5. Αλγόριθμοι Ανάλυσης και Παραγοντοποίησης Πίνακα

***Σύνοψη**: Στο πέμπτο κεφάλαιο θα μελετήσουμε τους βασικούς **αλγορίθμους ανάλυσης και παραγοντοποίησης πινάκων** (**matrix factorization**). Θα ορίσουμε το πρόβλημα της σύστασης στοιχείων σε χρήστες ως έναν δισδιάστατο πίνακα που θα κρατά τις βαθμολογίες που δίνουν οι δεύτεροι στα πρώτα. Επίσης, θα ορίσουμε μια **αντικειμενική συνάρτηση** για την ελαχιστοποίηση της διαφοράς μεταξύ της πραγματικής και της προβλεπόμενης βαθμολογίας ενός χρήστη για ένα στοιχείο. Στη συνέχεια, θα περιγράψουμε την **Αποσύνθεση Πίνακα βάσει Ιδιαζουσών Τιμών** (**Singular Value Decomposition**) και την **Ανάλυση Πίνακα βάσει Εναλλασσόμενων Ελαχίστων Τετραγώνων** (**Alternating Least Squares**). Θα ακολουθήσει ο ορισμός **πινάκων υψηλότερης τάξης (τενσόρων)**, όπου συμμετέχουν περισσότερες οντότητες (χρήστες, στοιχεία, κατηγορίες στοιχείων κ.λπ.), καθώς και η περιγραφή **αλγορίθμων ανάλυσης και παραγοντοποίησης τενσόρων**. Τέλος, θα γίνει ορισμός του προβλήματος ως **pairwise ranking** (**κατάταξη στοιχείων ανά ζεύγος**), που σημαίνει ότι θα επιχειρήσουμε να προβλέψουμε τη σωστή κατάταξη των στοιχείων στη λίστα συστάσεων που αφορά σ' έναν χρήστη. Πιο συγκεκριμένα, θα ορίσουμε μία νέα αντικειμενική συνάρτηση στόχος της οποίας θα είναι να προτείνουμε πάντα σε καλύτερη κατάταξη ένα στοιχείο με το οποίο έχει αλληλεπιδράσει ο χρήστης στο παρελθόν συγκριτικά με κάποιο άλλο στοιχείο με το οποίο δεν υπήρξε διάδραση.*

***Προαπαιτούμενη γνώση**: Ο αναγνώστης καλό θα ήταν να έχει προηγουμένως μελετήσει το δεύτερο κεφάλαιο.*

5.1 Εισαγωγή

Στο παρόν κεφάλαιο περιγράφουμε τις βασικές μεθόδους *ανάλυσης πινάκων* για τα συστήματα συστάσεων. Είναι αλήθεια άλλωστε ότι πολλά προβλήματα στη ζωή μας αναπαρίστανται με πίνακες. Όμως λόγω της υψηλής *διαστατικότητας (dimensionality)* -άρα, πολλοί χρήστες και πολλά στοιχεία- που χαρακτηρίζει τα δεδομένα των προβλημάτων ο αρχικός πίνακας περιέχει πολλές

γραμμές και στήλες και γι αυτόν το λόγο συνήθως παραγοντοποιείται σε δύο ή περισσότερους μικρότερους ή καλύτερα "λεπτότερους" πίνακες. Αυτοί οι πίνακες έχουν το πλεονέκτημα των μικρότερων διαστάσεων με αποτέλεσμα να μειώνεται έτσι ο απαιτούμενος αποθηκευτικός χώρος και να χρειάζεται λιγότερος χρόνος για την επεξεργασία τους. Ως εκ τούτου είναι δυνατός ο αποτελεσματικότερος χειρισμός τους από αλγορίθμους συγκριτικά με τον αρχικό πίνακα. Υπάρχουν, λοιπόν, πολλές μέθοδοι [Rajaraman and Ullman, 2011] για την *αποσύνθεση ενός πίνακα* και την *επεξεργασία* ενός συνόλου *δεδομένων υψηλής διαστατικότητας*. Ακολουθούν οι κυριώτεροι.

Η *Ανάλυση Κύριων Συνιστωσών* (*Principle Component Analysis - PCA*) είναι μια τεχνική *Εξόρυξης Δεδομένων* (*Data Mining*) που αναπαριστά τα αρχικά *δεδομένα υψηλής διαστατικότητας* με την ταυτόχρονη προβολή όμως των σημαντικότερων τάσεων που ενυπάρχουν σ' αυτά. Πρόκειται για μια απλή μέθοδο η οποία είναι γνωστή ως *αποσύνθεση ιδιοτιμών (Eigenvalue Decomposition)* ([Bensmail and Celeux, 1996], [Golub and Van Loan, 1996], [Moler and Stewart, 1973], [Quarteroni et al., 2000]) και βασίζεται στις *ιδιοτιμές* και τα *ιδιοδιανύσματα* ενός πίνακα, έννοιες που θα αναλυθούν στην επόμενη ενότητα.

