Θεώρησε σταθερό τον $\boldsymbol{P}$ και υπολόγισε τον $\boldsymbol{Q}$ (Εξ. (3.19))
6. Θεώρησε σταθερό τον $\boldsymbol{Q}$ και υπολόγισε τον $\boldsymbol{P}$ (Εξ. (3.6))
7. $n \leftarrow n + 1$
8. Υπολόγισε το σφάλμα πρόβλεψης e_n για την εποχή n
9. **Μέχρις ότου** $|e_n - e_{n-1}| < \epsilon$ ή $n > epoch_{max}$

Η μέθοδος ALS είναι γενικά πιο αργή και λιγότερο ακριβής από την SGD. Όμως έχει το πλεονέκτημα της εύκολης παραλληλοποίησης της διαδικασίας καθώς ο υπολογισμός του κάθε διανύσματος $\boldsymbol{p}_u$ είναι ανεξάρτητος από τους υπόλοιπους παράγοντες των χρηστών και ο υπολογισμός του κάθε διανύσματος $\boldsymbol{q}_i$ είναι ανεξάρτητος από τους υπόλοιπους παράγοντες των αντικειμένων (Koren et al., 2009).

3.3.4 Biased SVD

Μια παραλλαγή της Παραγοντοποίησης Πινάκων που περιγράφηκε στην προηγούμενη παράγραφο, μπορεί να πραγματοποιηθεί με την ενσωμάτωση των βασικών εκτιμητών, που περιγράφηκαν στην παράγραφο 3.3.1, στην συνάρτηση κόστους (Koren et al., 2009). Αντίστοιχη προσέγγιση με άλλη ονομασία και μικρές διαφορές συναντάται και σε άλλες εργασίες όπως *"Improved Regularized SVD"* στην (Paterek, 2007) και Biased RISMF (BRISMF) στην (Takács et al., 2009).

Η προσέγγιση αυτή βασίζεται στο γεγονός ότι η διαμόρφωση των τιμών των γνωστών αξιολογήσεων δεν οφείλεται αποκλειστικά στην αλληλεπίδραση των χρηστών με τα αντικείμενα, αλλά υπάρχουν και άλλοι παράγοντες (biases) που την επηρεάζουν. Έτσι, μπορεί να δημιουργηθεί ένα μοντέλο που να περιλαμβάνει biases αλλά και μοντελοποίηση των προτιμήσεων των χρηστών μέσω λανθανόντων παραγόντων. Αυτό σημαίνει ότι πλέον η αξιολόγηση ενός χρήστη u για ένα αντικείμενο i αναλύεται με βάση την παρακάτω σχέση:

$$\hat{r}_{u,i} = \mu + b_u + b_i + \boldsymbol{p}_u \boldsymbol{q}_i \tag{3.20}$$

όπου μ είναι η μέση τιμή όλων των γνωστών αξιολογήσεων και οι παράμετροι b_u και b_i είναι αποκλίσεις των γνωστών αξιολογήσεων του χρήστη u και του αντικειμένου i από το μ, αντίστοιχα. Τέλος, το γινόμενο $\boldsymbol{p}_u \boldsymbol{q}_i$, μοντελοποιεί μέσω λανθανόντων παραγόντων, το υπόλοιπο της τιμής αξιολόγησης (residual) που δεν μοντελοποιήθηκε μέσω των biases.

Η συνάρτηση κόστους του μοντέλου Biased SVD, αποτελεί συνδυασμό των συναρτήσεων κόστους των δύο επιμέρους μοντέλων (3.5) και (3.12):

$$\min_{\boldsymbol{p}*,\boldsymbol{q}*,b*} \sum_{(u,i)\in\mathcal{R}} \left(r_{u,i} - (\mu + b_u + b_i + \boldsymbol{p}_u \boldsymbol{q}_i)\right)^2 + \lambda\left(\|\boldsymbol{p}_u\|^2 + \|\boldsymbol{q}_i\|^2 + b_u^2 + b_i^2\right) \tag{3.21}$$

Η νέα αυτή συνάρτηση κόστους μπορεί να ελαχιστοποιηθεί είτε με Στοχαστική Κατάβαση Δυναμικού χρησιμοποιώντας τον αλγόριθμο 3.1, είτε με Alternating Least Squares χρησιμοποιώντας τον αλγόριθμο 3.2.

Όπως στον αλγόριθμο Παραγοντοποίησης Πινάκων τύπου SVD, έτσι και εδώ, στη συνάρτηση κόστους μπορούν να χρησιμοποιηθούν ξεχωριστοί συντελεστές regularization για καλύτερο έλεγχο της πολυπλοκότητας του μοντέλου. Για παράδειγμα, μπορούν να χρησιμοποιηθούν ξεχωριστοί συντελεστές για το τμήμα της παραγοντοποίησης (λ_{ui}) και το τμήμα του βασικού εκτιμητή ($\lambda_{b_{ui}}$) και επιπλέον ο καθένας από αυτούς να αναλυθεί σε ξεχωριστούς συντελεστές για χρήστες και αντικείμενα του τμήματος παραγοντοποίησης (λ_u και λ_i) και ξεχωριστούς συντελεστές για χρήστες και αντικείμενα του βασικού εκτιμητή (λ_{b_u} και λ_{b_i}).

3.3.5 SVD++

3.3.5.1 Εξαγωγή implicit feedback από δεδομένα αξιολογήσεων

Ένας πίνακας με δεδομένα αξιολογήσεων, θεωρείται ότι περιέχει άμεση (explicit) πληροφορία σχετικά με τις προτιμήσεις των χρηστών για αντικείμενα. Όμως, εκτός από την τιμή αξιολόγησης, μπορούμε να συνάγουμε και ένα είδος έμμεσης (implicit) πληροφορίας: τα δεδομένα αξιολόγησης δεν μας πληροφορούν μόνο για το βαθμό προτίμησης, αλλά και για το *ποιά* αντικείμενα αξιολόγησε ο χρήστης, ασχέτως αν τα αξιολόγησε υψηλά ή χαμηλά.

Από αυτά τα έμμεσα δεδομένα, μπορούμε να κατασκευάσουμε ένα πίνακα δυαδικών τιμών, στον οποίο η τιμή 1 σημαίνει «αξιολόγησε» και η τιμή 0 «δεν αξιολόγησε». Ο πίνακας αυτός μπορεί να χρησιμοποιηθεί για την βελτίωση της ακρίβειας των προβλέψεων των μοντέλων Παραγοντοποίησης Πινάκων (Mnih and Salakhutdinov, 2007; Paterek, 2007).

3.3.5.2 Βελτίωση Παραγοντοποίησης Πινάκων με implicit feedback

Μια μέθοδος Παραγοντοποίησης Πινάκων που χρησιμοποιεί την έμμεση πληροφορία που περιέχεται στα δεδομένα αξιολογήσεων, είναι η SVD++ (Koren, 2008). Για την εκμετάλλευση της έμμεσης πληροφορίας, προστίθεται στο μοντέλο SVD ένα διάνυσμα που συνδέει τον κάθε χρήστη u με τα αντικείμενα που αξιολόγησε. Μια τιμή αξιολόγησης $\hat{r}_{u,i}$ δίνεται από τη σχέση:

$$\hat{r}_{u,i} = b_{u,i} + \boldsymbol{q}_i \left(\boldsymbol{p}_u + |R(u)|^{-\frac{1}{2}} \sum_{j \in R(u)} y_j \right) \tag{3.22}$$

όπου το σύνολο $R(u)$ περιέχει τα αντικείμενα που αξιολόγησε ο χρήστης u.

Στο διάνυσμα του χρήστη $\boldsymbol{p}_u$ προστίθεται το άθροισμα $|R(u)|^{-\frac{1}{2}} \sum_{j \in R(u)} y_j$, το οποίο αντιπροσωπεύει την έμμεση πληροφορία για τα αντικείμενα που αξιολόγησε ο χρήστης. Επειδή ο κάθε χρήστης αξιολογεί διαφορετικό αριθμό αντικειμένων, το άθροισμα κανονικοποιείται με τη βοήθεια του $|R(u)|^{-\frac{1}{2}}$. Το μοντέλο SVD++ μπορεί να εκπαιδευτεί χρησιμοποιώντας Στοχαστική Κατάβαση Δυναμικού.

3.4 Αξιολόγηση των Συστημάτων Συστάσεων

Η αξιολόγηση των συστημάτων συστάσεων αποτελεί ένα σημαντικό μέρος της διαδικασίας ανάπτυξης τους. Καταρχήν, κατά της φάση σχεδιασμού ενός συστήματος, η αξιολόγηση είναι αναγκαία για να τεκμηριωθούν οι επιλογές που πάρθηκαν για την αρχιτεκτονική και τους

επιμέρους αλγορίθμους του. Μετέπειτα, στη διάρκεια της λειτουργίας του, είναι απαραίτητο να διερευνηθεί κατά πόσο καλύπτει τις ανάγκες των χρηστών του. Επιπλέον, η αξιολόγηση μπορεί να είναι πολύπλευρη, δηλαδή να μην περιορίζεται μόνο στην ακρίβεια των προβλέψεων, αλλά να αξιολογούνται επίσης και άλλοι τομείς όπως ταχύτητα, ευχρηστία κλπ.

Ο σχεδιασμός μεθοδολογιών αξιολόγησης για συστήματα συστάσεων αποτελεί ένα σημαντικό πεδίο έρευνας στην σχετική βιβλιογραφία. Στην εργασία (Gunawardana and Shani, 2015) οι συγγραφείς αναφέρουν τρεις μεθοδολογίες αξιολόγησης: την offline αξιολόγηση, τις μελέτες χρηστών και την online αξιολόγηση.

Στην offline αξιολόγηση, που είναι και η ευκολότερη από τις τρεις, εκτελούνται πειράματα σε ήδη υπάρχουσες συλλογές δεδομένων και χρησιμοποιούνται διάφορες μετρικές απόδοσης για να αξιολογηθούν οι δυνατότητες του συστήματος, όπως π.χ. η ακρίβεια των προβλέψεων. Οι μελέτες χρηστών είναι μια πιο σύνθετη και δαπανηρή διαδικασία, όπου σε μια ομάδα χρηστών ανατίθεται να εκτελέσουν κάποια σενάρια χρήσης στο υπό αξιολόγηση σύστημα και στη συνέχεια να απαντήσουν σε ερωτηματολόγια σχετικά με την εμπειρία τους. Τέλος, σε ένα σύστημα εν λειτουργία, μπορεί να εφαρμοστεί online αξιολόγηση, δηλαδή να μετρηθεί η απόδοση του συστήματος με βάση τις ενέργειες των πραγματικών χρηστών του, συνήθως χωρίς να το γνωρίζουν οι ίδιοι. Τυπικά η online αξιολόγηση πραγματοποιείται μέσω της τεχνικής του A/B testing, δηλαδή με τη χρήση μιας διαφορετικής τεχνικής σύστασης σε μια τυχαία επιλεγμένη ομάδα χρηστών με ταυτόχρονη αξιολόγηση διαφόρων μετρικών απόδοσης.

Στην παρούσα εργασία πραγματοποιήθηκαν πειράματα offline αξιολόγησης σε γνωστές συλλογές δεδομένων που είναι δημόσια διαθέσιμες στο διαδίκτυο, ώστε να διερευνηθεί η ακρίβεια των προβλέψεων διαφόρων αλγορίθμων συνεργατικού φιλτραρίσματος.

3.5 Μετρικές αξιολόγησης

3.5.1 Πυκνότητα και Αραιότητα αξιολογήσεων

Η πυκνότητα και αραιότητα αξιολογήσεων (Rating density and sparsity) δεν αποτελούν μετρικές απόδοσης, όμως είναι χρήσιμες κατά την σύγκριση της απόδοσης ενός αλγορίθμου σε συλλογές δεδομένων με διαφορετική κατανομή αξιολογήσεων.

Η *πυκνότητα αξιολογήσεων (ratings density)* είναι ουσιαστικά το ποσοστό των γνωστών αξιολογήσεων ενός συστήματος σε σχέση με το μέγιστο δυνατό αριθμό αξιολογήσεων, με άλλα λόγια το ποσοστό πληρότητας σε αξιολογήσεις. Ορίζεται ως ο λόγος του πλήθους των αποθηκευμένων αξιολογήσεων στο σύστημα προς το μέγιστο δυνατό αριθμό αξιολογήσεων

που θα υπήρχαν αν όλοι οι χρήστες του συστήματος είχαν αξιολογήσει όλα τα αντικείμενα του. Δηλαδή η πυκνότητα αξιολογήσεων ορίζεται ως:

$$R_{density} = \frac{|R|}{|U| \times |I|} \ (\%)$$

όπου *R* είναι το σύνολο των γνωστών αξιολογήσεων, *U* το σύνολο των χρηστών του συστήματος και *I* το σύνολο των αντικειμένων του συστήματος.

Η *αραιότητα αξιολογήσεων (ratings sparsity)* αποτελεί συμπληρωματικό μέγεθος της πυκνότητας αξιολογήσεων και ορίζεται ως:

$$R_{sparsity} = 1 - R_{density} \ (\%)$$

3.5.2 Μετρικές ακρίβειας των προβλέψεων

Στην διαδικασία αξιολόγησης ενός συστήματος συστάσεων χρησιμοποιούνται μετρικές ακρίβειας των προβλέψεων (Prediction accuracy metrics). Στην βιβλιογραφία αναφέρονται ποικίλες μετρικές και επιλέγονται ανάλογα με τα ειδικά χαρακτηριστικά και τους στόχους του συστήματος. Έτσι για παράδειγμα για την πρόβλεψη αξιολογήσεων οι πιο δημοφιλείς μετρικές είναι το Μέσο Απόλυτο Σφάλμα και η Ρίζα του Μέσου Τετραγωνικού Σφάλματος. Οι μετρικές αυτές ουσιαστικά μετρούν την απόκλιση των προβλέψεων από τις πραγματικές τιμές και αξιολογούν την συνολική απόδοση του συστήματος, οπότε εφαρμόζονται συχνά σε offline πειραματικές διαδικασίες. Η ακρίβεια των προβλέψεων συνδέεται με την ποιότητα των παραγόμενων συστάσεων, διότι όσο πιο ακριβές είναι το σύστημα στις προβλέψεις του τόσο πιο επιτυχημένες θα είναι και οι συστάσεις του.

3.5.2.1 Μέσο Απόλυτο Σφάλμα (MAE)

Το Μέσο Απόλυτο Σφάλμα (Mean Absolute Error – MAE) χρησιμοποιείται για την μέτρηση της μέσης απόλυτης απόκλισης μεταξύ των πραγματικών και των εκτιμώμενων αξιολογήσεων των χρηστών (Herlocker et al., 2004). Για τον υπολογισμό της μετρικής ας υποτεθεί ότι ένας αλγόριθμος πρόβλεψης έχει παραγάγει *N* προβλέψεις για ζεύγη χρηστών-αντικειμένων $(u,i) \in S$ όπου $\hat{r}_{u,i}$ είναι η τιμή της πρόβλεψης και $r_{u,i}$ είναι η πραγματική τιμή της αξιολόγησης. Έτσι η διαφορά $\hat{r}_{u,i} - r_{u,i}$ είναι το σφάλμα της πρόβλεψης, οπότε το MAE υπολογίζεται ως εξής:

$$MAE = \frac{1}{N} \sum_{(u,i) \in S} \left| \hat{r}_{u,i} - r_{u,i} \right|$$

Μια παραλλαγή της MAE είναι το Κανονικοποιημένο Μέσο Απόλυτο Σφάλμα (Normalized MAE – NMAE) και αφορά την κανονικοποίηση του MAE ως προς το εύρος των δυνατών τιμών της αξιολογικής κλίμακας. Υπολογίζεται ως εξής:

$$NMAE = \frac{MAE}{r_{max} - r_{min}}$$

όπου r_{max} και r_{min} η μέγιστη και η ελάχιστη δυνατή τιμή της αξιολογικής κλίμακας αντίστοιχα. Η συγκεκριμένη μετρική προτάθηκε από τον (Goldberg et al., 2001) με σκοπό τη σύγκριση της απόδοσης ενός αλγορίθμου μεταξύ διαφορετικών συνόλων δεδομένων (που χρησιμοποιούν διαφορετική αξιολογική κλίμακα). Παρόλα αυτά, σύμφωνα με τον (Herlocker et al., 2004) η χρησιμότητα της δεν έχει επαληθευτεί και γενικά δεν χρησιμοποιείται συχνά στην βιβλιογραφία.

3.5.2.2 Ρίζα του Μέσου Τετραγωνικού Σφάλματος (RMSE)

Η Ρίζα του Μέσου Τετραγωνικού Σφάλματος (Root Mean Squared Error – RMSE) αποτελεί παραλλαγή του MAE, όπου το σφάλμα κάθε πρόβλεψης υψώνεται στο τετράγωνο, υπολογίζεται η μέση τιμή τους και τέλος υπολογίζεται η τετραγωνική ρίζα του αποτελέσματος. Σε συνοπτική μορφή δίνεται από τον τύπο:

$$RMSE = \sqrt{\frac{1}{N} \sum_{(u,i) \in S} \left(\hat{r}_{u,i} - r_{u,i}\right)^2}$$

Επειδή στην RMSE το σφάλμα υψώνεται στο τετράγωνο, η μετρική αυτή δίνει περισσότερο βάρος στα μεγαλύτερα σφάλματα σε σχέση με τα μικρότερα. Για παράδειγμα ένα σφάλμα με τιμή 1 αυξάνει το άθροισμα κατά 1 αλλά ένα σφάλμα με τιμή 2 αυξάνει το άθροισμα κατά 4. Αντίθετα η MAE δίνει το ίδιο βάρος σε όλα τα σφάλματα (Herlocker et al., 2004).

Η μετρική RMSE είναι ιδιαίτερα δημοφιλής μέθοδος μέτρησης της απόδοσης συστημάτων συστάσεων όσον αφορά την πρόβλεψη αξιολογήσεων. Σε αυτό συνέβαλλε και η χρήση της στο διαγωνισμό Netflix Prize, όπου αποτελούσε την επίσημη μέθοδο μέτρησης την απόδοσης.