Μια άλλη μέθοδος που προέρχεται από το επιστημονικό πεδίο της *Ανάκτησης Πληροφορίας* (*Information Retrieval*) είναι η *Latent Semantic Indexing (LSI)*, η οποία εφαρμόζει την *Αποσύνθεση Ιδιαζουσών Τιμών (Singular Value Decomposition - SVD)*, που με τη σειρά της χρησιμοποιεί τις *ιδιάζουσες* τιμές του αρχικού πίνακα για να τον *παραγοντοποιήσει* ([Berry et al., 1994], [Brand, 2002], [Cao, 2006], [Chin et al., 2006], [Lathauwer et al., 2000], [Furnas et al., 1988], [Sarwar et al., 2002]). Η Ενότητα αυτή ολοκληρώνεται με την περιγραφή και άλλων σύγχρονων *τεχνικών αποσύνθεσης πινάκων*, όπως η *Αποσύνθεση κατά UV* (*UV-decomposition*), η *Μέθοδος Εναλλασσόμενων Ελαχίστων Τετραγώνων* (*wALS*), η *Αποσύνθεση Τανυστή* (*Tensor decomposition*), καθώς και οι μέθοδοι *Bayesian Pairwise Ranking* και *Listwise Ranking* (*Κατάταξη στοιχείων ανά λίστα*), εκ των οποίων οι δύο τελευταίες βασίζονται στον καθορισμό μιας *αντικειμενικής συνάρτησης* που βελτιστοποιεί την κατάταξη των προτεινόμενων στοιχείων της λίστας συστάσεων.

5.2 Αποσύνθεση ιδιοτιμών (Eigenvalue Decomposition)

Σ' αυτήν την ενότητα θα παρουσιάσουμε μια μέθοδο αποσύνθεσης γνωστή ως *Αποσύνθεση Ιδιοτιμών* (*Eigenvalue Decomposition*). Αναλυτικότερα, στη γραμμική Άλγεβρα η παραπάνω μέθοδος είναι στην ουσία η παραγοντοποίηση ενός πίνακα A: οι *ιδιοτιμές* και τα *ιδιοδιανύσματα* του A χρησιμοποιούνται για την αναπαράστασή του. Για να εφαρμοστεί αυτή η μέθοδος σ' έναν πίνακα, όμως, αυτός πρέπει να είναι όχι μόνο τετραγωνικός αλλά και "διαγωνοποιήσιμος".

Έστω, λοιπόν, A ένας τετραγωνικός και διαγωνοποιήσιμος πίνακας, λ μια σταθερά και $\mathbf{e}$ ένα μη μηδενικό διάνυσμα στήλης με τον ίδιο αριθμό γραμμών μ' αυτές του πίνακα A. Τότε, η λ είναι μια *ιδιοτιμή* του A και $\mathbf{e}$ είναι το αντίστοιχο *ιδιοδιάνυσμα* του A εάν:

$$A \cdot \mathbf{e} = \lambda \cdot \mathbf{e} \tag{5.1}$$

Για έναν πίνακα A, ο οποίος είναι τάξης r, μπορούμε να ομαδοποιήσουμε τις r μη μηδενικές *ιδιοτιμές* του σ' έναν $r \times r$ διαγώνιο πίνακα Λ και τα *ιδιοδιανύσματά* του σ' έναν $n \times r$ πίνακα E. Έτσι έχουμε:

$$A \cdot E = \Lambda \cdot E \tag{5.2}$$

Επιπλέον, στην περίπτωση που η τάξη r του πίνακα A είναι ίση με τη διάσταση του n, τότε ο πίνακας A μπορεί να παραγοντοποιηθεί ως εξής:

$$A = E \cdot \Lambda \cdot E^{-1} \tag{5.3}$$

η οποία είναι μια "διαγωνοποίηση" πίνακα παρόμοια με τη *μέθοδο αποσύνθεσης SVD*, και θα περιγραφεί αναλυτικά στην επόμενη ενότητα.

Παράδειγμα 5.1 Σ' αυτό το παράδειγμα, βασιζόμενοι στην Εξίσωση 5.3, θα εκτελέσουμε την ανάλυση *ιδιοτιμών* στον ακόλουθο 2×2 πίνακα A:

$$A = \begin{bmatrix} 1 & 0 \\ 1 & 3 \end{bmatrix} \tag{5.4}$$

Το ζητούμενο είναι να *αποσυνθέσουμε* τον πίνακα A σ' έναν διαγώνιο πίνακα μέσω του πολλαπλασιασμού του μ' έναν *μη-ιδιάζοντα (non singular)* πίνακα B:

$$B = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \in \mathbb{R}^{2\times 2} \tag{5.5}$$

Λαμβάνοντας, τώρα, υπόψη τη θεωρία των *ιδιοτιμών* και των *ιδιοδιανυσμάτων* ενός πίνακα και την Εξίσωση 5.3, παίρνουμε:

$$\begin{gathered} A = B \cdot \Lambda \cdot B^{-1} \iff \\ A \cdot B = B \iff \\ \begin{bmatrix} 1 & 0 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \end{gathered} \tag{5.6}$$

Χρησιμοποιώντας τους κανόνες της γραμμικής Άλγεβρας, η παραπάνω Εξίσωση μπορεί να γραφεί ως εξής:

$$\left.\begin{gathered} \begin{bmatrix} 1 & 0 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} a \\ c \end{bmatrix} = \begin{bmatrix} a\lambda_1 \\ c\lambda_1 \end{bmatrix} \\ \begin{bmatrix} 1 & 0 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} b \\ d \end{bmatrix} = \begin{bmatrix} b\lambda_2 \\ d\lambda_2 \end{bmatrix} \end{gathered}\right\}$$
$$\iff$$

$$\left.\begin{bmatrix}1 & 0\\ 1 & 3\end{bmatrix}\begin{bmatrix}a\\ c\end{bmatrix} = \lambda_1\begin{bmatrix}a\\ c\end{bmatrix}\right\}$$
$$\left.\begin{bmatrix}1 & 0\\ 1 & 3\end{bmatrix}\begin{bmatrix}b\\ d\end{bmatrix} = \begin{bmatrix}\lambda_2\end{bmatrix}\begin{bmatrix}b\\ d\end{bmatrix}\right\} \tag{5.7}$$

Το επόμενο βήμα είναι να αφήσουμε $\mathbf{a} = \begin{bmatrix}a\\ c\end{bmatrix}$ και $\mathbf{b} = \begin{bmatrix}b\\ d\end{bmatrix}$, γεγονός που μας δίνει δύο διανυσματικές εξισώσεις:

$$\left.\begin{aligned} A\cdot\mathbf{a} &= \lambda_1\cdot\mathbf{a}\\ A\cdot\mathbf{b} &= \lambda_2\cdot\mathbf{b}\end{aligned}\right\} \tag{5.8}$$

Η Εξίσωση 5.8 μπορεί να περιγραφεί με μια ενιαία διανυσματική εξίσωση που θα περιλαμβάνει τις δύο *ιδιοτιμές* ως λύσεις της. Βασιζόμενοι λοιπόν στην Εξίσωση 5.1, θέτουμε το λ να αντιπροσωπεύει τις δύο *ιδιοτιμές* λ_1 και λ_2, και το $\mathbf{e}$ να αντιπροσωπεύει τα δύο *ιδιοδιανύσματα* $\mathbf{a}$ και $\mathbf{b}$. Μετατοπίζοντας τώρα το $\lambda\cdot\mathbf{e}$ στην αριστερή πλευρά της Εξίσωσης, έχουμε:

$$A\cdot\mathbf{e} - \lambda\cdot\mathbf{e} = 0 \iff$$
$$(A - \lambda I)\cdot\mathbf{e} = 0 \iff$$
$$\xrightarrow{B\text{:μη ιδιάζον} \iff \mathbf{e}\text{:μη μηδενικός}}$$
$$(A - \lambda I) = 0 \iff$$

$$\begin{vmatrix}1-\lambda & 0\\ 1 & 3-\lambda\end{vmatrix} = 0 \iff$$

$$(1-\lambda)(3-\lambda) = 0$$

Έτσι, είμαστε έτοιμοι να υπολογίσουμε τις λύσεις και να πάρουμε τις *ιδιοτιμές* του πίνακα A, και οι οποίες είναι $\lambda_1 = 1$ και $\lambda_2 = 3$. Ο πίνακας που προκύπτει από την *Αποσύνθεση Ιδιοτιμών* του A έχει ως εξής:

$$\lambda = \begin{bmatrix}1 & 0\\ 0 & 3\end{bmatrix}$$

Τοποθετώντας τις λύσεις πίσω στην Εξίσωση 5.7, έχουμε:

$$\left.\begin{bmatrix}1 & 0\\ 1 & 3\end{bmatrix}\begin{bmatrix}a\\ c\end{bmatrix} = 1\begin{bmatrix}a\\ c\end{bmatrix}\right\}$$
$$\left.\begin{bmatrix}1 & 0\\ 1 & 3\end{bmatrix}\begin{bmatrix}b\\ d\end{bmatrix} = 3\begin{bmatrix}b\\ d\end{bmatrix}\right\} \tag{5.9}$$

Λύνοντας τις εξισώσεις, έχουμε:

$$a = -2c, \quad a \in \mathbb{R} \tag{5.10}$$

$$\wedge$$

$$b = 0, \quad d \in \mathbb{R} \tag{5.11}$$

Τέλος, ο πίνακας B είναι $\begin{bmatrix} -2c & 0 \\ c & d \end{bmatrix}$, και άρα η λύση στο ζητούμενο παράδειγμα είναι:

$$\begin{bmatrix} 1 & 0 \\ 1 & 3 \end{bmatrix} = \begin{bmatrix} -2c & 0 \\ c & d \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} -2c & 0 \\ c & d \end{bmatrix}^{-1}, \quad c, d \in \mathbb{R}$$

5.3 Ανάλυση Ιδιαζουσών Τιμών (Singular Value Decomposition)