4

Πειραματική αξιολόγηση αλγορίθμων Συνεργατικού Φιλτραρίσματος

4.1 Μεθοδολογία αξιολόγησης

Η μεθοδολογία αξιολόγησης που ακολουθήθηκε χωρίζεται σε 4 φάσεις:

- Προεπεξεργασία των δεδομένων
- Διαχωρισμός δεδομένων σε τμήματα εκπαίδευσης και ελέγχου
- Βελτιστοποίηση των υπερπαραμέτρων κάθε αλγορίθμου
- Εκπαίδευση τελικού μοντέλου και μέτρηση της ακρίβειας προβλέψεων

Στο Σχήμα 13 απεικονίζονται οι 4 φάσεις της διαδικασίας. Η κάθε φάση περιγράφεται συνοπτικά στη συνέχεια.

Σχήμα 13. Οι 4 φάσεις της μεθοδολογίας αξιολόγησης

4.1.1 Προεπεξεργασία των δεδομένων

Συνήθως τα δεδομένα που προέρχονται από τον πραγματικό κόσμο χρειάζονται κάποιου είδους προεπεξεργασία για να είναι κατάλληλα για χρήση σε μεθόδους μηχανικής μάθησης, καθώς μπορεί να περιέχουν λάθη, θόρυβο κλπ. Συνήθεις διαδικασίες προεπεξεργασίας είναι ο καθαρισμός των δεδομένων, το φιλτράρισμα τους, διάφοροι μετασχηματισμοί κ.α.

Στην παρούσα εργασία χρησιμοποιήθηκαν έτοιμες συλλογές δεδομένων, οι οποίες είναι ήδη σε κατάλληλη μορφή και δεν απαιτούν ενέργειες όπως αυτές που αναφέρθηκαν παραπάνω. Μόνη εξαίρεση αποτελεί η συλλογή δεδομένων Book Crossing (Παρ. 4.2.1.2), η οποία εκτός από δεδομένα αξιολογήσεων, περιέχει και έμμεση πληροφορία. Η έμμεση πληροφορία δεν χρησιμοποιείται στην διαδικασία οπότε και αφαιρέθηκε. Επιπλέον, η συγκεκριμένη συλλογή περιέχει δεδομένα από χρήστες με πολύ λίγες αξιολογήσεις (cold users), οι οποίοι αφαιρέθηκαν επίσης καθώς θα δυσκόλευαν την διαδικασία αξιολόγησης.

4.1.2 Διαχωρισμός δεδομένων σε τμήματα εκπαίδευσης και ελέγχου

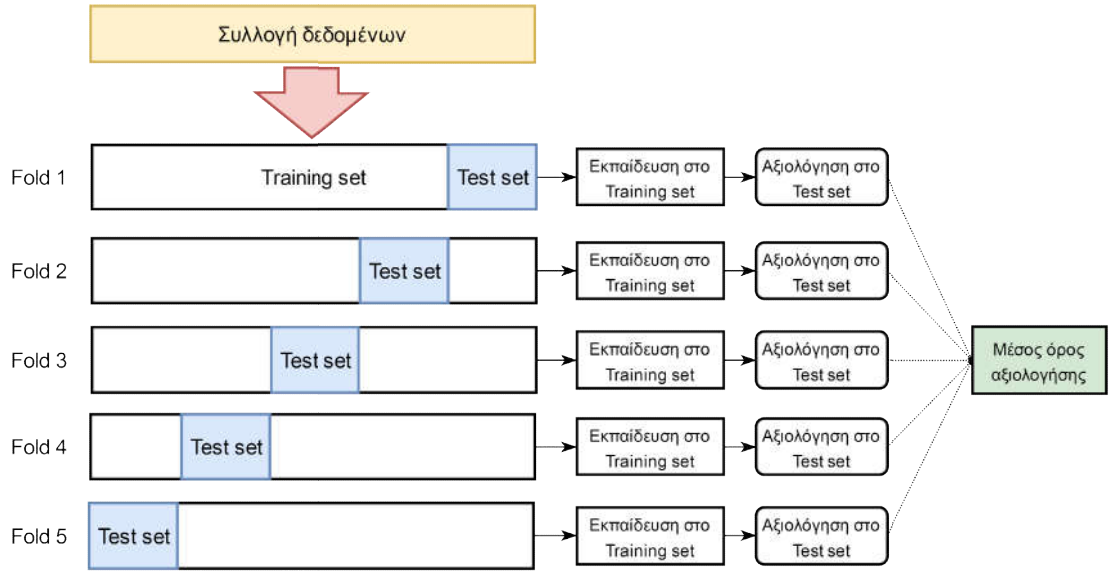
Ο τρόπος διαχωρισμού των δεδομένων σε τμήματα εκπαίδευσης (training set) και ελέγχου (testing set) είναι πολύ σημαντικός κατά την διεξαγωγή πειραμάτων. Διαφορετικοί τρόποι διαχωρισμού μπορεί να δώσουν διαφορετικά αποτελέσματα μετρήσεων απόδοσης, γι' αυτό πρέπει η στρατηγική διαχωρισμού να είναι σαφής και να διατηρηθεί σταθερή για όλα τα πειράματα.

Στην βιβλιογραφία που αναφέρεται σε Συστήματα Συστάσεων χρησιμοποιούνται διάφορες προσεγγίσεις ανάλογα με τον σκοπό της κάθε εργασίας. Αν τα δεδομένα αξιολογήσεων περιέχουν χρονικό προσδιορισμό όπως π.χ. τα σύνολα δεδομένων MovieLens, μπορεί να χρησιμοποιηθεί η προσέγγιση του διαχωρισμού με βάση το χρόνο, η οποία προσομοιάζει καλύτερα τις πραγματικές συνθήκες χρήσης ενός Συστήματος Συστάσεων.

Η γενική ιδέα του διαχωρισμού με βάση το χρόνο είναι να διαχωρίσουμε τις παλαιότερες αξιολογήσεις από τις νεότερες, ώστε να ελέγξουμε κατά πόσο ο αλγόριθμος μπορεί να προβλέψει τις μελλοντικές αξιολογήσεις βάσει των παλαιοτέρων. Ένας τρόπος να γίνει αυτό είναι να επιλεγεί ένα συγκεκριμένο σημείο στο χρόνο και στη συνέχεια όλα τα δεδομένα που είναι παλαιότερα να θεωρηθούν δεδομένα εκπαίδευσης, ενώ όλα τα μεταγενέστερα ως δεδομένα ελέγχου. Μια άλλη προσέγγιση είναι να γίνει ο διαχωρισμός κατά χρήστη, δηλαδή να ταξινομήσουμε τις αξιολογήσεις κάθε χρήστη με βάση το χρόνο και στη συνέχεια να ξεχωρίσουμε ένα ποσοστό από τις πιο πρόσφατες αξιολογήσεις του.

Αν αγνοήσουμε τα χρονικά δεδομένα ή δεν υπάρχει χρονική σήμανση, μπορούν να χρησιμοποιηθούν μέθοδοι τυχαίου διαχωρισμού των δεδομένων: α) επιλέγεται τυχαία ένας

συγκεκριμένος αριθμός αξιολογήσεων από κάθε χρήστη ως δεδομένα ελέγχου, β) επιλέγεται τυχαία ένα ποσοστό αξιολογήσεων ως δεδομένα ελέγχου και γ) επιλέγεται τυχαία ένα ποσοστό αξιολογήσεων χρησιμοποιώντας μέθοδο K-fold Cross Validation.



Σχήμα 14. Μέθοδος 5-Fold Cross Validation

Από τις τρεις μεθόδους που αναφέρθηκαν, η δημοφιλέστερη στη βιβλιογραφία φαίνεται να είναι η μέθοδος με K-fold Cross Validation (Kohavi and others, 1995) και είναι αυτή που ακολουθείται και στην παρούσα εργασία. Σύμφωνα με την μέθοδο (Σχήμα 14), η συλλογή δεδομένων χωρίζεται τυχαία σε K μη επικαλυπτόμενα τμήματα, όπου συνήθως K είναι 5 ή 10. Στη συνέχεια, ένα τμήμα επιλέγεται ως σύνολο ελέγχου (Test set) και τα υπόλοιπα K-1 τμήματα αποτελούν το σύνολο εκπαίδευσης. Αυτός ο διαχωρισμός ονομάζεται Fold 1. Το μοντέλο εκπαιδεύεται στο σύνολο εκπαίδευσης και η απόδοση του αξιολογείται στο σύνολο ελέγχου του Fold 1. Στην συνέχεια δημιουργείται το Fold 2, επιλέγοντας ένα άλλο τμήμα ως σύνολο ελέγχου και τα υπόλοιπα K-1 τμήματα ως σύνολο εκπαίδευσης. Το μοντέλο εκπαιδεύεται και αξιολογείται επίσης στο Fold 2. Η παραπάνω διαδικασία επαναλαμβάνεται έως τη δημιουργία του Fold K. Η τελική μέτρηση απόδοσης του μοντέλου είναι ο μέσος όρος των K μετρήσεων που προήλθαν από κάθε Fold. Στο Σχήμα 14 απεικονίζεται η μέθοδος για K = 5, οπότε η συγκεκριμένη μέθοδος ονομάζεται 5-Fold Cross Validation.

Επιπλέον, η δειγματοληψία για το σύνολο ελέγχου μπορεί να γίνει είτε ως ποσοστό των συνολικών δεδομένων (global ratio), είτε ως ποσοστό των αξιολογήσεων κάθε χρήστη (per-user ratio). Η δειγματοληψία ανά χρήστη εξασφαλίζει ότι για κάθε χρήστη του συνόλου ελέγχου, υπάρχουν τουλάχιστον κάποια δεδομένα αξιολόγησης και στο σύνολο εκπαίδευσης.

4.1.3 Βελτιστοποίηση υπερπαραμέτρων αλγορίθμων

Πολλοί αλγόριθμοι μηχανικής μάθησης, εκτός από τις παραμέτρους που πρέπει να υπολογιστούν μέσω της διαδικασίας εκπαίδευσης του μοντέλου, διαθέτουν και παραμέτρους οι οποίοι επηρεάζουν την διαδικασία εκπαίδευσης και γενικά τη συμπεριφορά του αλγορίθμου. Οι παράμετροι αυτού του είδους ονομάζονται *υπερπαράμετροι (hyperparameters)* του μοντέλου και πρέπει να οριστούν πριν από την έναρξη της εκπαίδευσης. Παραδείγματα υπερπαραμέτρων είναι ο αριθμός των λανθανόντων παραγόντων στα μοντέλα παραγοντοποίησης πινάκων, ο ρυθμός μάθησης στον αλγόριθμο κατάβασης δυναμικού, οι συντελεστές regularization σε διάφορους αλγορίθμους, ο αριθμός κρυφών επιπέδων και ο αριθμός νευρώνων κάθε επιπέδου στα μοντέλα νευρωνικών δικτύων κ.α. Στην βιβλιογραφία συναντώνται επίσης και με την ονομασία *μετα-παράμετροι (meta-parameters)* ή *ελεύθεροι παράμετροι (free parameters)*.

Οι υπερπαράμετροι πρέπει να οριστούν πριν την έναρξη της εκπαίδευσης και επηρεάζουν την απόδοση του μοντέλου. Για την εύρεση των βέλτιστων υπερπαραμέτρων, δηλαδή αυτών που μεγιστοποιούν την απόδοση του μοντέλου, υπάρχουν διάφορες μέθοδοι. Η απλούστερη ίσως μέθοδος είναι χρησιμοποιώντας *αναζήτηση πλέγματος (grid search)* (Snoek et al., 2012).

4.1.3.1 Αναζήτηση πλέγματος (grid search)

Η αναζήτηση πλέγματος εκτελεί ουσιαστικά εξαντλητική αναζήτηση σε ένα προεπιλεγμένο υποσύνολο συνδυασμών (πλέγμα) από τιμές υπερπαραμέτρων. Οι τιμές για κάθε υπερπαράμετρο προέρχονται από ένα επίσης συγκεκριμένο σύνολο τιμών. Για κάθε συνδυασμό του πλέγματος, το μοντέλο εκπαιδεύεται και αξιολογείται με μέθοδο K-Fold Cross Validation και επιλέγεται ο συνδυασμός τιμών υπερπαραμέτρων με την καλύτερη απόδοση.

Στα προβλήματα μηχανικής μάθησης, η βέλτιστη απόδοση ενός μοντέλου αναφέρεται στην ικανότητα γενίκευσης του. Υπό αυτό το πρίσμα, η επιλογή των βέλτιστων τιμών υπερπαραμέτρων πρέπει να γίνει με βάση την απόδοση των αντίστοιχων μοντέλων σε νέα «άγνωστα» δεδομένα. Για το λόγο αυτό, στην αναζήτηση πλέγματος η αξιολόγηση της απόδοσης κάθε μοντέλου δεν γίνεται στο σύνολο ελέγχου, αλλά χρησιμοποιείται ένα ξεχωριστό σύνολο επικύρωσης (validation set), το οποίο συνήθως προέρχεται από το αρχικό σύνολο εκπαίδευσης.

Στην παρούσα εργασία επιλέχθηκε η συγκεκριμένη μέθοδος για την βελτιστοποίηση των υπερπαραμέτρων, λόγω της απλότητας υλοποίησης της. Παρόλα αυτά η μέθοδος δεν εγγυάται ότι θα βρει το ολικό μέγιστο του χώρου των υπερπαραμέτρων. Επίσης, δεν ενδείκνυται για τη ταυτόχρονη βελτιστοποίηση μεγάλου πλήθους υπερπαραμέτρων, καθώς ο

συνολικός αριθμός των ερευνώμενων συνδυασμών των τιμών αυξάνεται με ρυθμό γεωμετρικής προόδου σε σχέση με το πλήθος των υπερπαραμέτρων. Αυτό καθιστά την διαδικασία υπολογιστικά δαπανηρή, έχει όμως το πλεονέκτημα της πολύ εύκολης παραλληλοποίησης, καθώς η αξιολόγηση κάθε συνδυασμού τιμών είναι ανεξάρτητη από τις υπόλοιπες.

4.1.3.2 Άλλες μέθοδοι βελτιστοποίησης υπερπαραμέτρων

Εκτός από την αναζήτηση πλέγματος, μια άλλη μέθοδος εύρεσης των βέλτιστων τιμών υπερπαραμέτρων είναι με *τυχαία αναζήτηση (random search)*, κατά την οποία επιλέγονται τυχαίες τιμές για κάθε υπερπαράμετρο από ένα συγκεκριμένο πεδίο τιμών (Bergstra and Bengio, 2012). Επίσης, υπάρχουν μέθοδοι που αντιμετωπίζουν την εύρεση των βέλτιστων υπερπαραμέτρων, σαν πρόβλημα βελτιστοποίησης χρησιμοποιώντας *Μπεϋζιανή βελτιστοποίηση (Bayesian optimization)* (Bergstra et al., 2011) ή μέθοδο κατάβασης δυναμικού (Maclaurin et al., 2015). Οι συγκριμένες μέθοδοι έχει αποδειχθεί ότι βρίσκουν τις βέλτιστες τιμές υπερπαραμέτρων σε συντομότερο χρονικό διάστημα συγκριτικά την μέθοδο αναζήτησης πλέγματος.

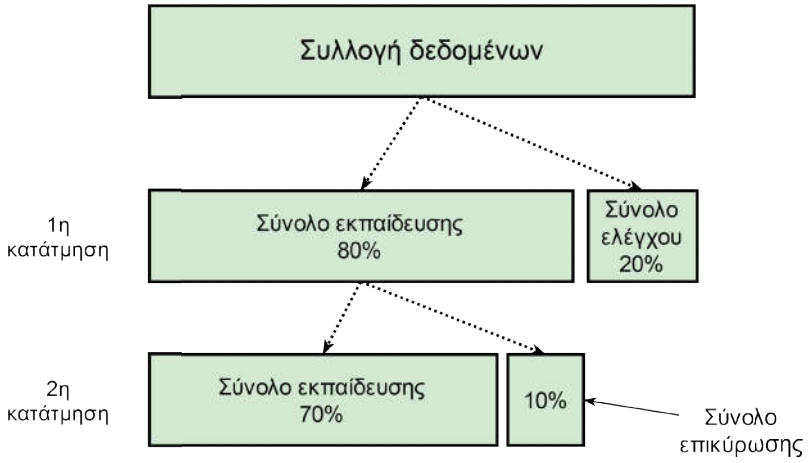
4.1.4 Εκπαίδευση τελικού μοντέλου και μέτρηση της ακρίβειας προβλέψεων

Μετά την φάση βελτιστοποίησης των υπερπαραμέτρων κάθε μοντέλου, εκπαιδεύεται ένα τελικό μοντέλο χρησιμοποιώντας τις βέλτιστες τιμές υπερπαραμέτρων που επιλέχθηκαν.

Τα μοντέλα που βασίζονται σε Παραγοντοποίηση Πινάκων με Στοχαστική Κατάβαση Δυναμικού, παρουσιάζουν έντονα το φαινόμενο της υπερπροσαρμογής μετά από κάποιο αριθμό εποχών εκπαίδευσης. Για την αποφυγή του προβλήματος χρησιμοποιήθηκε η τεχνική της *πρώιμης διακοπής εκπαίδευσης με επικύρωση (early stopping with validation)*. Σύμφωνα με την τεχνική αυτή, η ικανότητα γενίκευσης του εκπαιδευόμενου μοντέλου ελέγχεται σε ένα ξεχωριστό σύνολο επικύρωσης (validation set), με σκοπό την διακοπή της διαδικασίας εκπαίδευσης μόλις το σφάλμα επικύρωσης αρχίσει να αυξάνεται και οι τιμές των παραμέτρων του μοντέλου, επανέρχονται στις τιμές της προηγούμενης εποχής εκπαίδευσης.

Όπως αναφέρθηκε προηγουμένως, η τεχνική early stopping απαιτεί την ύπαρξη ενός ακόμα συνόλου δεδομένων, άρα ο διαχωρισμός των δεδομένων της συλλογής γίνεται σε 3 τμήματα: σύνολο εκπαίδευσης, σύνολο επικύρωσης και σύνολο ελέγχου. Συγκεκριμένα, η κατάτμηση μπορεί να γίνει σε δύο βήματα. Πρώτα να χωριστεί σε σύνολο εκπαίδευσης και σύνολο ελέγχου και στη συνέχεια το σύνολο εκπαίδευσης να χωριστεί σε ένα μικρότερο σύνολο εκπαίδευσης και σε ένα σύνολο επικύρωσης. Στην παρούσα εργασία η αναλογία κατάτμησης ήταν 80 % : 20 % για το πρώτο βήμα (μιας και επιλέχθηκε 5-Fold Cross Validation) και

87,5 % : 12,5 % με βάση το αρχικό σύνολο εκπαίδευσης για το δεύτερο βήμα. Αυτό δίνει την τελική κατάτμηση με αναλογίες 70 % : 10 % : 20 %. Ο τρόπος κατάτμησης παρουσιάζεται στο Σχήμα 15.



Σχήμα 15. Τρόπος κατάτμησης της συλλογής δεδομένων σε σύνολα εκπαίδευσης, επικύρωσης και ελέγχου

Για τον έλεγχο της γενίκευσης του μοντέλου, ορίστηκε μια τιμή κατωφλίου ε η οποία αφορά την μέγιστη επιτρεπτή διαφορά του σφάλματος επικύρωσης $e_{τρέχον}$ σε σχέση με το ελάχιστο σφάλμα e_{min} που έχει παρατηρηθεί μέχρι την τρέχουσα εποχή. Συνεπώς η εκπαίδευση διακόπτεται όταν:

$$e_{τρέχον} - e_{min} > \varepsilon$$

Τυπικές τιμές για το ε είναι 0.01 και 0.001 ανάλογα με το μοντέλο και τη συλλογή δεδομένων.

Τα βήματα για την εφαρμογή της τεχνικής early stopping είναι τα παρακάτω (Prechelt, 1998):

Αλγόριθμος 4.1 Early Stopping with Validation

Είσοδοι: Σύνολο δεδομένων, αλγόριθμος μάθησης
Έξοδοι: Εκπαιδευμένο μοντέλο

1. Κατάτμησε τα δεδομένα σε σύνολο εκπαίδευσης T και σύνολο επικύρωσης V
2. $n \leftarrow 0$, $e_{min} \leftarrow \infty$
3. **Επανάλαβε**
4. Εκπαίδευσε το μοντέλο στο σύνολο T για μία εποχή
5. Υπολόγισε το σφάλμα πρόβλεψης e_n στο σύνολο V για την εποχή n
6. **Αν** $e_n < e_{min}$
7. Αποθήκευσε τις παραμέτρους του μοντέλου
8. $e_{min} \leftarrow e_n$
9. **Τέλος Αν**
10. **Μέχρις ότου** $e_n - e_{min} > \varepsilon$
11. Χρησιμοποίησε το τελευταίο μοντέλο που αποθηκεύτηκε

Για την σύγκριση της απόδοσης των αλγορίθμων χρησιμοποιήθηκε η μέθοδος της διασταυρωμένης επικύρωσης 5 τμημάτων (5-fold Cross Validation) σε συνδυασμό με την τεχνική early stopping with validation, σε διάφορα γνωστά σύνολα δεδομένων που περιγράφονται στην συνέχεια (παράγραφος 4.2.1). Η εκτέλεση των πειραμάτων πραγματοποιήθηκε σε επεξεργαστή Intel i7 920 (4 πυρήνες – 2.8 GHz) με 12 GB μνήμης.

4.2 Οργάνωση πειραμάτων

4.2.1 Συλλογές δεδομένων

Για την πειραματική αξιολόγηση είναι διαθέσιμες αρκετές έτοιμες *συλλογές δεδομένων (datasets)* σχεδιασμένες ειδικά για πειράματα σχετικά με συστήματα συστάσεων. Τα δεδομένα είναι δομημένα συνήθως με την μορφή τριάδων (χρήστης, αντικείμενο, αξιολόγηση) και ανάλογα με τη συλλογή μπορεί να συμπεριλαμβάνονται και πρόσθετες πληροφορίες, όπως για παράδειγμα μια χρονοσφραγίδα (timestamp) ανά τριάδα, έτσι ώστε να μπορεί να προσδιοριστεί χρονολογικά κάθε αξιολόγηση.

Εκτός από τη βασική πληροφορία, τις βαθμολογίες αξιολογήσεων, στις συλλογές δεδομένων μπορεί να περιλαμβάνονται και *πρόσθετες πληροφορίες (side information)*. Για παράδειγμα για τους χρήστες μπορεί να υπάρχουν δημογραφικές πληροφορίες, όπως η ηλικία, το φύλο, η επαγγελματική απασχόληση κ.α. Αντίστοιχα, για τα αντικείμενα μπορεί να περιλαμβάνονται πληροφορίες ανάλογα με το είδος τους. Έτσι, αν η συλλογή δεδομένων αφορά αξιολογήσεις ταινιών, μπορεί να περιλαμβάνονται πρόσθετες πληροφορίες όπως ο τίτλος, η ημερομηνία παραγωγής και το είδος ταινίας (movie genre).

Στη συνέχεια περιγράφονται οι συλλογές δεδομένων που χρησιμοποιήθηκαν στα πειράματα, αναφέροντας τα ιδιαίτερα χαρακτηριστικά της καθεμιάς. Ο Πίνακας 3 παρουσιάζει συνοπτικά κάποια βασικά χαρακτηριστικά τους.

4.2.1.1 MovieLens

Από τις πλέον γνωστές συλλογές δεδομένων για συστήματα συστάσεων είναι αυτές που προέρχονται από το διαδικτυακό τόπο Movielens (“Movielens,” n.d.), μια διαδικτυακή υπηρεσία παροχής συστάσεων για ταινίες. Η υπηρεσία δημιουργήθηκε από μια ερευνητική ομάδα του πανεπιστημίου της Μινεσότα και με βάση τα δεδομένα που έχει συλλέξει, έχει δημοσιεύσει αρκετές συλλογές δεδομένων διαφορετικού μεγέθους όπως οι MovieLens-100k και η MovieLens-1M. Τα δεδομένα των συλλογών, είναι με τη μορφή τετράδων (χρήστης, ταινία, αξιολόγηση, χρονοσφραγίδα) όπου οι αξιολογήσεις είναι διακριτές τιμές από μία κλίμακα από 1 έως 5.

Η MovieLens-100k περιέχει 100.000 αξιολογήσεις από 943 χρήστες για 1682 ταινίες. Η συλλογή αυτή είναι η παλαιότερη χρονολογικά καθώς περιέχει δεδομένα αξιολογήσεων ταινιών από τα έτη 1997 και 1998. Η MovieLens-1M αποτελείται από 1.000.209 αξιολογήσεις πάνω σε περίπου 3.900 ταινίες που έγιναν από 6.040 χρήστες. Τα δεδομένα συλλέχθηκαν την περίοδο 2000-2003. Και στις δύο συλλογές περιλαμβάνονται χρήστες που έχουν αξιολογήσει τουλάχιστον 20 ταινίες. Η συλλογή 1M είναι αραιότερη από την 100k, όπως φαίνεται από τα χαρακτηριστικά που περιλαμβάνει ο Πίνακας 3. Αναλυτικότερη περιγραφή των συλλογών του MovieLens υπάρχει στο (Harper and Konstan, 2016).

4.2.1.2 Book-Crossing

Η συλλογή δεδομένων Book-Crossing περιέχει αξιολογήσεις για βιβλία και κατασκευάστηκε με δεδομένα που συλλέχθηκαν από τον ομώνυμο διαδικτυακό τόπο, που είναι υπηρεσία κοινωνικής δικτύωσης για βιβλιόφιλους. Χρησιμοποιήθηκε πρώτη φορά για έρευνα στο (Ziegler et al., 2005). Περιλαμβάνει περίπου 1,1 εκατομμύρια αξιολογήσεις που έγιναν σε περίπου 270.000 βιβλία από 280.000 περίπου χρήστες της υπηρεσίας. Η αξιολογική κλίμακα που χρησιμοποιείται περιλαμβάνει διακριτές τιμές από 1 έως 10, όμως υπάρχουν και αρκετές αξιολογήσεις με τιμή 0, οι οποίες αναφέρονται σε έμμεσες (implicit) αξιολογήσεις.

Στην παρούσα εργασία μας ενδιαφέρουν μόνο η άμεσες αξιολογήσεις, οπότε οι αξιολογήσεις με τιμή 0 αφαιρέθηκαν, καθώς και όσοι χρήστες είχαν λιγότερες από 6 αξιολογήσεις. Έτσι, το τελικό σύνολο δεδομένων που χρησιμοποιήθηκε περιέχει 329.336 άμεσες αξιολογήσεις από 12.019 χρήστες για 156.069 βιβλία. Πρόκειται για μια ιδιαίτερα αραιή συλλογή δεδομένων με πυκνότητα αξιολογήσεων περίπου 0,017 %

4.2.1.3 Jester

Οι συλλογές δεδομένων Jester (Goldberg et al., 2001) περιέχουν αξιολογήσεις για ανέκδοτα σε μια συνεχή κλίμακα με τιμές από -10 έως 10. Υπάρχουν διάφορες εκδόσεις των συνόλων, όμως το μεγαλύτερο που είναι γνωστό ως «Jester Dataset 1» περιέχει περίπου 4,1 εκατομμύρια αξιολογήσεις από 73.421 χρήστες που συλλέχθηκαν στο διάστημα 1999 - 2003. Η ιδιαιτερότητα της είναι ότι οι αξιολογήσεις αναφέρονται μόνο σε 100 ανέκδοτα καθιστώντας τη συγκεκριμένη συλλογή πολύ πιο πυκνή από τις υπόλοιπες. Στην παρούσα εργασία χρησιμοποιήθηκε ένα υποσύνολο της με 1,8 εκατομμύρια αξιολογήσεις από 24.983 χρήστες οι οποίοι έχουν αξιολογήσει τουλάχιστον 36 ανέκδοτα. Το συγκεκριμένο υποσύνολο έχει πυκνότητα αξιολογήσεων 72,4 %.

Πίνακας 3. Χαρακτηριστικά των συλλογών δεδομένων που χρησιμοποιήθηκαν στα πειράματα

Σύνολο δεδομένων	Είδος	Αξιολογική κλίμακα	Χρήστες	Αντικεί-μενα	Αριθμός αξιολογήσεων	Πυκνό-τητα
MovieLens 100k	Ταινίες	1-5 (διακριτή)	943	1.682	100.000	6,30 %
MovieLens 1M	Ταινίες	1-5 (διακριτή)	6.040	3.706	1.000.209	4,47 %
Book-Crossing	Βιβλία	1-10 (διακριτή)	12.019	156.069	329.336	0,017 %
Jester (dataset 1)	Ανέκδοτα	-10 έως 10 (συνεχής)	24.983	100	1.8M	72,4 %

4.2.2 Πειραματική διαδικασία

Η πειραματική διαδικασία βασίστηκε στην μεθοδολογία αξιολόγησης που περιγράφηκε στην παράγραφο 4.1. Στη συνέχεια δίνονται λεπτομέρειες για την εκτέλεση της.

4.2.2.1 Βασικοί Εκτιμητές

Για κάθε συλλογή δεδομένων πρώτα υπολογίστηκαν οι τρεις απλοί βασικοί εκτιμητές Global Average, User Average και Item Average. Στη συνέχεια για το μοντέλο User-Item Baseline έγινε grid search για να βρεθούν οι τιμές για τους συντελεστές Regularization. Στους πίνακες και στα διαγράμματα αποτελεσμάτων, οι τεχνικές αυτές συμβολίζονται ως Global Avg, User Avg, Item Avg και U-I Baseline αντίστοιχα.

4.2.2.2 Τεχνικές βασισμένες στη μνήμη

Σε αυτή την κατηγορία τεχνικών εκπαιδεύτηκαν μοντέλα χρησιμοποιώντας τον αλγόριθμο KNN, με βάση τις τεχνικές User-based και Item-based. Για κάθε τεχνική χρησιμοποιήθηκαν οι δύο δημοφιλέστερες μετρικές ομοιότητας: η ομοιότητα Συνημιτόνου και η ομοιότητα με βάση τον συντελεστή συσχέτισης του Pearson. Επιπλέον, αξιολογήθηκε και μια παραλλαγή των παραπάνω, που ενσωματώνει και τον βασικό εκτιμητή User-Item Baseline. Οι βέλτιστες τιμές Regularization που χρησιμοποιήθηκαν για τον βασικό εκτιμητή είναι αυτές που βρέθηκαν στο προηγούμενο βήμα.

Συνολικά αξιολογήθηκαν 8 τεχνικές βασισμένες στη μνήμη. Στους πίνακες και στα διαγράμματα αποτελεσμάτων, οι τεχνικές User-based συμβολίζονται ως UserKNN-Cosine και UserKNN-Pearson και οι τεχνικές Item-based ως ItemKNN-Cosine και ItemKNN-Pearson, ανάλογα με την μετρική ομοιότητας που χρησιμοποιείται. Όταν στις παραπάνω τεχνικές προστίθεται και ο βασικός εκτιμητής User-Item Baseline μπροστά στο όνομα μπαίνει το διακριτικό Biased, που σημαίνει ότι η τεχνική χρησιμοποιεί biases.

4.2.2.3 Τεχνικές Παραγοντοποίησης Πινάκων

Οι τεχνικές Παραγοντοποίησης Πινάκων διαθέτουν αρκετές υπερπαραμέτρους. Όλοι οι αλγόριθμοι διαθέτουν τον αριθμό λανθανόντων παραγόντων k και ανάλογα με τον αλγόριθμο που χρησιμοποιείται υπάρχουν διάφοροι άλλοι υπερπαράμετροι, όπως ρυθμοί εκμάθησης και συντελεστές regularization.

Για την εύρεση των βέλτιστων τιμών αυτών των υπερπαραμέτρων χρησιμοποιήθηκε μέθοδος grid search σε δύο φάσεις. Στην πρώτη φάση επιλέχθηκε ένας σταθερός και μικρός αριθμός λανθανόντων παραγόντων (k = 10) και εκτελέστηκε grid search για τις υπόλοιπες υπερπαραμέτρους του κάθε αλγορίθμου. Από τα αποτελέσματα του grid search επιλέχθηκαν οι τιμές υπερπαραμέτρων για τις οποίες τα μοντέλα παρουσίασαν το ελάχιστο σφάλμα RMSE. Στη δεύτερη φάση του grid search, οι τιμές που επιλέχθηκαν στην πρώτη φάση κρατήθηκαν σταθερές και πραγματοποιήθηκε grid search για διάφορες τιμές του πλήθους λανθανόντων παραγόντων k.

Στις τεχνικές Παραγοντοποίησης Πινάκων παρουσιάζεται έντονα το φαινόμενο υπερπροσαρμογής μετά από αρκετές εποχές εκπαίδευσης του αλγορίθμου, οπότε σε όλα τα πειράματα χρησιμοποιήθηκε η μέθοδος early stopping with validation, όπως περιγράφτηκε στην παράγραφο 4.1.4.

4.2.3 Τιμές υπερπαραμέτρων που χρησιμοποιήθηκαν

Στη συνέχεια παρατίθενται οι τιμές υπερπαραμέτρων που χρησιμοποιήθηκαν για την τελική σύγκριση των αλγορίθμων για κάθε συλλογή δεδομένων ξεχωριστά. Οι τιμές αυτές βρέθηκαν με μέθοδο grid search.

4.2.3.1 Μοντέλο User-Item Baseline

Το μοντέλο User-Item Baseline εκπαιδεύτηκε με αλγόριθμο βελτιστοποίησης Alternating Least Squares (ALS) και διαθέτει δύο συντελεστές regularization λ_u και λ_i. Οι συντελεστές αυτοί ελέγχουν το regularization για την πόλωση (bias) των χρηστών και των αντικειμένων αντίστοιχα.

Για την επιλογή τους έγινε grid search και για τους δύο, για τις τιμές [0.01, 0.1, 1, 5, 10, 15, 20, 50, 100, 1000]. Οι ίδιες τιμές χρησιμοποιήθηκαν για όλες τις συλλογές. Ο Πίνακας 4 περιέχει τις βέλτιστες τιμές λ_u και λ_i που επιλέχθηκαν σε κάθε συλλογή.

Πίνακας 4. Οι βέλτιστες τιμές των συντελεστών regularization για το μοντέλο User-Item Baseline. Οι τιμές επιλέχθηκαν με grid search για κάθε συλλογή δεδομένων.

Συλλογή δεδομένων	λ_u	λ_i
MovieLens 100k	5	1
MovieLens 1M	5	1
Jester	1	1
Book Crossing	1	10

4.2.3.2 Τεχνικές βασισμένες στη μνήμη

Τα μοντέλα αυτής της κατηγορίας αξιολογήθηκαν με grid search για 10 τιμές του k, δηλαδή του πλήθους των πλησιέστερων γειτόνων που λαμβάνονται υπόψη. Συγκεκριμένα για τις τιμές από 10 έως 100 με βήμα 10. Στην περίπτωση της συλλογής book crossing αξιολογήθηκαν επίσης και οι τιμές k = [1, 3, 5].

4.2.3.3 Μοντέλο ALS (Alternating Least Squares)

Το μοντέλο ALS βασίζεται σε παραγοντοποίηση πινάκων τύπου SVD με regularization και εκπαιδεύεται με τη μέθοδο Alternating Least Squares (ALS). Διαθέτει 2 υπερπαραμέτρους, τον αριθμό των παραγόντων k και τον συντελεστή regularization λ.

Για κάθε συλλογή πραγματοποιήθηκε grid search με Cross Validation για την εύρεση κατάλληλων τιμών του λ. Από δοκιμές που έγιναν αποδείχθηκε ότι ο συγκεκριμένος αλγόριθμος είναι ευαίσθητος σε πολύ μικρές διακυμάνσεις του συντελεστή λ. Το εύρος τιμών που διερευνήθηκε δίνεται από την σχέση $\lambda = 10^{-\alpha}$ όπου $\alpha = [1, 2, \dots, 10]$, κρατώντας σταθερό το $k = 10$. Στη συνέχεια έγινε μια δεύτερη πιο λεπτομερής διερεύνηση τιμών γύρω από την βέλτιστη τιμή του προηγούμενου βήματος. Τα μοντέλα εκπαιδεύτηκαν για 15 εποχές. Ο Πίνακας 5 περιέχει τις βέλτιστες τιμές που επιλέχθηκαν για κάθε συλλογή μέσω της μεθόδου grid search.