Στο παρόν θα περιγράψουμε τη μέθοδο *Ανάλυσης Ιδιαζουσών Τιμών (SVD)* για συστήματα συστάσεων. Η *Ανάλυση Ιδιαζουσών Τιμών* θα εφαρμοστεί επί του πίνακα βαθμολογίας χρήστη-στοιχείου A. Από τον πίνακα A επομένως επιλέγουμε έναν c αριθμό *πρωτευόντων συνιστωσών (διαστάσεων)* με σκοπό να αποκαλύψουμε τις μείζονες τάσεις των χρηστών. Το μέγεθος της τιμής της διάστασης c καθορίζεται από το ποσοστό πληροφορίας που θέλουμε να διατηρηθεί σε σύγκριση με τον αρχικό πίνακα. Δημιουργείται, άρα, ένας c-διάστατος χώρος όπου κάθε διάσταση c_i αντιστοιχεί σε μια διακριτή τάση των χρηστών σχετικά με τη βαθμολογία επί των στοιχείων/προϊόντων. Στη συνέχεια, βάσει των αξιολογήσεων του υπό εξέταση χρήστη u, εισάγουμε το διάνυσμά του στον c-διάστατο χώρο. Τέλος, βρίσκουμε τους k *κοντινότερους γείτονες* του διανύσματος του u στον c-διάστατο χώρο και εφαρμόζουμε τον *αλγόριθμο φιλτραρίσματος βάσει ομοιότητας χρήστη* είτε *βάσει ομοιότητας στοιχείου*, για να δημιουργήσουμε τη λίστα με τα προτεινόμενα στοιχεία.

Παράδειγμα 5.2 Για να καταστεί ευκολότερη η περιγραφή της παραγοντοποίησης ενός πίνακα μέσω της *μεθόδου SVD*, θα χρησιμοποιήσουμε ως τρέχον παράδειγμα τα δεδομένα του Σχήματος 5.1, όπου I_1-I_4 είναι τα στοιχεία, ενώ U_1-U_4 είναι οι χρήστες.

	I_1	I_2	I_3	I_4
U_1	4	1	1	4
U_2	1	4	2	0
U_3	2	1	4	5

(α)

	I_1	I_2	I_3	I_4
U_4	1	4	1	0

(β)

Σχήμα 5.1: (α) Σύνολο *δεδομένων εκπαίδευσης* ($n \times m$), (β) Σύνολο *δεδομένων ελέγχου*.

Πιο συγκεκριμένα, τα δεδομένα του παραδείγματος (Σχήμα 5.1) χωρίζονται σε σύνολα *δεδομένων εκπαίδευσης* και *δεδομένων ελέγχου*. Τα κενά κελιά (χωρίς βαθμολογία) εμφανίζονται με την τιμή μηδέν.

5.3.1 Εφαρμογή SVD σε Δεδομένα Εκπαίδευσης

Αρχικά, εφαρμόζουμε τη *μέθοδο SVD* στα *δεδομένα εκπαίδευσης* του $n \times m$ πίνακα βαθολογίας χρηστών-στοιχείων A, με αποτέλεσμα να παράγονται τρεις πίνακες, οι οποίοι, πολλαπλασιαζόμενοι στη συνέχεια, δημιουργούν μια προσέγγιση του αρχικού πίνακα όπως φαίνεται στην παρακάτω Εξίσωση 5.12 καθώς και στο Σχήμα 5.2:

$$A_{n\times m} = U_{n\times n} \cdot S_{n\times m} \cdot V'_{m\times m} \tag{5.12}$$

$$\begin{bmatrix} 4 & 1 & 1 & 4 \\ 1 & 4 & 2 & 0 \\ 2 & 1 & 4 & 5 \end{bmatrix} = \begin{bmatrix} -0.61 & 0.28 & -0.74 \\ -0.29 & -0.95 & -0.12 \\ -0.74 & 0.14 & 0.66 \end{bmatrix} \times \begin{bmatrix} 8.87 & 0.00 & 0.00 & 0.00 \\ 0.00 & 4.01 & 0.00 & 0.00 \\ 0.00 & 0.00 & 2.51 & 0.00 \end{bmatrix}$$

$$A_{n\times m} \qquad\qquad U_{n\times n} \qquad\qquad S_{n\times m}$$

$$\times \begin{bmatrix} -0.47 & -0.28 & -0.47 & -0.69 \\ 0.11 & -0.85 & -0.27 & 0.45 \\ -0.71 & -0.23 & 0.66 & 0.13 \\ -0.52 & 0.39 & -0.53 & 0.55 \end{bmatrix}$$

$$V^{\top}{}_{m\times m}$$

Σχήμα 5.2: Τρέχον παράδειγμα: $A_{n\times m}$ (αρχικός πίνακας A), $U_{n\times m}$ (αριστερά *μοναδιαία ιδιοδιανύσματα* του A), $S_{n\times m}$ (*μοναδιαίες ιδιοτιμές* του A), $V'_{m\times m}$ (δεξιά *μοναδιαία ιδιοδιανύσματα* του A).