Πίνακας 5. Οι βέλτιστες τιμές των υπερπαραμέτρων για τον αλγόριθμο ALS ανά συλλογή δεδομένων

Συλλογή δεδομένων	λ
MovieLens 100k	0.0001
MovieLens 1M	10^{-5}
Jester	0.0002
Book Crossing	$5.8 \cdot 10^{-5}$

4.2.3.4 Μοντέλο MF (Matrix Factorization)

Το μοντέλο MF βασίζεται σε παραγοντοποίηση πινάκων τύπου SVD με regularization και εκπαιδεύεται με στοχαστική κατάβαση δυναμικού. Διαθέτει 3 υπερπαραμέτρους, τον αριθμό των παραγόντων k, τον ρυθμό εκμάθησης η και τον συντελεστή regularization λ.

Για κάθε συλλογή πραγματοποιήθηκε grid search για την εύρεση κατάλληλων τιμών του η και λ. Οι τιμές που διερευνήθηκαν ήταν για τον ρυθμό εκπαίδευσης $\eta = [0.0005, 0.001, 0.005, 0.01, 0.02, 0.03]$ και για τον συντελεστή regularization $\lambda = [0.01, 0.015, 0.05, 0.1, 0.15, 0.5]$, κρατώντας σταθερό το $k = 10$. Ο μέγιστος αριθμός εποχών εκπαίδευσης ήταν 100 εποχές και χρησιμοποιήθηκε τεχνική early stopping with validation. Ο Πίνακας 6 περιέχει τις βέλτιστες τιμές που επιλέχθηκαν για κάθε συλλογή μέσω της μεθόδου grid search.

Πίνακας 6. Οι βέλτιστες τιμές των υπερπαραμέτρων για τον αλγόριθμο MF ανά συλλογή δεδομένων

Συλλογή δεδομένων	η	λ
MovieLens 100k	0.01	0.1
MovieLens 1M	0.01	0.05
Jester	0.0005	0.1
Book Crossing	0.03	0.1

4.2.3.5 Μοντέλο BMF (Biased Matrix Factorization)

Το μοντέλο MF βασίζεται σε παραγοντοποίηση πινάκων τύπου SVD με regularization, ενσωματώνει User-Item biases και εκπαιδεύεται με στοχαστική κατάβαση δυναμικού. Διαθέτει 5 υπερπαραμέτρους, τον αριθμό των παραγόντων k, τον ρυθμό εκμάθησης η, τους συντελεστές regularization των χρηστών λ_u και των αντικειμένων λ_i και τον συντελεστή regularization για τα biases χρηστών-αντικειμένων $\lambda_{b_{ui}}$.

Για κάθε συλλογή πραγματοποιήθηκε grid search για την εύρεση κατάλληλων τιμών των η, λ_u, λ_i και $\lambda_{b_{ui}}$. Οι τιμές που διερευνήθηκαν ήταν για τον ρυθμό εκπαίδευσης $\eta = [0.0001, 0.0005, 0.001, 0.005, 0.01, 0.02, 0.03]$ και για τους συντελεστές regularization $\lambda_u, \lambda_i, \lambda_{b_{ui}} = [0.01, 0.015, 0.05, 0.1, 0.15, 0.2, 0.5]$, κρατώντας σταθερό το $k = 10$. Ο μέγιστος αριθμός εποχών εκπαίδευσης ήταν 100 εποχές και χρησιμοποιήθηκε τεχνική early stopping with validation. Ο Πίνακας 7 περιέχει τις βέλτιστες τιμές που επιλέχθηκαν για κάθε συλλογή μέσω της μεθόδου grid search.

Πίνακας 7. Οι βέλτιστες τιμές των υπερπαραμέτρων για τον αλγόριθμο BMF ανά συλλογή δεδομένων

Συλλογή δεδομένων	η	λ_u	λ_i	$\lambda_{b_{ui}}$
MovieLens 100k	0.005	0.1	0.1	0.2
MovieLens 1M	0.01	0.2	0.01	0.2
Jester	0.0001	0.5	0.5	0.5
Book Crossing	0.01	0.1	0.1	0.1

4.2.3.6 Μοντέλο SVD++

Το μοντέλο SVD++ ενσωματώνει User-Item biases, καθώς και έμμεση πληροφορία σχετικά με ποια αντικείμενα αξιολόγησε ο χρήστης. Η εκπαίδευση γίνεται με στοχαστική κατάβαση δυναμικού. Διαθέτει 5 υπερπαραμέτρους: τον αριθμό των λανθανόντων παραγόντων k, τον ρυθμό εκμάθησης η, τον συντελεστή regularization λ, τους συντελεστές regularization για τα biases χρηστών-αντικειμένων $\lambda_{b_{ui}}$ και τον ρυθμό εκπαίδευσης των biases χρηστών-αντικειμένων $\eta_{b_{ui}}$.

Για κάθε συλλογή πραγματοποιήθηκε grid search για την εύρεση κατάλληλων τιμών των η, λ, $\eta_{b_{ui}}$ και $\lambda_{b_{ui}}$. Οι τιμές που διερευνήθηκαν για τους ρυθμούς εκπαίδευσης ήταν: $\eta = [0.001, 0.005, 0.01, 0.02, 0.03]$ και $\eta_{b_{ui}} = [0.07, 0.7]$. Οι τιμές που διερευνήθηκαν για τους συντελεστές regularization ήταν: $\lambda = [0.01, 0.05, 0.1, 0.2, 0.5, 1]$ και $\lambda_{b_{ui}} = [0.001, 0.005, 0.01, 0.33]$. Ο αριθμός των λανθανόντων παραγόντων ήταν σταθερός: $k = 10$. Ο μέγιστος αριθμός εποχών εκπαίδευσης ήταν 100 εποχές και χρησιμοποιήθηκε τεχνική early stopping with validation. Ο Πίνακας 8 περιέχει τις βέλτιστες τιμές που επιλέχθηκαν για κάθε συλλογή μέσω της μεθόδου grid search.

Πίνακας 8. Οι βέλτιστες τιμές των υπερπαραμέτρων για τον αλγόριθμο SVD++ ανά συλλογή δεδομένων

Συλλογή δεδομένων	η	λ	$\eta_{b_{ui}}$	$\lambda_{b_{ui}}$
MovieLens 100k	0.01	1	0.07	0.005
MovieLens 1M	0.005	0.05	0.7	0.33
Jester	0.005	0.5	0.7	0.33
Book Crossing	0.005	0.5	0.7	0.33

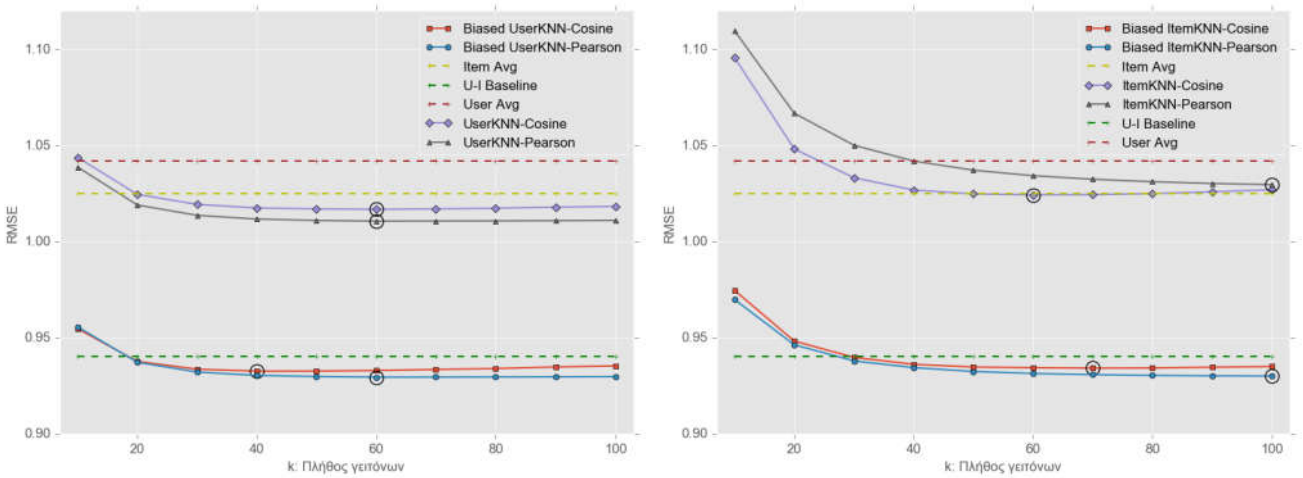
4.3 Αποτελέσματα πειραμάτων

Η μέτρηση της ακρίβειας των προβλέψεων έγινε χρησιμοποιώντας τις μετρικές RMSE και MAE. Σύμφωνα με τον (Pilászy, 2010), για την μέτρηση ακρίβειας προβλέψεων με RMSE,

τα 4 δεκαδικά ψηφία είναι αρκετά για την σύγκριση μεταξύ των μοντέλων. Την ίδια τακτική ακολουθούμε και για το MAE.

4.3.1 Αποτελέσματα στη συλλογή MovieLens 100k

Τα μοντέλα που βασίζονται σε KNN ακολουθούν δύο προσεγγίσεις: User-based και Item-based. Στην πρώτη ο αλγόριθμος προσπαθεί να εντοπίσει παρόμοιους χρήστες, ενώ στην δεύτερη παρόμοια αντικείμενα. Στο Σχήμα 16 απεικονίζονται τα αποτελέσματα των μοντέλων ομαδοποιημένα ανά προσέγγιση, δηλαδή στο αριστερό διάγραμμα βλέπουμε τα User-based μοντέλα και στο δεξιό τα Item-based. Επιπλέον απεικονίζονται και τρεις βασικοί εκτιμητές: με βάση τη μέση τιμή αξιολογήσεων ανά χρήστη (User Avg) και ανά αντικείμενο (Item Avg) και ο βασικός εκτιμητής που συνδυάζει τα biases χρηστών και αντικειμένων (U-I Baseline).



Σχήμα 16. Σύγκριση των μοντέλων User-based KNN (αριστερά) και Item-based KNN (δεξιά) στη συλλογή MovieLens-100k, περιλαμβάνοντας και τρεις βασικούς εκτιμητές. Με κύκλο σημειώνεται το σημείο ελαχίστου σφάλματος RMSE για κάθε μοντέλο.

Παρατηρούμε ότι γενικά η αύξηση του πλήθους των γειτόνων έχει ως αποτέλεσμα την μείωση του RMSE, μέχρι περίπου την τιμή $k = 60$, για την οποία κάποια μοντέλα παρουσιάζουν ελάχιστο RMSE και για τα υπόλοιπα το σφάλμα είναι πολύ κοντά στο ελάχιστο.

Για τις απλές προσεγγίσεις με βάση το KNN, τα μοντέλα User-based γενικά παρουσιάζουν χαμηλότερο σφάλμα RMSE σε σχέση με τα αντίστοιχα Item-based. Όμως, ενώ τα μοντέλα User-based είναι καλύτερα από τους βασικούς εκτιμητές User Avg και Item Avg, το μοντέλο Item-based Cosine οριακά καταφέρνει να ξεπεράσει τον βασικό εκτιμητή Item Avg (Πίνακας 9).

Όσον αφορά την σύγκριση μεταξύ των μοντέλων, φαίνεται ότι στις 3 από τις 4 προσεγγίσεις η συσχέτιση Pearson δίνει καλύτερο αποτέλεσμα. Πάντως τα αποτελέσματα των δύο μετρικών ομοιότητας συνημιτόνου και συσχέτισης Pearson είναι αρκετά κοντά μεταξύ τους.

Τέλος και στα δύο διαγράμματα φαίνεται ότι η χρήση βασικών εκτιμητών σε συνδυασμό με τα μοντέλα KNN, μειώνει πολύ το σφάλμα RMSE. Η μέση μείωση, συγκρίνοντας το ελάχιστο RMSE κάθε μοντέλου, είναι 8,2 % στην User based και 9,2 % στην Item based προσέγγιση. Βέβαια η μεγάλη μείωση δεν προκαλεί έκπληξη, καθώς ήδη το RMSE του βασικού εκτιμητή είναι αρκετά χαμηλότερο από τις απλές προσεγγίσεις με βάση το KNN.

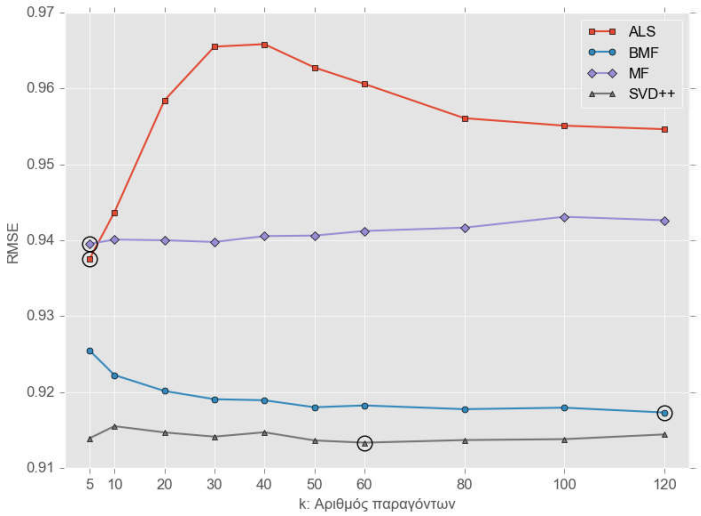
Ο Πίνακας 9 δείχνει τα αποτελέσματα των πειραμάτων στην συλλογή MovieLens 100k. Το βέλτιστο μοντέλο βασισμένο σε KNN είναι το User-based με biases και μετρική ομοιότητας συσχέτιση Pearson και παρουσιάζει ελάχιστο RMSE για k = 60.

Πίνακας 9: Αποτελέσματα ακρίβειας προβλέψεων βασικών εκτιμητών και τεχνικών βασισμένων σε KNN (MovieLens 100k). Με έντονη γραφή σημειώνεται η προσέγγιση με το ελάχιστο σφάλμα

Αλγόριθμος / Μετρικές		RMSE	+/- std dev	MAE	+/- std dev	Χρόνος εκπαίδευσης
Baselines	Global Average	1.1257	0.0029	0.9447	0.0029	0.004 sec
	User Average	1.0419	0.0049	0.8350	0.0029	0.004 sec
	Item Average	1.0249	0.0051	0.8173	0.0051	0.004 sec
	User-Item Baseline	0.9403	0.0061	0.7424	0.0049	0.23 sec
UserKNN-Cosine (k=60)		1.0168	0.0048	0.8048	0.0051	8.2 sec
UserKNN-Pearson (k=60)		1.0106	0.0059	0.8028	0.0054	8.4 sec
ItemKNN-Cosine (k=60)		1.0243	0.0056	0.8104	0.0039	10.7 sec
ItemKNN-Pearson (k=100)		1.0296	0.0055	0.8236	0.0032	10.2 sec
Biased UserKNN-Cosine (k=40)		0.9326	0.0060	0.7344	0.0053	8.8 sec
Biased UserKNN-Pearson (k=60)		**0.9295**	0.0056	**0.7294**	0.0045	14.1 sec
Biased ItemKNN-Cosine (k=70)		0.9342	0.0065	0.7350	0.0052	17.6 sec
Biased ItemKNN-Pearson (k=100)		0.9301	0.0064	0.7312	0.0049	18.4 sec

Η σύγκριση των μεθόδων βασισμένων σε μοντέλο (Σχήμα 17), δείχνει ότι η τεχνική της παραγοντοποίησης πινάκων (MF) έχει αρκετά χαμηλότερο σφάλμα από τις αντίστοιχες μεθόδους που είναι βασισμένες στη μνήμη (χωρίς biases). Επιπλέον, η χρήση των biases βελτιώνει κατά πολύ την απόδοση της παραγοντοποίησης πινάκων (BMF), όπως αντίστοιχα είδαμε και στις μεθόδους που είναι βασισμένες στη μνήμη (Σχήμα 16).

Η χρήση έμμεσης πληροφορίας στον αλγόριθμο (SVD++), βελτιώνει ακόμα περισσότερο την ακρίβεια των προβλέψεων, όμως το υπολογιστικό κόστος είναι πολύ μεγαλύτερο (Πίνακας 10).



Σχήμα 17. Σύγκριση των μεθόδων βασισμένων σε μοντέλο στη συλλογή MovieLens-100k. Με κύκλο σημειώνεται το σημείο ελαχίστου σφάλματος RMSE για κάθε μέθοδο.

Όσον αφορά την απόδοση του κάθε αλγορίθμου σε σχέση με τον αριθμό λανθανόντων παραγόντων, βλέπουμε ότι ο απλός MF και ο ALS παρουσιάζουν το ελάχιστο σφάλμα χρησιμοποιώντας πολύ λίγους παράγοντες (k = 5) και όσο αυξάνεται το k, αυξάνεται και το σφάλμα RMSE, μάλιστα η αύξηση είναι ιδιαίτερα έντονη στον ALS. Αυτό μπορεί να οφείλεται σε φαινόμενο υπερπροσαρμογής καθώς αυξάνεται ο αριθμός των διαθέσιμων δεδομένων, πράγμα που δεν παρατηρείται στα άλλα δύο μοντέλα που χρησιμοποιούν biases. Έτσι, το BMF παρουσιάζει μια καθοδική τάση του RMSE όσο αυξάνουν οι παράγοντες, ενώ ο SVD++ είναι σχετικά ασταθής για μικρό k, αλλά σταθεροποιείται για k > 50, παρουσιάζοντας ελάχιστο RMSE για k = 60.

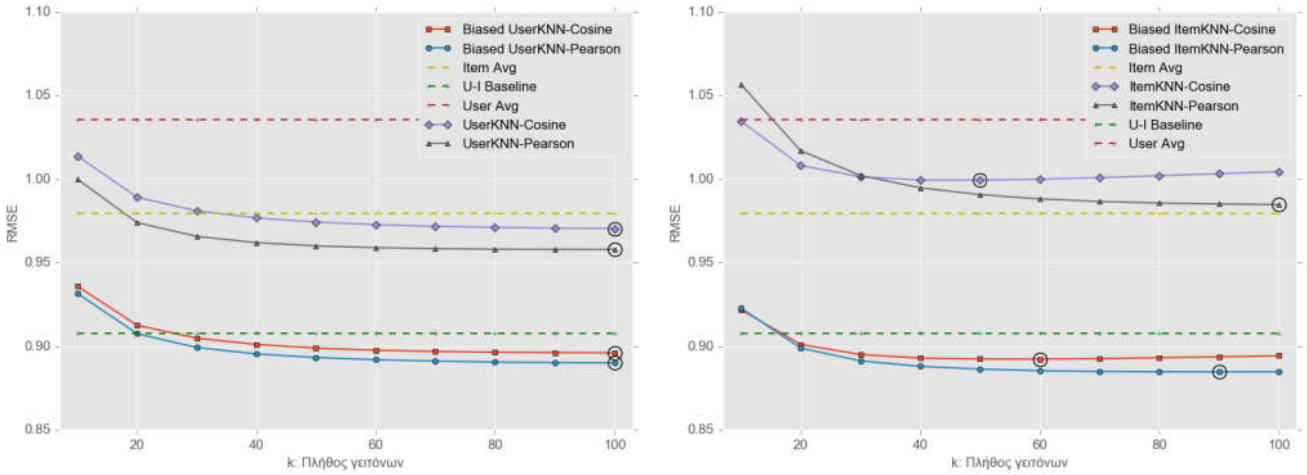
Ο Πίνακας 10 δείχνει τα αποτελέσματα των πειραμάτων στην συλλογή MovieLens 100k. Τα μοντέλα είναι εκπαιδευμένα στο πλήρες σύνολο εκπαίδευσης (χωρίς σύνολο επικύρωσης), οπότε το RMSE είναι χαμηλότερο. Η βέλτιστη προσέγγιση βασισμένη σε μοντέλο είναι ο αλγόριθμος SVD++ και παρουσιάζει ελάχιστο RMSE για k = 50 παράγοντες.

Πίνακας 10. Αποτελέσματα ακρίβειας προβλέψεων τεχνικών με βάση μοντέλα (MovieLens 100k). Με έντονη γραφή σημειώνεται η προσέγγιση με το ελάχιστο σφάλμα

Μετρικές / Αλγόριθμος	RMSE	+/- std dev	MAE	+/- std dev	Χρόνος εκπαίδευσης
ALS (5 factors)	0.9376	0.0054	0.7382	0.0044	0.43 sec
MF (5 factors)	0.9305	0.0039	0.7355	0.0028	7.3 sec
BMF (120 factors)	0.9121	0.0056	0.7224	0.0044	73.2 sec
SVD++ (60 factors)	**0.9073**	0.0056	**0.7153**	0.0042	932 sec

4.3.2 Αποτελέσματα στη συλλογή MovieLens 1M

Αντίστοιχα με τα αποτελέσματα για τη συλλογή MovieLens-100k, στο Σχήμα 18 απεικονίζονται τα αποτελέσματα των μοντέλων ομαδοποιημένα ανά προσέγγιση, δηλαδή στο αριστερό διάγραμμα βλέπουμε τα User-based μοντέλα και στο δεξιό τα Item-based. Επιπλέον απεικονίζονται και τρεις βασικοί εκτιμητές: με βάση τη μέση τιμή αξιολογήσεων ανά χρήστη (User Avg) και ανά αντικείμενο (Item Avg) και ο βασικός εκτιμητής που συνδυάζει τα biases χρηστών και αντικειμένων (U-I Baseline).



Σχήμα 18. Σύγκριση των μοντέλων User-based KNN (αριστερά) και Item-based KNN (δεξιά) στη συλλογή MovieLens-1M, περιλαμβάνοντας και τρεις βασικούς εκτιμητές. Με κύκλο σημειώνεται το σημείο ελαχίστου σφάλματος RMSE για κάθε μοντέλο.

Παρατηρούμε ότι γενικά η αύξηση του πλήθους των γειτόνων έχει ως αποτέλεσμα την μείωση του RMSE στα User-based μοντέλα, στα οποία το ελάχιστο σφάλμα παρατηρείται στην τιμή $k = 100$. Στα Item-based μοντέλα η αύξηση του k μειώνει το σφάλμα και οι ελάχιστες τιμές παρατηρούνται για $k = 50$ και 60 για την μετρική συνημιτόνου, αλλά για αρκετά υψηλότερα $k = 90$ και 100 για την μετρική με βάση την συσχέτιση του Pearson. Επίσης, η προσέγγιση Item-based Cosine για τιμές $k > 50$ παρουσιάζει σχετικά μεγάλη αύξηση.

Για τις απλές προσεγγίσεις με βάση το KNN, τα μοντέλα User-based γενικά παρουσιάζουν χαμηλότερο σφάλμα RMSE σε σχέση με τα αντίστοιχα Item-based. Όμως, ενώ τα μοντέλα User-based είναι καλύτερα από τους βασικούς εκτιμητές User Avg και Item Avg, τα μοντέλα Item-based δεν ξεπερνούν τον βασικό εκτιμητή Item Avg (Πίνακας 11).

Σχετικά την σύγκριση των μοντέλων μεταξύ τους, φαίνεται ότι και στις 4 περιπτώσεις, η μετρική ομοιότητας με συσχέτιση Pearson δίνει καλύτερο αποτέλεσμα από την μετρική ομοιότητας συνημιτόνου.

Η χρήση του βασικού εκτιμητή σε συνδυασμό με τα μοντέλα KNN, μειώνει πολύ το σφάλμα RMSE και στις δύο προσεγγίσεις User based και Item based. Η μέση μείωση, συγκρίνοντας

το ελάχιστο RMSE κάθε μοντέλου, αγγίζει το 7,4 % και 10,4 % για την προσέγγιση User based και Item based αντίστοιχα. Όπως και στη συλλογή MovieLens-100k, το RMSE του βασικού εκτιμητή είναι αρκετά χαμηλότερο από αυτό των απλών προσεγγίσεων με βάση το KNN.

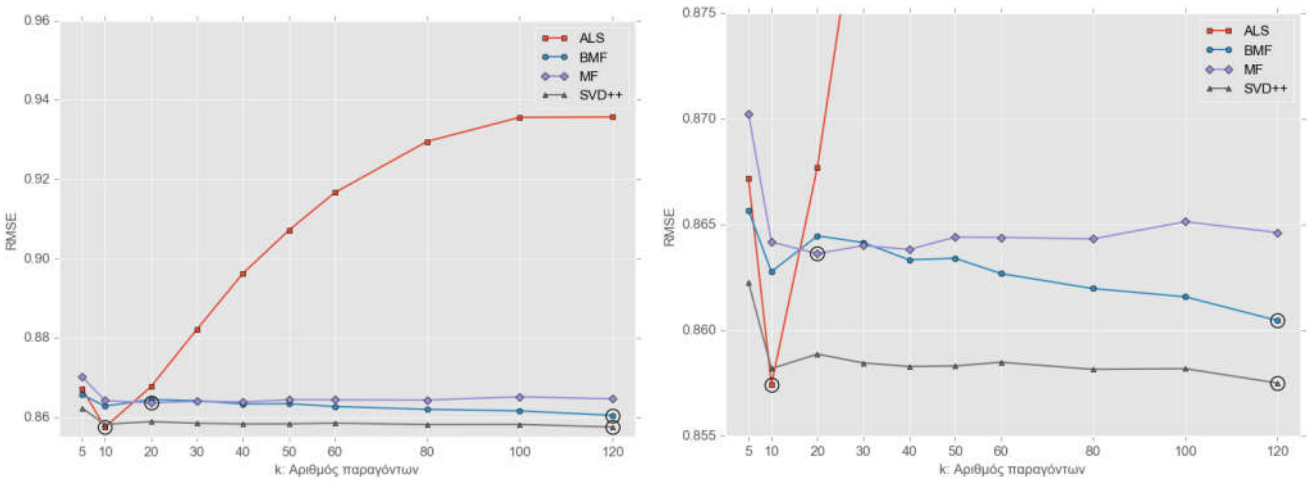
Ο Πίνακας 11 δείχνει τα αποτελέσματα των πειραμάτων στην συλλογή MovieLens 1M. Το βέλτιστο μοντέλο βασισμένο σε KNN είναι το Item-based με biases και μετρική ομοιότητας συσχέτιση Pearson και παρουσιάζει ελάχιστο RMSE για k = 90.

Πίνακας 11: Αποτελέσματα ακρίβειας προβλέψεων βασικών εκτιμητών και τεχνικών βασισμένων σε KNN (MovieLens 1M). Με έντονη γραφή σημειώνεται η προσέγγιση με το ελάχιστο σφάλμα

Αλγόριθμος \ Μετρικές		RMSE	+/- τυπική απόκλιση	MAE	+/- τυπική απόκλιση	Χρόνος εκπαίδευσης
Baselines	Global Average	1.1171	0.0014	0.9339	0.0013	0.03 sec
	User Average	1.0355	0.0014	0.8290	0.0011	0.03 sec
	Item Average	0.9795	0.0016	0.7824	0.0012	0.03 sec
	UserItemBaseline	0.9077	0.0020	0.7170	0.0013	1.7 sec
UserKNN-Cosine (k=100)		0.9703	0.0015	0.7660	0.0010	310.8 sec
UserKNN-Pearson (k=100)		0.9579	0.0020	0.7629	0.0014	310.4 sec
ItemKNN-Cosine (k=50)		0.9994	0.0015	0.7810	0.0009	116.3 sec
ItemKNN-Pearson (k=100)		0.9847	0.0015	0.7894	0.0012	161.4 sec
Biased UserKNN-Cosine (k=100)		0.8960	0.0021	0.7076	0.0013	470.5 sec
Biased UserKNN-Pearson (k=100)		0.8900	0.0023	0.7005	0.0016	411.8 sec
Biased ItemKNN-Cosine (k=60)		0.8923	0.0020	0.7019	0.0013	135.1 sec
Biased ItemKNN-Pearson (k=90)		**0.8847**	0.0024	**0.6956**	0.0016	159.7 sec

Η σύγκριση των μεθόδων βασισμένων σε μοντέλο (Σχήμα 19), δείχνει ότι η τεχνική της παραγοντοποίησης πινάκων (MF) έχει αρκετά χαμηλότερο σφάλμα από τις αντίστοιχες μεθόδους που είναι βασισμένες στη μνήμη (χωρίς biases). Επίσης, η χρήση των biases βελτιώνει το αποτέλεσμα και μάλιστα η βελτίωση είναι μεγαλύτερη καθώς αυξάνεται ο αριθμός των λανθανόντων παραγόντων (BMF).

Ο αλγόριθμος ALS παρουσιάζει μια έντονη διακύμανση, έχοντας ελάχιστη τιμή RMSE για k = 10, η οποία πολύ κοντά στις ελάχιστες τιμές των άλλων αλγορίθμων, αλλά καθώς αυξάνεται το k, αυξάνεται πάρα πολύ και το RMSE. Αυτή η συμπεριφορά οφείλεται σε υπερπροσαρμογή στα δεδομένα εκπαίδευσης καθώς αυξάνεται ο αριθμός των λανθανόντων παραγόντων.



Σχήμα 19. Σύγκριση των μεθόδων βασισμένων σε μοντέλο στη συλλογή MovieLens-1M. Με κύκλο σημειώνεται το σημείο ελαχίστου σφάλματος RMSE για κάθε μέθοδο. Αριστερά το πλήρες διάγραμμα, δεξιά το διάγραμμα είναι προσαρμοσμένο στις κατώτερες τιμές σφάλματος για καλύτερη ευκρίνεια στη σύγκριση μεταξύ MF, BMF και SVD++

Ο αλγόριθμος MF δείχνει μια σχετική σταθερότητα σε σχέση με την μεταβολή του k, με αύξουσα τάση του σφάλματος για μεγαλύτερες τιμές του k, παρουσιάζοντας ελάχιστο σφάλμα για k = 20. Αντίθετα ο BMF αν και για μικρές τιμές του k έχει χαμηλότερη ακρίβεια από τον MF, καθώς αυξάνεται το k παρουσιάζει μεγάλη βελτίωση, με ελάχιστο σφάλμα στην τιμή k = 120. Τέλος, ο SVD++ έχει συνολικά μεγαλύτερη ακρίβεια από τους άλλους τρεις, χωρίς ιδιαίτερες διακυμάνσεις για διάφορες τιμές του k, παρουσιάζοντας όμως ελάχιστο σφάλμα για k = 120.

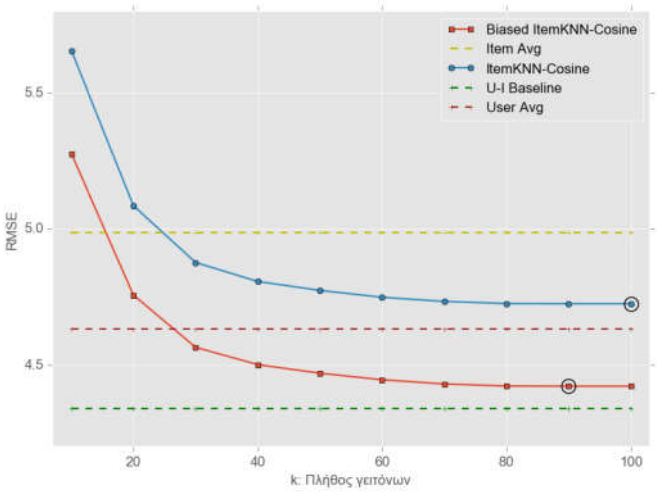
Ο Πίνακας 12 δείχνει τα αποτελέσματα των πειραμάτων στην συλλογή MovieLens 1M. Τα μοντέλα είναι εκπαιδευμένα στο πλήρες σύνολο εκπαίδευσης (χωρίς σύνολο επικύρωσης), οπότε το RMSE είναι χαμηλότερο. Η βέλτιστη προσέγγιση βασισμένη σε μοντέλο είναι ο αλγόριθμος SVD++ και παρουσιάζει ελάχιστο RMSE για k = 120 παράγοντες.

Πίνακας 12. Αποτελέσματα ακρίβειας προβλέψεων τεχνικών με βάση μοντέλα (MovieLens 1M). Με έντονη γραφή σημειώνεται η προσέγγιση με το ελάχιστο σφάλμα

Μετρικές / Αλγόριθμος	RMSE	+/- τυπική απόκλιση	MAE	+/- τυπική απόκλιση	Χρόνος εκπαίδευσης
ALS (10 factors)	0.8574	0.0025	0.6714	0.0016	9.2 sec
MF (20 factors)	0.8566	0.0022	0.6744	0.0016	38.5 sec
BMF (120 factors)	0.8543	0.0021	0.6767	0.0015	307.2 sec
SVD++ (120 factors)	**0.8505**	0.0023	**0.6677**	0.0014	34286 sec (9.5h)

4.3.3 Αποτελέσματα στη συλλογή Jester-1

Στην συλλογή Jester-1 η τεχνικές με βάση την μνήμη αρχίζουν να δείχνουν τα όρια τους, λόγω του μεγάλου αριθμού χρηστών της συλλογής (≈ 25000). Έτσι, στον υπολογιστή που πραγματοποιήθηκαν τα πειράματα, η τεχνική User Based με μετρική ομοιότητας συνημιτόνου κατά τον υπολογισμό του πίνακα ομοιότητας χρησιμοποιεί περίπου 25 Gb μνήμης και ο χρόνος για την εκπαίδευση ενός μοντέλου με Cross Validation απαιτεί περίπου 40 ώρες, συνεπώς το χρονικό κόστος πραγματοποίησης grid search είναι απαγορευτικό. Στην περίπτωση της τεχνικής User-based με μετρική ομοιότητας την συσχέτιση Pearson, ο αλγόριθμος δεν ολοκλήρωσε το πείραμα λόγω έλλειψης μνήμης. Τα προβλήματα μη ολοκλήρωσης λόγω χρόνου και μνήμης σημειώνονται με *time_error* και *mem_error* στα αποτελέσματα που δείχνει ο Πίνακας 13). Τέλος οι προσεγγίσεις Item based με μετρική ομοιότητας την συσχέτιση Pearson, παρουσίασαν τεχνικό πρόβλημα κατά τον υπολογισμό της ομοιότητας και δεν ολοκληρώθηκαν τα πειράματα (σημειώνεται με *error)*.



Σχήμα 20. Σύγκριση των μοντέλων Item-based KNN στη συλλογή Jester-1, περιλαμβάνοντας και τρεις βασικούς εκτιμητές. Με κύκλο σημειώνεται το σημείο ελαχίστου σφάλματος RMSE για κάθε μοντέλο

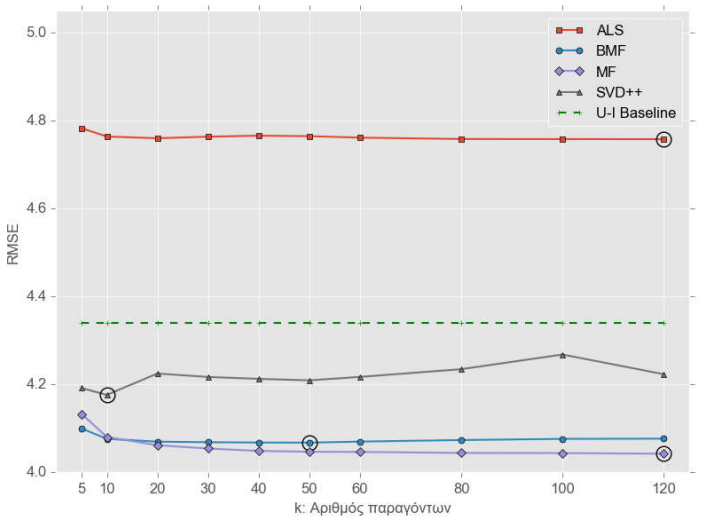
Η ακρίβεια της τεχνικής Item based με μετρική ομοιότητας συνημιτόνου φαίνεται να είναι ικανοποιητική, ξεπερνώντας τον βασικό εκτιμητή Item Avg, αλλά όχι τον User Avg. Η ενσωμάτωση biases επίσης βελτιώνει αρκετά την ακρίβεια, χωρίς όμως να μπορεί να ξεπεράσει τον βασικό εκτιμητή User-Item baseline.