5.3.2 Διατήρηση Πρωτευόντων Συνιστωσών

Είναι δυνατό να μειωθούν οι διαστάσεις του πίνακα S, ώστε να λάβουμε μόνο τις μεγαλύτερες μοναδιαίες του *ιδιοτιμές* c (*singular eigenvalues*). Τότε, ο αναδομημένος πίνακας $\hat{A}$ είναι η πλησιέστερη προσέγγιση τάξης-c του αρχικού πίνακα A, όπως εμφανίζεται στην Εξίσωση 5.13 και στο Σχήμα 5.3 για c=2:

$$\hat{A}_{n\times m} = U_{n\times c} \cdot S_{c\times c} \cdot V'_{c\times m} \tag{5.13}$$

$$\begin{bmatrix} 2.69 & 0.57 & 2.22 & 4.25 \\ 0.78 & 3.93 & 2.21 & 0.04 \\ 3.17 & 1.38 & 2.92 & 4.78 \end{bmatrix} = \begin{bmatrix} -0.61 & 0.28 \\ -0.29 & -0.95 \\ -0.74 & 0.14 \end{bmatrix} \times \begin{bmatrix} 8.87 & 0.00 \\ 0.00 & 4.01 \end{bmatrix}$$

$$\hat{A}_{n\times i} \qquad U_{n\times c} \qquad S_{c\times c}$$

$$\times \begin{bmatrix} -0.47 & -0.28 & -0.47 & -0.69 \\ 0.11 & -0.85 & -0.27 & 0.45 \end{bmatrix}$$

$$V^{\top}{}_{c\times m}$$

Σχήμα 5.3: Τρέχον παράδειγμα: $\hat{A}_{n\times m}$ (*πίνακας προσέγγισης* του A), $U_{n\times c}$ (αριστερά *μοναδιαία ιδιοδιανύσματα* του $\hat{A}$), $S_{c\times c}$ (*μοναδιαίες ιδιοτιμές του* $\hat{A}$), $V'_{c\times m}$ (δεξιά *μοναδιαία ιδιοδιανύσματα* του $\hat{A}$).

Ακολούθως, από τον πίνακα A επιλέγουμε έναν c αριθμό *πρωτευόντων συνιστωσών (διαστάσεων)* με σκοπό να αποκαλύψουμε τις μείζονες τάσεις βαθμολογικής συμπεριφοράς των χρηστών. Η τιμή των c διαστάσεων του πίνακα καθορίζεται από το ποσοστό πληροφορίας που διατηρείται σε σύγκριση με τον πρωτότυπο πίνακα. Έτσι, δημιουργείται ένας c-διάστατος χώρος όπου καθεμία από τις c διαστάσεις αντιστοιχεί σε μια διακριτή τάση των χρηστών για τη βαθμολογία των στοιχείων. Παρατηρούμε ότι στο τρέχον παράδειγμα δημιουργούμε έναν 2-διάστατο χώρο (c=2) χρησιμοποιώντας (12,88/15,39 = 83%) της συνολικής πληροφορίας της διαγωνίου του πίνακα S: ((8.87+4.01 = 12,88)/(8.87+4.01+2.51=15,39)).

5.3.3 Εισαγωγή του Προφίλ ενός Χρήστη σε Μικρότερο Διανυσματικό Χώρο

Στα συστήματα συστάσεων ο Sarwar και οι λοιποί συν-συγγραφείς [Sarwar et al., 2000] έχουν μελετήσει τη *μέθοδο SVD* σε συνδυασμό με τη *μέθοδο συνεργατικού φιλτραρίσματος*, εσφαλμένα θεωρώντας όμως τα *δεδομένα ελέγχου (test data)* ως εκ προοιμίου γνωστά. Είναι προφανές ότι για τον *αλγόριθμο συνεργατικού φιλτραρίσματος βάσει χρήστη (user-based collaborative filtering)* τα *δεδομένα ελέγχου* θα πρέπει να εξετάζονται ως άγνωστα στον c-διάστατο χώρο. Άρα, θα πρέπει να ακολουθείται μια εξειδικευμένη διαδικασία εισαγωγής των δεδομένων ελέγχου σ' αυτόν. Γι' αυτόν τον λόγο θα πρέπει να εισάγουμε τις βαθμολογίες του υπό εξέταση χρήστη u ως ένα διάνυσμα *ψευδο-χρήστη (pseudo-user)* στον c-διάστατο χώρο (δηλαδή, αναπαράσταση του χρήστη σ' έναν μικρότερο διανυσματικό χώρο) χρησιμοποιώντας την παρακάτω Εξίσωση 5.14:

$$\mathbf{u_{new}} = \mathbf{u} \cdot V_{m\times c} \cdot S^{-1}_{c\times c} \tag{5.14}$$

Έπειτα, στο τρέχον παράδειγμα 5.2 εισάγουμε τον U_4 στον 2-διάστατο χώρο όπως εμφανίζεται παρακάτω στο Σχήμα 5.4. Στην Εξίσωση 5.14 το διάνυσμα $\mathbf{u_{new}}$ αντιπροσωπεύει τις βαθμολογίες του υπό εξέταση χρήστη $\mathbf{u}$, όπου $V_{m\times c}$ και $S^{-1}_{c\times c}$ είναι οι πίνακες που παράγονται από τη *μέθοδο SVD*. Αυτό το $\mathbf{u_{new}}$ διάνυσμα θα πρέπει προστεθεί στο τέλος του $U_{n\times c}$ πίνακα που εμφανίζεται στο Σχήμα 5.3. Παρατηρούμε ότι οι νέες τιμές του διανύσματος του υπό εξέταση χρήστη U_4 είναι πολύ παρόμοιες με εκείνες του U_2 μετά την εισαγωγή. Αυτό είναι λογικό, διότι αυτοί οι δύο χρήστες έχουν παρόμοιες βαθμολογίες όπως φαίνεται στο Σχήμα 5.1.