Η μέγιστη ακρίβεια των δύο προσεγγίσεων παρουσιάζεται για μεγάλα πλήθη γειτόνων και δείχνει να σταθεροποιείται σε εκείνη την περιοχή τιμών. Ο Πίνακας 13 δείχνει τα αποτελέσματα των πειραμάτων στην συλλογή Jester-1. Το βέλτιστο μοντέλο βασισμένο σε KNN είναι το Item-based με biases και μετρική ομοιότητας συνημιτόνου και παρουσιάζει ελάχιστο RMSE για k = 90, παρόλο που ακόμα χαμηλότερο σφάλμα παρουσιάζει ο βασικός εκτιμητής User-Item baseline.

Πίνακας 13: Αποτελέσματα ακρίβειας προβλέψεων βασικών εκτιμητών και τεχνικών βασισμένων σε KNN (Jester-1). Με έντονη γραφή σημειώνεται η προσέγγιση με το ελάχιστο σφάλμα

Αλγόριθμος \ Μετρικές		RMSE	+/- τυπική απόκλιση	MAE	+/- τυπική απόκλιση	Χρόνος εκπαίδευσης
Baselines	Global Average	5.2358	0.0059	4.3706	0.0061	0.07 sec
	User Average	4.6323	0.0078	3.7269	0.0058	0.05 sec
	Item Average	4.9870	0.0059	4.1146	0.0057	0.05 sec
	UserItemBaseline	**4.3390**	0.0075	**3.4472**	0.0060	3.58 sec
UserKNN-Cosine		*time_error*	-	-	-	40 h
UserKNN-Pearson		*mem_error*	-	-	-	-
ItemKNN-Cosine (k=100)		4.7244	0.0075	3.7522	0.0061	51.5 sec
ItemKNN-Pearson		*error*	-	-	-	-
Biased UserKNN-Cosine		*time_error*	-	-	-	40 h
Biased UserKNN-Pearson		*mem_error*	-	-	-	-
Biased ItemKNN-Cosine (k=90)		**4.4213**	0.0120	**3.4884**	0.0069	78.2 sec
Biased ItemKNN-Pearson		*error*	-	-	-	-

Ο αλγόριθμος MF έχει την καλύτερη απόδοση στη συλλογή Jester-1 και η χρήση biases (BMF) όχι μόνο δεν βοηθά, αλλά αντίθετα αυξάνει το σφάλμα RMSE. Αντίθετα, την χειρότερη απόδοση έχει ο αλγόριθμος ALS, ο οποίος διατηρεί σταθερά υψηλότερο σφάλμα από τους υπολοίπους και δεν μπορεί να ξεπεράσει τον βασικό εκτιμητή User-Item Baseline. (Σχήμα 21).



Σχήμα 21. Σύγκριση των μεθόδων βασισμένων σε μοντέλο στη συλλογή Jester-1. Με κύκλο σημειώνεται το σημείο ελαχίστου σφάλματος RMSE για κάθε μέθοδο.

Η συμπεριφορά των αλγορίθμων ALS και MF σε σχέση με την αύξηση των λανθανόντων παραγόντων δείχνει να είναι σταθερή, παρουσιάζοντας μια μικρή καθοδική τάση στο

σφάλμα. Από την άλλη ο αλγόριθμος BMF παρουσιάζει ελάχιστο RMSE για k = 50 και στη συνέχεια παρουσιάζει ανοδική τάση.

Στην περίπτωση του SVD++ η χρήση έμμεσης πληροφορίας φαίνεται να περιπλέκει τα πράγματα, κάνοντας την συμπεριφορά του αλγορίθμου ασταθή σε σχέση με την αύξηση του k. Επιπλέον η ακρίβεια είναι αρκετά χειρότερη σε σχέση με τα μοντέλα MF και BMF, σαφώς όμως καλύτερη από τις μεθόδους βασισμένες στη μνήμη και τους βασικούς εκτιμητές. Αυτό πιθανόν να οφείλεται στο μικρό πλήθος αντικειμένων που περιέχει η συλλογή (100), τα οποία δεν επαρκούν για να δώσουν επιπλέον πληροφορία στον αλγόριθμο.

Ο Πίνακας 14 δείχνει τα αποτελέσματα των πειραμάτων στην συλλογή Jester-1. Τα μοντέλα είναι εκπαιδευμένα στο πλήρες σύνολο εκπαίδευσης (χωρίς σύνολο επικύρωσης), οπότε το RMSE είναι χαμηλότερο. Η βέλτιστη προσέγγιση βασισμένη σε μοντέλο είναι ο αλγόριθμος MF και παρουσιάζει ελάχιστο RMSE για k = 120 παράγοντες.

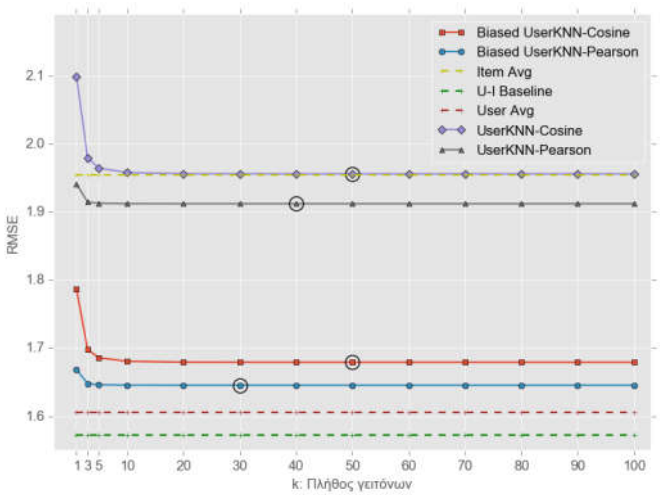
Πίνακας 14. Αποτελέσματα ακρίβειας προβλέψεων τεχνικών με βάση μοντέλα (Jester-1). Με έντονη γραφή σημειώνεται η προσέγγιση με το ελάχιστο σφάλμα

Μετρικές / Αλγόριθμος	RMSE	+/- τυπική απόκλιση	MAE	+/- τυπική απόκλιση	Χρόνος εκπαίδευσης
ALS (120 factors)	4.7579	0.0068	3.7490	0.0061	803.2 sec
MF (120 factors)	**4.0069**	0.0093	**3.1180**	0.0068	897.9 sec
BMF (50 factors)	4.0327	0.0094	3.1396	0.0066	500.2 sec
SVD++ (10 factors)	4.1535	0.0234	3.2337	0.0093	94.4 sec

4.3.4 Αποτελέσματα στη συλλογή Book Crossing

Η συλλογή Book Crossing περιέχει ένα πολύ μεγάλο αριθμό αντικειμένων (156.069), το οποίο αποτελεί πρόβλημα κατά τον υπολογισμό της ομοιότητας μεταξύ αντικειμένων. Έτσι, οι τεχνικές Item-based δεν μπόρεσαν να ολοκληρώσουν τα πειράματα λόγω έλλειψης μνήμης (σημειώνονται με *mem_error* στα αποτελέσματα που δείχνει ο Πίνακας 15).

Οι τεχνικές User based δεν αποδίδουν καλά καθώς δεν μπορούν να ξεπεράσουν τους βασικούς εκτιμητές και επιπλέον η μεταβολή του πλήθους των γειτόνων δείχνει να έχει μικρή επίδραση στο RMSE. Συγκεκριμένα, για πολύ λίγους γείτονες το σφάλμα είναι σχετικά υψηλό, αλλά καθώς το πλήθος k αυξάνεται, το σφάλμα μειώνεται και από το σημείο ελαχίστου RMSE και μετά παραμένει πρακτικά σταθερό (Σχήμα 22).



Σχήμα 22. Σύγκριση των μοντέλων User-based KNN στη συλλογή Book-Crossing, περιλαμβάνοντας και τρεις βασικούς εκτιμητές. Με κύκλο σημειώνεται το σημείο ελαχίστου σφάλματος RMSE για κάθε μοντέλο

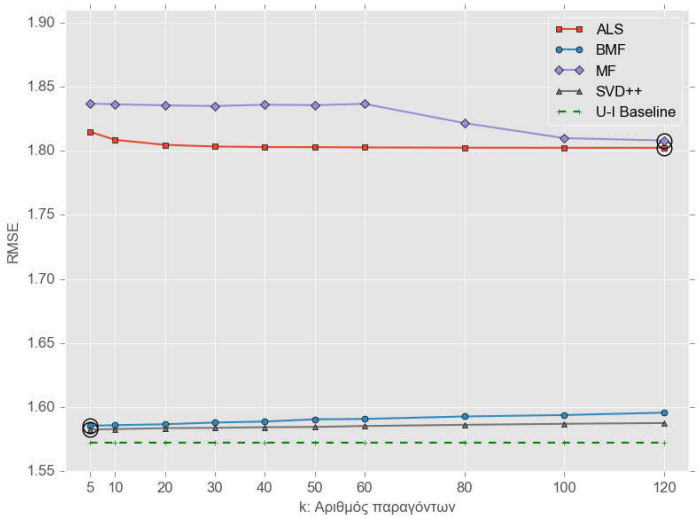
Αυτό σημαίνει ότι λόγω της μεγάλης αραιότητας της συλλογής, ο αλγόριθμος πέρα από το σημείο ελαχίστου σφάλματος (30 – 50 γείτονες) δεν μπορεί να βρει άλλους γείτονες που να είναι αρκετά κοντά ώστε να βελτιώσουν την ακρίβεια των προβλέψεων.

Ο Πίνακας 15 δείχνει τα αποτελέσματα των πειραμάτων στην συλλογή Book Crossing. Το βέλτιστο μοντέλο βασισμένο σε KNN είναι το User-based με biases και μετρική ομοιότητας συσχέτιση Pearson και παρουσιάζει ελάχιστο RMSE για $k = 30$, παρόλο που ακόμα χαμηλότερο σφάλμα παρουσιάζουν οι βασικοί εκτιμητές User Avg και User-Item baseline.

Πίνακας 15: Αποτελέσματα ακρίβειας προβλέψεων βασικών εκτιμητών και τεχνικών βασισμένων σε KNN (Book Crossing). Με έντονη γραφή σημειώνεται η προσέγγιση με το ελάχιστο σφάλμα

Αλγόριθμος \ Μετρικές		RMSE	+/- τυπική απόκλιση	MAE	+/- τυπική απόκλιση	Χρόνος εκπαίδευσης
Baselines	Global Average	1.8370	0.0041	1.4922	0.0027	0.01 sec
	User Average	**1.6055**	0.0034	1.2200	0.0017	0.01 sec
	Item Average	1.9547	0.0056	1.5420	0.0042	0.01 sec
	UserItemBaseline	**1.5724**	0.0042	1.1984	0.0025	3.45 sec
UserKNN-Cosine (k=90)		1.9557	0.0058	1.5471	0.0044	18.4 sec
UserKNN-Pearson (k=40)		1.9118	0.0074	1.5298	0.0050	36 sec
ItemKNN-Cosine		*mem_error*	-	-	-	-
ItemKNN-Pearson		*mem_error*	-	-	-	-
Biased UserKNN-Cosine (k=50)		1.6791	0.0048	1.2681	0.0037	41 sec
Biased UserKNN-Pearson (k=30)		**1.6456**	0.0054	**1.2457**	0.0037	61.3 sec
Biased ItemKNN-Cosine		*mem_error*	-	-	-	-
Biased ItemKNN-Pearson		*mem_error*	-	-	-	-

Οι μέθοδοι βασισμένες σε μοντέλο παρουσιάζουν επίσης χαμηλή απόδοση, καθώς δεν μπορούν να ξεπεράσουν το βασικό εκτιμητή User-Item baseline. Οι αλγόριθμοι ALS και MF βρίσκονται αρκετά κοντά και έχουν ελάχιστο σφάλμα για k = 120. Η χρήση των biases στον αλγόριθμο BMF βελτιώνει πολύ την απόδοση της παραγοντοποίησης πινάκων και ο SVD++ έχει λίγο μεγαλύτερη ακρίβεια από τον BMF, όμως και οι δύο αλγόριθμοι παρουσιάζουν ελάχιστο σφάλμα για μικρή τιμή k = 5 και στη συνέχεια έχουν ανοδική τάση.



Σχήμα 23. Σύγκριση των μεθόδων βασισμένων σε μοντέλο στη συλλογή Book Crossing. Με κύκλο σημειώνεται το σημείο ελαχίστου σφάλματος RMSE για κάθε μέθοδο.

Ο Πίνακας 16 δείχνει τα αποτελέσματα των πειραμάτων στη συλλογή Book Crossing. Τα μοντέλα είναι εκπαιδευμένα στο πλήρες σύνολο εκπαίδευσης (χωρίς σύνολο επικύρωσης), οπότε το RMSE είναι χαμηλότερο. Η βέλτιστη προσέγγιση βασισμένη σε μοντέλο είναι ο αλγόριθμος SVD++ και παρουσιάζει ελάχιστο RMSE για k = 5 παράγοντες.

Πίνακας 16. Αποτελέσματα ακρίβειας προβλέψεων τεχνικών με βάση μοντέλα (Book Crossing). Με έντονη γραφή σημειώνεται η προσέγγιση με το ελάχιστο σφάλμα

Μετρικές / Αλγόριθμος	RMSE	+/- τυπική απόκλιση	MAE	+/- τυπική απόκλιση	Χρόνος εκπαίδευσης
ALS (120 factors)	1.8024	0.0043	1.4505	0.0029	329.8 sec
MF (80 factors)	1.8332	0.0041	1.4828	0.0033	21.3 sec
BMF (5 factors)	1.5804	0.0040	**1.2165**	0.0022	23.2 sec
SVD++ (5 factors)	**1.5774**	0.0048	1.2212	0.0032	908.7 sec

4.3.5 Σύγκριση αποτελεσμάτων μεταξύ διαφορετικών κλιμάκων αξιολόγησης

Οι συλλογές δεδομένων χρησιμοποιούν διαφορετικές κλίμακες αξιολόγησης, συνεπώς οι τιμές των μετρικών RMSE και MSE δεν μπορούν να συγκριθούν μεταξύ συλλογών με διαφορετικές αξιολογικές κλίμακες, π.χ. MovieLens και Jester. Παρόλα αυτά μπορεί να γίνει

σύγκριση αν οι τιμές κανονικοποιηθούν με βάση το εύρος της κάθε κλίμακας (Dror et al., 2012).

Για να υπολογίσουμε το εύρος κάθε κλίμακας χρησιμοποιούμε την σχέση:

$$εύρος = r_{max} - r_{min}$$

όπου r_{max} η μέγιστη τιμή αξιολόγησης και r_{min}η ελάχιστη τιμή αξιολόγησης. Τα στοιχεία για κάθε συλλογή παρουσιάζονται στον παρακάτω πίνακα:

Πίνακας 17: Εύρος κλίμακας αξιολόγησης

Συλλογή δεδομένων	r_{min}	r_{max}	**εύρος**
MovieLens 100k – 1M	1	5	4
Jester	-10	10	20
Book Crossing	1	10	9

Στη συνέχεια μπορούμε να αντιστοιχήσουμε γραμμικά τις τιμές των μεγαλυτέρων σε εύρος κλιμάκων στις τιμές των μικρότερων σε εύρος (MovieLens), υπολογίζοντας απλά τον λόγο του εύρους της κάθε κλίμακας με το εύρος της κλίμακας της συλλογής MovieLens. Έτσι:

$$\lambda_{Jester} = \frac{εύρος_{Jester}}{εύρος_{MovieLens}} = \frac{20}{4} = 5$$

και

$$\lambda_{Book\ Crossing} = \frac{εύρος_{Book\ Crossing}}{εύρος_{MovieLens}} = \frac{9}{4} = 2{,}25$$

Δηλαδή το εύρος της Jester και της Book Crossing είναι 5 και 2,25 φορές μεγαλύτερο από το εύρος της MovieLens αντίστοιχα. Για να κανονικοποιήσουμε τις τιμές των μετρικών αρκεί να διαιρέσουμε την κάθε τιμή με το αντίστοιχο λ. Για παράδειγμα το Global Average RMSE στο Book Crossing είναι 1.8370. Διαιρώντας την τιμή με το $\lambda_{Book\ Crossing}$ = 2,25 λαμβάνουμε την κανονικοποιημένη τιμή 0.8164 η οποία είναι συγκρίσιμη με το Global Average RMSE του MovieLens 1M που είναι 1.1171.

5

Τεχνικές λεπτομέρειες

Στο κεφάλαιο αυτό αναφέρονται τα προγραμματιστικά εργαλεία που χρησιμοποιήθηκαν για την υλοποίηση του κώδικα που χρησιμοποιήθηκε κατά την εκπόνηση της διπλωματικής εργασίας, καθώς και κάποιες τεχνικές λεπτομέρειες σχετικά με την απόδοση του κώδικα.

5.1 Πλατφόρμες και προγραμματιστικά εργαλεία

Ο κώδικας για την υλοποίηση της συγκριτικής μελέτης, γράφτηκε σε python και για την αξιολόγηση των αλγορίθμων χρησιμοποιήθηκαν γνωστές και δοκιμασμένες βιβλιοθήκες ανοιχτού κώδικα για συστήματα συστάσεων.

5.1.1 Python

Η Python είναι μια εύκολη στην εκμάθηση, ισχυρή γλώσσα προγραμματισμού. Έχει αποδοτικές δομές δεδομένων υψηλού επιπέδου και μια απλή αλλά αποτελεσματική προσέγγιση στον αντικειμενοστρεφή προγραμματισμό. Η κομψή σύνταξη της Python και οι δυναμικοί τύποι της, μαζί με τη λειτουργία της ως διερμηνευόμενης (αντί μεταγλωττιζόμενης) γλώσσας, την καθιστούν την ιδανική γλώσσα για δημιουργία σεναρίων εντολών και για ταχεία ανάπτυξη εφαρμογών σε πολλούς τομείς και στις περισσότερες πλατφόρμες (Swaroop, 2003).

Ένα βασικό χαρακτηριστικό της Python είναι ότι διαθέτει μια τεράστια συλλογή πακέτων (packages), δηλαδή βιβλιοθηκών, δίνοντας την δυνατότητα να χρησιμοποιηθεί εύκολα για μεγάλη ποικιλία εφαρμογών, όπως επιστημονικές και εφαρμογές που απαιτούν μαθηματικούς υπολογισμούς. Σε αυτό συμβάλλει και το γεγονός ότι αναπτύσσεται ως ανοιχτό λογισμικό και έτσι οποιοσδήποτε μπορεί εύκολα να δημιουργήσει βιβλιοθήκες και να τις προσφέρει στην κοινότητα ανοιχτού λογισμικού.

Στην παρούσα εργασία χρησιμοποιήθηκε η Python 2, καθώς κάποιες βιβλιοθήκες δεν είναι ακόμα συμβατές με την νεότερη έκδοση 3. Για λόγους ευκολίας χρησιμοποιήθηκε μια διανομή Python επιστημονικού χαρακτήρα, η Anaconda Python ("Anaconda Python," n.d.). Η διανομή Anaconda περιλαμβάνει τα γνωστότερα πακέτα για python, που χρησιμοποιούνται για επιστημονικές εφαρμογές, όπως numpy, scipy, matplotlib, καθώς και το Spyder, ένα Ολοκληρωμένο Περιβάλλον Ανάπτυξης (IDE) για Python.

5.1.2 Scikit-learn

Η γνωστότερη βιβλιοθήκη μηχανικής μάθησης σε python είναι η scikit-learn[5]. Η scikit-learn παρέχει αλγορίθμους για ταξινόμηση, παλινδρόμηση, συσταδοποίηση αλλά και μείωση διαστατικότητας. Επίσης, περιλαμβάνει modules για εξαγωγή χαρακτηριστικών, προεπεξεργασία δεδομένων και αξιολόγηση μοντέλων.

Στην παρούσα εργασία η βιβλιοθήκη scikit-learn χρησιμοποιήθηκε για τον διαχωρισμό των συλλογών δεδομένων σε τυχαίες κατατμήσεις (folds) για να χρησιμοποιηθούν στην μέθοδο 5-fold Cross Validation, καθώς και για την δημιουργία του πλέγματος παραμέτρων για την μέθοδο grid search.

5.1.3 Python Data Analysis Library (pandas)

Η βιβλιοθήκη pandas[6] παρέχει αποδοτικές και εύκολες στη διαχείριση δομές δεδομένων σε μορφή πινάκων, καθώς και διάφορα εργαλεία ανάλυσης δεδομένων για την γλώσσα Python.

Στην παρούσα εργασία η βιβλιοθήκη pandas χρησιμοποιήθηκε για την φόρτωση των αρχείων csv των συλλογών δεδομένων και την προεπεξεργασία τους, όπου χρειαζόταν (π.χ. στη συλλογή book-crossing).

---

[5] http://scikit-learn.org/

[6] http://pandas.pydata.org/

5.1.4 MyMediaLite Recommender System Library

Η MyMediaLite Library[7] είναι μια βιβλιοθήκη αλγορίθμων για Συστήματα Συστάσεων, σχεδιασμένη για το προγραμματιστικό περιβάλλον CLR (αλλιώς γνωστό και ως .NET). Είναι βιβλιοθήκη ελεύθερου λογισμικού/λογισμικού ανοιχτού κώδικα, υλοποιημένη σε γλώσσα C# και μπορεί να εκτελεστεί σε όλες τις υπολογιστικές πλατφόρμες που υποστηρίζονται από το Mono project.

Διαθέτει αλγορίθμους για Συνεργατικό Φιλτράρισμα, με χρήση explicit αλλά και implicit δεδομένων και λύνει το πρόβλημα της πρόβλεψης αξιολογήσεων και της κατάταξης αντικειμένων.

Στα πειράματα χρησιμοποιήθηκε η έκδοση 3.11 και η κλήση των αλγορίθμων έγινε από γλώσσα Python μέσω ενός module που επιτρέπει την διασύνδεση .NET assemblies μέσα από python scripts. Το εν λόγω python module ονομάζεται *pythonnet*.

Στην παρούσα εργασία η βιβλιοθήκη MyMediaLite χρησιμοποιήθηκε για την αξιολόγηση των βασικών εκτιμητών και των αλγορίθμων Παραγοντοποίησης Πινάκων MF, BMF και SVD++.

5.1.5 Surprise python recommender system library

Η βιβλιοθήκη Surprise[8] είναι μια βιβλιοθήκη ανοιχτού κώδικα για κατασκευή και αξιολόγηση συστημάτων συστάσεων σε Python. Διαθέτει αλγορίθμους για μεθόδους βασισμένες στη μνήμη, αλλά και βασισμένες σε μοντέλα. Επίσης, παρέχει μεθόδους για την εύκολη διαχείριση των συλλογών δεδομένων και την εφαρμογή μεθόδων grid search και cross validation.

Στην παρούσα εργασία η βιβλιοθήκη Surprise χρησιμοποιήθηκε για την αξιολόγηση των μεθόδων με βάση τη μνήμη User-based KNN και Item-based KNN, καθώς στην πράξη διαπιστώθηκε ότι οι υλοποιήσεις τους ήταν πιο γρήγορες από τις αντίστοιχες της βιβλιοθήκης MyMediaLite.

[7] http://mymedialite.net/

[8] http://surpriselib.com/

5.1.6 GraphLab Create framework

Το GraphLab Create[9] είναι ένα framework που προορίζεται για ανάλυση μεγάλου όγκου δεδομένων και ανάπτυξη εφαρμογών μηχανικής μάθησης επαγγελματικού επιπέδου. Η ανάπτυξη γίνεται σε python και οι υλοποιήσεις αλγορίθμων μηχανικής μάθησης που παρέχει είναι βελτιστοποιημένες για υψηλή απόδοση. Το GraphLab Create δεν είναι ανοικτού κώδικα, αλλά παρέχεται δωρεάν για ακαδημαϊκή χρήση.

Για την παρούσα εργασία χρησιμοποιήθηκε ο αλγόριθμος παραγοντοποίησης πινάκων με μέθοδο ALS. Η επιλογή έγινε λόγω της ευχρηστίας και υψηλής ταχύτητας που παρέχει το συγκεκριμένο framework.

5.2 Λεπτομέρειες υλοποίησης

Στη συνέχεια παρουσιάζονται κάποιες αποφάσεις σχετικά με λεπτομέρειες υλοποίησης που αφορούν την απόδοση του κώδικα που αναπτύχθηκε.

5.2.1 Χρήση αραιών πινάκων (sparse matrices)

Αραιός πίνακας ονομάζεται ένας πίνακας δεδομένων του οποίου οι περισσότερες τιμές είναι άγνωστες. Συνήθως οι πίνακες αξιολογήσεων χρηστών που χρησιμοποιούνται στα συστήματα συστάσεων είναι αραιοί, καθώς κάθε χρήστης τυπικά αξιολογεί μόνο ένα μικρό ποσοστό των διαθέσιμων αντικειμένων του συστήματος.

Όταν χρειάζεται να χειριστούμε μεγάλους αραιούς πίνακες δεν είναι εύκολο να τους αποθηκεύσουμε στη μνήμη του υπολογιστή. Αυτό συμβαίνει γιατί οι άγνωστες τιμές πρέπει να συμβολιστούν με το ψηφίο μηδέν ή κάποιο άλλο αριθμό και έτσι αν αποθηκευτούν με τον συμβατικό τρόπο, απαιτούν τεράστια ποσά μνήμης. Για το λόγο αυτό χρησιμοποιούνται εξειδικευμένες δομές δεδομένων που ονομάζονται sparse matrices και ουσιαστικά αποθηκεύουν μόνο τις γνωστές τιμές δεδομένων. Υπάρχουν διάφορες μέθοδοι κατασκευής δομών αραιών πινάκων και γενικά χωρίζονται σε δύο κατηγορίες. Η πρώτη κατηγορία μεθόδων είναι αυτές που επιτρέπουν την εύκολη κατασκευή και ενημέρωση των δεδομένων όπως LIL (List of lists) και COO (Coordinate list). Στη δεύτερη κατηγορία ανήκουν μέθοδοι που επιτρέπουν την γρήγορες λειτουργίες προσπέλασης και πράξεων μεταξύ των πινάκων, όπως CSR (Compressed Sparse Row) και CSC (Compressed Sparse Column), όπου τα δεδομένα αποθηκεύονται κατά γραμμές και κατά στήλες αντίστοιχα (Buluç et al., 2009).

[9] https://turi.com/

Η επιλογή της κατάλληλης δομής αραιού πίνακα εξαρτάται από την εφαρμογή για την οποία προορίζεται, όμως ο αραιός πίνακας τύπου CSR είναι διαδεδομένος λόγω της υψηλής απόδοσης του.

5.2.2 Βελτιστοποίηση κώδικα με Cython

Για την ανάπτυξη του κώδικα των πειραμάτων, χρησιμοποιήθηκε η γλώσσα προγραμματισμού python, λόγω της απλότητας και ταχύτητας προτυποποίησης που παρέχει. Παρόλα αυτά αποδείχθηκε αρκετά αργή για επεξεργασία μεγάλου όγκου δεδομένων, οπότε έγινε μια προσπάθεια για επιλογή υλοποιήσεων αλγορίθμων σε αποδοτικότερες γλώσσες προγραμματισμού (π.χ. C++, C#).

Υπάρχουν τρόποι βελτιστοποίησης της απόδοσης του κώδικα python χρησιμοποιώντας διάφορα τεχνάσματα και τεχνικές, όμως μια από τις καλύτερες προσεγγίσεις είναι η χρήση του Cython[10]. Πρόκειται για ένα compiler που έχει την δυνατότητα να παράγει κώδικα C από κώδικα Python, απαιτώντας μικρές μόνο προσθήκες στον αρχικό κώδικα. Ο κώδικας C που παράγεται είναι εξαιρετικά αποδοτικός και μπορεί να μεταγλωττιστεί με έναν C/C++ compiler. Έτσι, αυτή η προσέγγιση είναι ιδανική για την επιτάχυνση τμημάτων κώδικα αλγορίθμων που επιτελούν επαναληπτικές διαδικασίες και πολύπλοκους υπολογισμούς.

Η βιβλιοθήκη Surprise python recommender system library επιλέχθηκε γιατί τα κρίσιμα από πλευράς απόδοσης τμήματα των αλγορίθμων που παρέχει, έχουν υλοποιηθεί σε Cython, καθιστώντας τους πολύ αποδοτικούς στην εκτέλεση τους.

5.3 Εγκατάσταση εργαλείων

5.3.1 Anaconda Python

Για την εγκατάσταση της διανομής Python Anaconda, αρκεί να κατεβάσουμε τον installer από την επίσημη ιστοσελίδα[11] της και να τον εκτελέσουμε. Ο installer θα εγκαταστήσει τα περισσότερα από τα βασικά πακέτα που χρειαζόμαστε, όπως numpy, pandas, matplotlib, scikit-learn κλπ. Για την εργασία χρησιμοποιήθηκε η Anaconda 4.2.0 (Python 2.7 64-bit).

[10] http://cython.org/

[11] https://www.continuum.io/downloads

5.3.2 Εγκατάσταση απαιτούμενων πακέτων python

Για την αναβάθμιση και εγκατάσταση των απαιτούμενων πακέτων python εκτελούμε τις παρακάτω εντολές σε μια κονσόλα "Anaconda Prompt":

```
conda update conda
conda update –all

pip install scikit-surprise
pip intall pythonnet
```

5.3.3 Εγκατάσταση βιβλιοθήκης MyMediaLite

Η βιβλιοθήκη MyMediaLite είναι γραμμένη σε C#.NET, οπότε για να λειτουργήσει πρέπει επίσης να είναι εγκατεστημένο το .Net Framework ή το Mono. Για χρήση της βιβλιοθήκης, αρκεί να κατεβάσουμε από την ιστοσελίδα την έκδοση 3.11 και να πάρουμε τα αρχεία MyMediaLite.ddl, MathNet.Numerics.dll και C5.dll. Η κλήση των μεθόδων της βιβλιοθήκης MyMediaLite μέσα από την γλώσσα Python γίνεται με τη βοήθεια του module pythonnet.

5.3.4 Εγκατάσταση του framework GraphLab Create

Το GraphLab Create παρέχεται δωρεάν για ακαδημαϊκή χρήση, οπότε το πρώτο βήμα είναι η απόκτηση της σχετικής άδειας μέσω της ιστοσελίδας της εταιρίας[12]. Στη συνέχεια πριν από την εγκατάσταση του framework, μέσω της διανομής Anaconda δημιουργήθηκε ένα εικονικό περιβάλλον python (python virtual environment) για λόγους συμβατότητας των πακέτων python, καθώς η έκδοση GraphLab Create 2.1 που χρησιμοποιήθηκε είναι πιστοποιημένη για εγκατάσταση σε Anaconda 4.0.0. Τέλος, η εγκατάσταση έγινε χρησιμοποιώντας το εργαλείο εγκατάστασης πακέτων pip και δίνοντας και το σχετικό κλειδί από το προηγούμενο βήμα. Οι απαιτούμενες εντολές σύμφωνα με τις οδηγίες της εταιρίας είναι οι παρακάτω:

```
# Create a new conda environment with Python 2.7.x
conda create -n gl-env python=2.7 anaconda=4.0.0

# Activate the conda environment
activate gl-env
```

[12] https://turi.com/download/academic.html

```
# Install your licensed copy of GraphLab Create
pip install --upgrade --no-cache-dir https://get.graphlab.com/GraphLab-
Create/2.1/your registered email address here/your product key here/GraphLab-
Create-License.tar.gz
```

5.3.5 Εγκατάσταση κώδικα διπλωματικής εργασίας

Ο κώδικας που αναπτύχθηκε για την διεξαγωγή των πειραμάτων δεν απαιτεί κάποια εγκατάσταση, καθώς πρόκειται για κώδικα python που μπορεί να εκτελεστεί άμεσα από την γραμμή εντολών.

6

Επίλογος

Στο κεφάλαιο αυτό θα γίνει μια σύνοψη της διπλωματικής εργασίας, θα παρουσιαστούν κάποια γενικά συμπεράσματα με βάση τα πειραματικά αποτελέσματα και τέλος θα αναφερθούν πιθανές μελλοντικές επεκτάσεις της διπλωματικής.

6.1 Σύνοψη και συμπεράσματα

Ο σκοπός της διπλωματικής εργασίας ήταν να γίνει μια συγκριτική μελέτη μεθόδων μηχανικής μάθησης για συστήματα συστάσεων και πιο συγκεκριμένα αλγορίθμων για Συνεργατικό Φιλτράρισμα.

Στο Κεφάλαιο 1 γίνεται μια εισαγωγή στα Συστήματα Συστάσεων και στις εφαρμογές για τις οποίες κατασκευάζονται. Στο Κεφάλαιο 2 γίνεται μια επισκόπηση βασικών εννοιών που αφορούν τα συστήματα συστάσεων, αναφέρονται οι διάφορες κατηγορίες συστημάτων και οι τρόποι λειτουργίας τους, καθώς και διάφορα προβλήματα και προκλήσεις που συναντώνται κατά την ανάπτυξη τους. Επίσης, γίνεται αναφορά σε μεθόδους μηχανικής μάθησης που χρησιμοποιούνται σε συστήματα συστάσεων, δίνοντας έμφαση στις τεχνικές μείωσης διαστατικότητας. Στο Κεφάλαιο 3 αναλύονται οι αλγόριθμοι που συμμετέχουν στην συγκριτική μελέτη. Στο Κεφάλαιο 4 παρουσιάζεται μια μεθοδολογία αξιολόγησης συστημάτων Συνεργατικού Φιλτραρίσματος και παρουσιάζονται τα αποτελέσματα των πειραμάτων. Η σύγκριση έγινε με βάση την μετρική RMSE, αξιολογώντας την διαδικασία πρόβλεψης αξιολογήσεων χρηστών σε αντικείμενα. Τέλος στο Κεφάλαιο 5 παρουσιάζονται

τα εργαλεία που χρησιμοποιήθηκαν και κάποιες τεχνικές λεπτομέρειες σχετικά με την απόδοση των αλγορίθμων.

Η αξιολόγηση των μεθόδων πρόβλεψης αξιολογήσεων ανέδειξε τα παρακάτω γενικά συμπεράσματα:

1. Οι τεχνικές βασισμένες σε μοντέλο έδειξαν γενικά μεγαλύτερη ακρίβεια προβλέψεων σε σχέση με τις τεχνικές βασισμένες στη μνήμη. Σε όλες τις συλλογές δεδομένων είχαν σταθερά μεγαλύτερη ακρίβεια στις προβλέψεις.
2. Στις τεχνικές βασισμένες στη μνήμη, οι προσεγγίσεις που χρησιμοποιούν ως μετρική ομοιότητας τον συντελεστή συσχέτισης του Pearson, παρουσίασαν σε όλες τις συλλογές δεδομένων μεγαλύτερη ακρίβεια από τις αντίστοιχες με μετρική ομοιότητας Συνημιτόνου.
3. Οι τεχνικές παραγοντοποίησης πινάκων αποτελούν την state-of-the-art προσέγγιση στο πρόβλημα της πρόβλεψης αξιολογήσεων, όμως παρουσιάζουν έντονα το φαινόμενο της υπερπροσαρμογής. Για το λόγο αυτό, απαιτούν προσεκτική επιλογή υπερπαραμέτρων και επιπλέον μέτρα κατά της υπερπροσαρμογής όπως για παράδειγμα την μέθοδο early stopping with validation.
4. Οι τεχνικές παραγοντοποίησης πινάκων παρουσιάζουν καλύτερη δυνατότητα κλιμάκωσης (scalability), σε σχέση με τις τεχνικές βασισμένες στη μνήμη, οι οποίες παρουσιάζουν προβλήματα όταν ο αριθμός χρηστών ή αντικειμένων είναι πολύ μεγάλος (συλλογές Jester-1 και Book Crossing).
5. Η χρήση biases βελτιώνει σε μεγάλο ποσοστό την ακρίβεια των προβλέψεων, τόσο των τεχνικών βασισμένων στη μνήμη, όσο και των τεχνικών βασισμένων σε μοντέλα. Η βελτίωση φάνηκε σε όλες τις συλλογές δεδομένων, με μόνη εξαίρεση τις τεχνικές παραγοντοποίησης πινάκων στην συλλογή Jester-1.
6. Οι τεχνικές παραγοντοποίησης πινάκων λειτουργούν καλύτερα από τις τεχνικές βασισμένες στη μνήμη, όταν τα δεδομένα είναι πολύ αραιά, όπως στην περίπτωση της συλλογής δεδομένων Book crossing.
7. Η χρήση έμμεσης πληροφορίας σε συνδυασμό με τα δεδομένα αξιολογήσεων, δείχνει να βελτιώνει την ακρίβεια των προβλέψεων σε όλες τις συλλογές, εκτός από την Jester-1, έχει όμως πολύ μεγάλο πρόσθετο υπολογιστικό κόστος, που δεν είναι σε καμία περίπτωση ανάλογο της βελτίωσης που προσφέρει.