$$\begin{bmatrix} -0.23 & -0.89 \end{bmatrix} = \begin{bmatrix} 1 & 4 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} -0.47 & 0.11 \\ -0.28 & -0.85 \\ -0.47 & -0.27 \\ -0.69 & 0.45 \end{bmatrix} \times \begin{bmatrix} 0.11 & 0.00 \\ 0.00 & 0.25 \end{bmatrix}$$

$$\mathbf{u}_{new} \qquad \mathbf{u} \qquad V_{m\times c} \qquad {S^{-1}}_{c\times c}$$

Σχήμα 5.4: Τρέχον παράδειγμα: $\mathbf{u_{new}}$ (νέα διανυσματική αναπαράσταση του χρήστη), $\mathbf{u}$ (το αρχικό διάνυσμα του χρήστη), $V_{m\times c}$ (δύο αριστερά *μοναδιαία ιδιοδιανύσματα* του V), $S^{-1}_{c\times c}$ (δύο *μοναδιαίες ιδιοτιμές* του αντιστρόφου του πίνακα S).

5.3.4 Δημιουργία Γειτονιάς Χρηστών/Στοιχείων

Διαθέτοντας πλέον μια μικρότερη αναπαράσταση του αρχικού διανυσματικού χώρου, διαμορφώνουμε τη γειτονιά των χρηστών/στοιχείων σ' αυτόν. Εκκινώντας από το *συνεργατικό φιλτράρισμα βάσει χρήστη (user-based collaborative filtering)*, βρίσκουμε τους k *κοντινότερους γείτονες* του διανύσματος *ψευδο-χρήστη* στον c-διάστατο χώρο. Οι ομοιότητες μεταξύ των χρηστών και του υπό εξέταση χρήστη μπορούν να βασισθούν στην *Ομοιότητα Συνημιτόνου (cosine similarity)*: πρώτα υπολογίζουμε τον πίνακα $U_{n\times c} \cdot S_{c\times c}$, και κατόπιν βρίσκουμε την *ομοιότητα συνημιτόνου* μεταξύ των διανυσμάτων των χρηστών και του υπό εξέταση χρήστη.

Περνώντας στο *συνεργατικό φιλτράρισμα βάσει στοιχείου (item-based collaborative filtering)*, βρίσκουμε τους k *κοντινότερους γείτονες* του διανύσματος του στοιχείου στον c-διάστατο χώρο. Πρώτα υπολογίζουμε τον πίνακα $S_{c\times c} \cdot V_{c\times m}$, και κατόπιν βρίσκουμε την *ομοιότητα συνημιτόνου* μεταξύ των διανυσμάτων των στοιχείων και του υπό εξέταση στοιχείου.

5.3.5 Δημιουργία Λίστας Προτεινόμενων Στοιχείων

Όπως έχει ήδη αναφερθεί στο 2ο κεφάλαιο, κατά το δεύτερο στάδιο της τεχνικής του *συνεργατικού φιλτραρίσματος* το βασικό κριτήριο για την κατάταξη των προτεινόμενων στοιχείων της

λίστας συστάσεων βασίζεται στο *πιο συχνά εμφανιζόμενο στοιχείο στη γειτονιά (most frequent item in the neighborhood)*. Εναλλακτικά, μπορεί να χρησιμοποιηθεί ένα κριτήριο κατάταξης που χρησιμοποιεί τις *προβλεφθείσες βαθμολογίες (predicted ratings)* ενός χρήστη για κάθε στοιχείο. Αυτές υπολογίζονται από τις Εξισώσεις 5.15 και 5.16 για τις περιπτώσεις του *συνεργατικού φιλτραρίσματος βάσει χρηστών* και *βάσει στοιχείων* αντίστοιχα:

$$p_{u,i} = \overline{r}_u + \frac{\sum_{v \in U} \text{sim}(u,v)(r_{v,i} - \overline{r}_v)}{\sum_{v \in U} |\text{sim}(u,v)|} \tag{5.15}$$

$$p_{u,i} = \overline{r}_i + \frac{\sum_{j \in \mathcal{I}} \text{sim}(i,j)(r_{u,j} - \overline{r}_j)}{\sum_{j \in \mathcal{I}} |\text{sim}(i,j)|} \tag{5.16}$$

Κατά συνέπεια, κατατάσσουμε σε φθίνουσα διάταξη τα στοιχεία σύμφωνα με την προβλεφθείσα βαθμολογία τους, και προτείνουμε τα εξ αυτών *κορυφαία N (top-N recommendation)*.