6.2 Μελλοντικές επεκτάσεις

Μια ενδιαφέρουσα επέκταση της συγκριτικής μελέτης θα ήταν η σύγκριση επιπλέον αλγορίθμων συνεργατικού φιλτραρίσματος με βάση μοντέλα, όπως μεθόδους

παραγοντοποίησης πινάκων (π.χ. PMF, NMF) και μεθόδους βασισμένες σε μοντέλα που χρησιμοποιούν νευρωνικά δίκτυα, όπως MLP, Restricted Boltzmann Machines και Auto-Encoders. Επίσης, να αξιολογηθούν αλγόριθμοι που έχουν την δυνατότητα να αξιοποιήσουν εκτός από τα δεδομένα αξιολογήσεων και πρόσθετες πληροφορίες, όπως χρονικό προσδιορισμό των αξιολογήσεων και χαρακτηριστικά χρηστών και αντικειμένων. Ενδεικτικά τέτοιοι αλγόριθμοι είναι ο timeSVD++ (Koren and Bell, 2015), οι Μηχανές Παραγοντοποίησης (Factorization Machines)(Rendle, 2010) και ο SVDFeature (Chen et al., 2012).

Μια παράλληλη σύγκριση με βάση την απόδοση σε top-N recommendation χρησιμοποιώντας κατάλληλες μετρικές κατάταξης συστάσεων (recommendation ranking). Στην σύγκριση αυτή είναι ενδιαφέρον να αξιολογηθούν τόσο οι ήδη αξιολογηθέντες αλγόριθμοι, μετατρέποντας τις προβλέψεις αξιολογήσεων σε λίστες N καλύτερων αντικειμένων, όσο και αλγόριθμοι φτιαγμένοι ειδικά για αυτή την εργασία όπως Sparse LInear Methods (SLIM), Bayesian Personalized Ranking (BPR) κλπ.

Όσον αφορά την υπάρχουσα συγκριτική μελέτη, θα μπορούσε να βελτιωθεί ο τρόπος εύρεσης των βέλτιστων υπερπαραμέτρων των αλγορίθμων, καθώς η μέθοδος grid search που χρησιμοποιήθηκε είναι η απλούστερη προσέγγιση και δεν εγγυάται ότι θα βρει πολύ καλές υπερπαραμέτρους σε εύλογο χρονικό διάστημα. Άλλες μέθοδοι που μπορούν να δοκιμαστούν είναι η randomized grid search και μέθοδοι μπεϋζιανής βελτιστοποίησης. Αυτές είναι πιθανόν να βρουν καλύτερες τιμές υπερπαραμέτρων σε μικρότερο χρονικό διάστημα σε σχέση με την μέθοδο grid search.

Τέλος, μια ακόμα ενδιαφέρουσα επέκταση είναι η δοκιμή αλγορίθμων Συνεργατικού Φιλτραρίσματος σε μεγαλύτερες συλλογές δεδομένων, με σκοπό τον έλεγχο της δυνατότητας κλιμάκωσης (scalability). Η σύγκριση αυτή μπορεί να συνδυαστεί και με παράλληλες υλοποιήσεις αλγορίθμων με χρήση CPU (multi-core) (Chin et al., 2015), αλλά και GPU (π.χ. με την τεχνολογία CUDA της nVidia) (Tan et al., 2016).

7

Βιβλιογραφία

Abdi, H., Williams, L.J., 2010. Principal component analysis. Wiley Interdiscip. Rev. Comput. Stat. 2, 433–459.

Adomavicius, G., Tuzhilin, A., 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Trans. Knowl. Data Eng. 17, 734–749.

Amatriain, X., Pujol, J.M., 2015. Data Mining Methods for Recommender Systems, in: Recommender Systems Handbook. Springer, pp. 227–262.

Anaconda Python [WWW Document], n.d. URL https://www.continuum.io/ (accessed 8.27.16).

Anil, R., Owen, S., Dunning, T., Friedman, E., 2010. Mahout in Action. Manning Publications Co. Sound View Ct. #3B Greenwich, CT 06830.

Bawden, D., Robinson, L., 2008. The dark side of information: overload, anxiety and other paradoxes and pathologies. J. Inf. Sci. 35, 180–191. doi:10.1177/0165551508095781

Bell, R.M., Koren, Y., 2007. Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights. IEEE, pp. 43–52. doi:10.1109/ICDM.2007.90

Bell, R.M., Koren, Y., Volinsky, C., 2008. The bellkor 2008 solution to the netflix prize. Stat. Res. Dep. ATT Res.

Bell, R.M., Koren, Y., Volinsky, C., 2007. The BellKor solution to the Netflix prize.

Bennett, J., Lanning, S., 2007. The netflix prize, in: Proceedings of KDD Cup and Workshop. p. 35.

Bergstra, J., Bengio, Y., 2012. Random search for hyper-parameter optimization. J. Mach. Learn. Res. 13, 281–305.

Bergstra, J.S., Bardenet, R., Bengio, Y., Kégl, B., 2011. Algorithms for hyper-parameter optimization, in: Advances in Neural Information Processing Systems. pp. 2546–2554.

Blei, D.M., Ng, A.Y., Jordan, M.I., 2003. Latent dirichlet allocation. J. Mach. Learn. Res. 3, 993–1022.

Braida, F., Mello, C.E., Pasinato, M.B., Zimbrão, G., 2015. Transforming collaborative filtering into supervised learning. Expert Syst. Appl. 42, 4733–4742. doi:10.1016/j.eswa.2015.01.023

Breiman, L., 2001. Random Forests. Mach. Learn. 45, 5–32. doi:10.1023/A:1010933404324

Buluç, A., Fineman, J.T., Frigo, M., Gilbert, J.R., Leiserson, C.E., 2009. Parallel sparse matrix-vector and matrix-transpose-vector multiplication using compressed sparse blocks, in: Proceedings of the Twenty-First Annual Symposium on Parallelism in Algorithms and Architectures. ACM, pp. 233–244.

Burke, R., 2007. Hybrid web recommender systems, in: The Adaptive Web. Springer, pp. 377–408.

Cacheda, F., Carneiro, V., Fernández, D., Formoso, V., 2011. Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems. ACM Trans. Web 5, 1–33. doi:10.1145/1921591.1921593

Calderón-Benavides, M.L., González-Caro, C.N., Pérez-Alcázar, J. de J., García-Díaz, J.C., Delgado, J., 2004. A comparison of several predictive algorithms for collaborative filtering on multi-valued ratings, in: Proceedings of the 2004 ACM Symposium on Applied Computing. ACM, pp. 1033–1039.

Cantador, I., Bellogín, A., Castells, P., 2008. A Multilayer Ontology-based Hybrid Recommendation Model. AI Commun 21, 203–210.

Chen, T., Zhang, W., Lu, Q., Chen, K., Zheng, Z., Yu, Y., 2012. SVDFeature: a toolkit for feature-based collaborative filtering. J. Mach. Learn. Res. 13, 3619–3622.

Chin, W.-S., Zhuang, Y., Juan, Y.-C., Lin, C.-J., 2015. A Fast Parallel Stochastic Gradient Method for Matrix Factorization in Shared Memory Systems. ACM Trans. Intell. Syst. Technol. TIST 6, 2.

Christakou, C., Vrettos, S., Stafylopatis, A., 2007. A hybrid movie recommender system based on neural networks. Int. J. Artif. Intell. Tools 16, 771–792. doi:http://dx.doi.org/10.1142/S0218213007003540

Cortes, C., Vapnik, V., 1995. Support-vector networks. Mach. Learn. 20, 273–297.

Davidson, J., Liebald, B., Liu, J., Nandy, P., Van Vleet, T., Gargi, U., Gupta, S., He, Y., Lambert, M., Livingston, B., others, 2010. The YouTube video recommendation system, in: Proceedings of the Fourth ACM Conference on Recommender Systems. ACM, pp. 293–296.

Debnath, S., Ganguly, N., Mitra, P., 2008. Feature weighting in content based recommendation system using social network analysis, in: Proceedings of the 17th International Conference on World Wide Web. ACM, pp. 1041–1042.

Domingos, P., 2012. A few useful things to know about machine learning. Commun. ACM 55, 78–87.

Dror, G., Koenigstein, N., Koren, Y., Weimer, M., 2012. The Yahoo! Music Dataset and KDD-Cup'11., in: KDD Cup. pp. 8–18.

Ekstrand, M.D., 2011. Collaborative Filtering Recommender Systems. Found. Trends® Human–Computer Interact. 4, 81–173. doi:10.1561/1100000009

Ekstrand, M.D., Ludwig, M., Konstan, J.A., Riedl, J.T., 2011. Rethinking the recommender research ecosystem: reproducibility, openness, and LensKit, in: Proceedings of the Fifth ACM Conference on Recommender Systems. ACM, pp. 133–140.

Fisher, D., Hildrum, K., Hong, J., Newman, M., Thomas, M., Vuduc, R., 2000. SWAMI (poster session): a framework for collaborative filtering algorithm development and evaluation, in: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, pp. 366–368.

Gantner, Z., Rendle, S., Freudenthaler, C., Schmidt-Thieme, L., 2011. MyMediaLite: A free recommender system library, in: Proceedings of the Fifth ACM Conference on Recommender Systems. ACM, pp. 305–308.

Goldberg, D., Nichols, D., Oki, B.M., Terry, D., 1992. Using collaborative filtering to weave an information tapestry. Commun. ACM 35, 61–70.

Goldberg, K., Roeder, T., Gupta, D., Perkins, C., 2001. Eigentaste: A constant time collaborative filtering algorithm. Inf. Retr. 4, 133–151.

Golub, G.H., Reinsch, C., 1970. Singular value decomposition and least squares solutions. Numer. Math. 14, 403–420.

González-Caro, C.N., Calderón-Benavides, M.L., Pérez-Alcázar, J. de J., García-Díaz, J.C., Delgado, J., 2002. Towards a More Comprehensive Comparison of Collaborative Filtering Algorithms, in: Laender, A.H.F., Oliveira, A.L. (Eds.), String Processing and Information Retrieval: 9th International Symposium, SPIRE 2002 Lisbon, Portugal, September 11–13, 2002 Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 248–253.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press.

Gorrell, G., Webb, B., 2005. Generalized hebbian algorithm for incremental latent semantic analysis., in: INTERSPEECH. pp. 1325–1328.

Gunawardana, A., Shani, G., 2015. Evaluating Recommender Systems, in: Ricci, F., Rokach, L., Shapira, B. (Eds.), Recommender Systems Handbook. Springer US, Boston, MA, pp. 265–308.

Harper, F.M., Konstan, J.A., 2016. The movielens datasets: History and context. ACM Trans. Interact. Intell. Syst. TiiS 5, 19.

Herlocker, J.L., Konstan, J.A., Borchers, A., Riedl, J., 1999. An Algorithmic Framework for Performing Collaborative Filtering, in: Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '99. ACM, New York, NY, USA, pp. 230–237. doi:10.1145/312624.312682

Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T., 2004. Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst. TOIS 22, 5–53.

Hu, Y., Koren, Y., Volinsky, C., 2008. Collaborative Filtering for Implicit Feedback Datasets. IEEE, pp. 263–272. doi:10.1109/ICDM.2008.22

Huang, Z., Zeng, D., Chen, H., 2007. A comparison of collaborative-filtering recommendation algorithms for e-commerce. IEEE Intell. Syst. 22.

Kim, T.-H., Yang, S.-B., 2007. An Effective Threshold-Based Neighbor Selection in Collaborative Filtering, in: Amati, G., Carpineto, C., Romano, G. (Eds.), Advances in Information Retrieval: 29th European Conference on IR Research, ECIR 2007, Rome, Italy, April 2-5, 2007. Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 712–715.

Kohavi, R., others, 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection, in: Ijcai. pp. 1137–1145.

Koren, Y., 2009. The bellkor solution to the netflix grand prize. Netflix Prize Doc. 81.

Koren, Y., 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp. 426–434.

Koren, Y., Bell, R., 2015. Advances in collaborative filtering, in: Recommender Systems Handbook. Springer, pp. 77–118.

Koren, Y., Bell, R., Volinsky, C., 2009. Matrix factorization techniques for recommender systems. Computer 30–37.

Lee, J., Sun, M., Lebanon, G., 2012. A comparative study of collaborative filtering algorithms. ArXiv Prepr. ArXiv12053193.

Maclaurin, D., Duvenaud, D.K., Adams, R.P., 2015. Gradient-based Hyperparameter Optimization through Reversible Learning., in: ICML. pp. 2113–2122.

Mitchell, T.M., 1997. Machine learning. McGraw Hill.

Mnih, A., Salakhutdinov, R., 2007. Probabilistic matrix factorization, in: Advances in Neural Information Processing Systems. pp. 1257–1264.

Movielens [WWW Document], n.d. URL https://movielens.org (accessed 8.31.16).

Narayanan, A., Shmatikov, V., 2007. How to break anonymity of the netflix prize dataset. ArXiv Prepr. Cs0610105.

Paterek, A., 2007. Improving regularized singular value decomposition for collaborative filtering, in: Proceedings of KDD Cup and Workshop. pp. 5–8.

Pazzani, M.J., 1999. A framework for collaborative, content-based and demographic filtering. Artif. Intell. Rev. 13, 393–408.

Pilászy, I., 2010. Factorization-based large scale recommendation algorithms. Citeseer.

Piotte, M., Chabbert, M., 2009. The pragmatic theory solution to the netflix grand prize. Netflix Prize Doc.

Prechelt, L., 1998. Early Stopping - But When?, in: Orr, G.B., Müller, K.-R. (Eds.), Neural Networks: Tricks of the Trade. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 55–69.

Quinlan, J.R., 1993. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Quinlan, J.R., 1986. Induction of Decision Trees. Mach. Learn. 1, 81–106. doi:10.1023/A:1022643204877

Rendle, S., 2010. Factorization machines, in: Data Mining (ICDM), 2010 IEEE 10th International Conference on. IEEE, pp. 995–1000.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J., 1994. GroupLens: an open architecture for collaborative filtering of netnews, in: Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work. ACM, pp. 175–186.

Ricci, F., Rokach, L., Shapira, B., 2015. Recommender Systems: Introduction and Challenges, in: Ricci, F., Rokach, L., Shapira, B. (Eds.), Recommender Systems Handbook. Springer US, Boston, MA, pp. 1–34.

Rosenblatt, F., 1958. The perceptron: a probabilistic model for information storage and organization in the brain. Psychol. Rev. 65, 386—408.

Said, A., Bellogín, A., 2014. Comparative recommender system evaluation: benchmarking recommendation frameworks. ACM Press, pp. 129–136. doi:10.1145/2645710.2645746

Salakhutdinov, R., Mnih, A., Hinton, G., 2007. Restricted Boltzmann machines for collaborative filtering, in: Proceedings of the 24th International Conference on Machine Learning. ACM, pp. 791–798.

Sarwar, B., Karypis, G., Konstan, J., Riedl, J., 2001. Item-based collaborative filtering recommendation algorithms, in: Proceedings of the 10th International Conference on World Wide Web. ACM, pp. 285–295.

Sarwar, B., Karypis, G., Konstan, J., Riedl, J., 2000. Application of dimensionality reduction in recommender system-a case study. DTIC Document.

Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S., 2007. Collaborative filtering recommender systems, in: The Adaptive Web. Springer, pp. 291–324.

Schafer, J.B., Konstan, J., Riedl, J., 1999. Recommender systems in e-commerce, in: Proceedings of the 1st ACM Conference on Electronic Commerce. ACM, pp. 158–166.

Snoek, J., Larochelle, H., Adams, R.P., 2012. Practical bayesian optimization of machine learning algorithms, in: Advances in Neural Information Processing Systems. pp. 2951–2959.

Su, X., Khoshgoftaar, T.M., 2009. A Survey of Collaborative Filtering Techniques. Adv. Artif. Intell. 2009, 1–19. doi:10.1155/2009/421425

Swaroop, C., 2003. A Byte of Python [WWW Document]. URL http://python.swaroopch.com/ (accessed 8.27.16).

Takács, G., Pilászy, I., Németh, B., Tikk, D., 2009. Scalable collaborative filtering approaches for large recommender systems. J. Mach. Learn. Res. 10, 623–656.

Tan, W., Cao, L., Fong, L., 2016. Faster and Cheaper: Parallelizing Large-Scale Matrix Factorization on GPUs. ArXiv Prepr. ArXiv160303820.

Töscher, A., Jahrer, M., 2008. The bigchaos solution to the netflix prize 2008. Netflix Prize Rep.

Töscher, A., Jahrer, M., Bell, R.M., 2009. The bigchaos solution to the netflix grand prize. Netflix Prize Doc.

Van der Maaten, L., Hinton, G., 2008. Visualizing data using t-SNE. J. Mach. Learn. Res. 9, 85.

Van Der Maaten, L., Postma, E., Van den Herik, J., 2009. Dimensionality reduction: a comparative review. J Mach Learn Res 10, 66–71.

Webb, B., 2006. Netflix Update: Try This at Home [WWW Document]. URL http://sifter.org/~simon/journal/20061211.html (accessed 1.14.17).

Zhou, Y., Wilkinson, D., Schreiber, R., Pan, R., 2008. Large-scale parallel collaborative filtering for the netflix prize, in: Algorithmic Aspects in Information and Management. Springer, pp. 337–348.

Ziegler, C.-N., McNee, S.M., Konstan, J.A., Lausen, G., 2005. Improving recommendation lists through topic diversification, in: Proceedings of the 14th International Conference on World Wide Web. ACM, pp. 22–32.