5.4 Αποσύνθεση Πίνακα κατά UV (UV-decomposition)

Εδώ θα παρουσιάσουμε το θεωρητικό και μαθηματικό υπόβαθρο του *αλγορίθμου UV-decomposition*. Θα χρησιμοποιήσουμε τη *μέθοδο UV-decomposition*, για να προβλέψουμε τις κενές (μηδενικές) καταχωρίσεις στον πίνακα βαθμολογίας χρήστη-στοιχείου A. Η μέθοδος αυτή λοιπόν, *UV-decomposition*, μπορεί να θεωρηθεί ως μια υποπερίπτωση της *μεθόδου SVD*, και εκφράζεται μαθηματικά ως εξής:

$$A = \tilde{U} \cdot \tilde{S} \cdot \tilde{V}^\top \tag{5.17}$$

Ο ισχυρισμός αυτός μπορεί να αποδειχθεί (μαθηματικά) αξιοποιώντας τις ιδιότητες της *γραμμικής παλινδρόμησης* (*linear regression*). Ας υποθέσουμε επομένως ότι:

- $\mathcal{U} = \tilde{U}$ όπου το $\tilde{U}$ περιέχει τα κορυφαία-k αριστερά *ιδιοδιανύσματα* του A.
- $\mathcal{V} = \tilde{S} \cdot \tilde{V}^\top$ όπου $\tilde{S} = diag(\sigma_1, \cdots, \sigma_k)$ και το $\tilde{V}$ περιέχει τα κορυφαία-k δεξιά *ιδιοδιανύσματα* του A.

Πιο συγκεκριμένα, η *γραμμική παλινδρόμηση* αναζητά δύο πίνακες $\mathcal{U} \in \mathbb{R}^{m \times k}$ και $\mathcal{V} \in \mathbb{R}^{k \times n}$ όπου $k \leq rank(A)$, ώστε αυτοί να ελαχιστοποιούν την ακόλουθη Εξίσωση:

$$minimize_{\mathcal{U}^\top \cdot \mathcal{U} = I_k} \|A - \mathcal{U} \cdot \mathcal{V}\|_F^2 \tag{5.18}$$

Σημειώστε ότι ο πίνακας $\mathcal{U}$ θα πρέπει να είναι ορθογώνιος. Δοθέντος ενός διανύσματος $\mathbf{a} \in \mathbb{R}^m$ και ενός πίνακα $\mathcal{U} \in \mathbb{R}^{m \times d}$, η *γραμμική παλινδρόμηση* βρίσκει ένα διάνυσμα συντελεστών $\mathbf{v} \in \mathbb{R}^d$ που προσεγγίζει γραμμικά το $\mathbf{a}$ με βάση το $\mathcal{U}$. Το διάνυσμα $\mathbf{v}$ ελαχιστοποιεί την τιμή $\|\mathbf{a} - \mathcal{U}\mathbf{v}\|_2^2$ υποθέτοντας ότι $rank(\mathcal{U}) = d$. Έτσι, το $\mathbf{v}$ μπορεί να εκφραστεί ως:

$$\mathbf{v} = (\mathcal{U}^\top \cdot \mathcal{U})^{-1} \cdot \mathcal{U}^\top \cdot \mathbf{a} \tag{5.19}$$

Με βάση τη *νόρμα Frobenius (Frobenius norm)* μπορούμε να ξαναγράψουμε την Εξίσωση 5.19 ως εξής:

$$\mathbf{v}_i = (\mathcal{U}^\top \cdot \mathcal{U})^{-1} \cdot \mathcal{U}^\top \cdot \mathbf{a}_i \tag{5.20}$$

και συνεπώς:

$$\begin{gathered}
\mathcal{V} = (\tilde{U}^\top \cdot \tilde{U})^{-1} \cdot \tilde{U}^\top \cdot A \iff \\
\mathcal{V} = (\tilde{U}^\top \cdot \tilde{U})^{-1} \cdot \tilde{U}^\top \cdot U \cdot S \cdot V^\top \iff \\
\mathcal{V} = I_k^{-1} \cdot (\tilde{U}^\top \cdot U) \cdot S \cdot V^\top \iff \\
\mathcal{V} = I_k^{-1} \cdot [I_k \quad 0] \cdot S \cdot V^\top \iff \\
\mathcal{V} = S \cdot V^\top
\end{gathered} \tag{5.21}$$

Με βάση, τώρα, την Εξίσωση 5.21 μπορούμε να ξαναγράψουμε τη *SVD* και την Εξίσωση 5.17 ως εξής:

$$A = \mathcal{U} \cdot \mathcal{V} \tag{5.22}$$

Αναλύοντας την παραπάνω απόδειξη βλέπουμε ότι υπάρχει ένας πίνακας $\mathcal{V}$, ο οποίος είναι το γινόμενο των S και $V^\top$. Άρα, ο πίνακας S αναμειγνύεται στα δεξιά (*right-blended*) με τον πίνακα $V^\top$ και παράγει τον πίνακα $\mathcal{V}$. Μπορούμε να υποθέσουμε ότι ο νέος πίνακας $\mathcal{V}$ περιέχει τόσο την πληροφορία με τις *ιδιοτιμές* του πίνακα βαθμολογίας χρήστη-στοιχείου A όσο και την πληροφορία του πίνακα V. Για τη *μέθοθο UV-decomposition* θα χρησιμοποιούμε στο εξής τα ίδια σύμβολα U και V για τους παραγοντοποιημένους πίνακές της και δεν θα υπάρχει καμμία σχέση με τους πίνακες U, V της μεθόδου *SVD*. Ένας εύκολος τρόπος για να προβλέψουμε τις κενές/μηδενικές τιμές του πίνακα βαθμολογίας χρήστη-στοιχείου A είναι να υποθέσουμε ότι ο πίνακας αυτός προκύπτει στην πραγματικότητα από το γινόμενο των δύο πινάκων U και V.

Για να διευκολύνουμε, λοιπόν, τη διερεύνηση του θέματός μας, θα παρουσιάσουμε βήμα-βήμα την εφαρμογή της μεθόδου *αποσύνθεσης κατά UV* στον πίνακα βαθμολογίας χρήστη-στοιχείου A του τρέχοντος παραδείγματός μας όπως φαίνεται στο Σχήμα 5.1. Σημειώστε ότι σ' αυτήν την περίπτωση συγχωνεύουμε το σύνολο των *δεδομένων εκπαίδευσης* του Σχ. 5.1a με το σύνολο των *δεδομένων ελέγχου* του Σχ. 5.1b και προκύπτει ένας πίνακας χρήστη-στοιχείου A με 4 γραμμές και 4 στήλες. Ο πίνακας βαθμολογίας χρήστη-στοιχείου A του τρέχοντος παραδείγματός μας έχει τώρα $|I|$ γραμμές και $|J|$ στήλες (μέγεθος $|I| \times |J| = 4 \times 4$). Στόχος μας είναι να βρούμε έναν πίνακα U (με μέγεθος $|I| \times k = 4 \times k$) και έναν πίνακα V (με μέγεθος $|J| \times k = 4 \times k$), έτσι ώστε το γινόμενό τους να προσεγγίζει τον A, ως ακολούθως:

$$A \approx U \cdot V^\top = \hat{A} \tag{5.23}$$

Γενικότερα, εάν έχουμε έναν πίνακα A με n γραμμές και m στήλες, μπορούμε να βρούμε δύο πίνακες: τον πίνακα U με n γραμμές και k στήλες και τον πίνακα V με m γραμμές και k στήλες, έτσι ώστε ο $UV^\top$ να προσεγγίζει τον A, και άρα οι περισσότερες κενές/μηδενικές τιμές του A να έχουν συμπληρωθεί με μια μικρή απόκλιση μεταξύ των αρχικών και των προβλεπόμενων τιμών του.

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} = \begin{bmatrix} u_{11} & \cdots & u_{1k} \\ \vdots & \ddots & \vdots \\ u_{m1} & \cdots & u_{mk} \end{bmatrix} \times \begin{bmatrix} v_{11} & \cdots & v_{1n} \\ \vdots & \ddots & \vdots \\ v_{k1} & \cdots & v_{kn} \end{bmatrix}$$

$$A_{m\times n} \qquad U_{m\times k} \qquad V^\top_{k\times n}$$

Σχήμα 5.5: Η γενική μορφή της *Αποσύνθεσης κατά UV*. Ο αρχικός πίνακας A μπορεί να παρουσιαστεί ως γινόμενο των δύο πινάκων U και V.

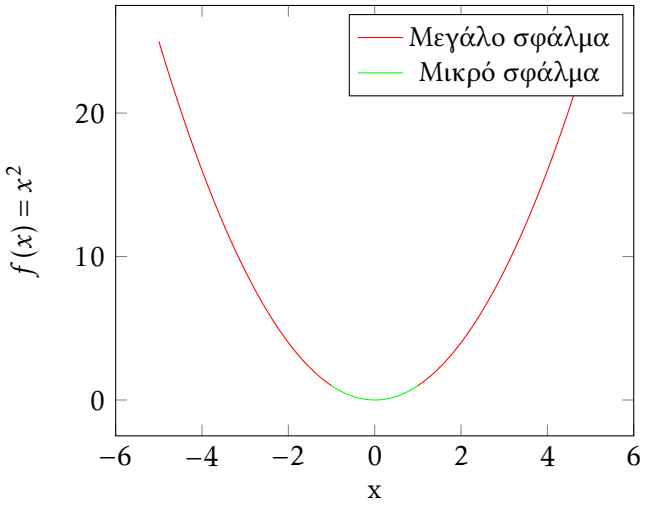
Για να προβλέψουμε τη βαθμολογία ενός χρήστη για ένα στοιχείο μπορούμε να υπολογίσουμε το εσωτερικό γινόμενο των δύο διανυσμάτων, τα οποία αντιστοιχούν στα $\mathbf{u}_i$ και $\mathbf{v}_j$, ως εξής:

$$\hat{a}_{ij} = \mathbf{u}_i \cdot \mathbf{v}_j^\top = \sum_{k=1}^{K} u_{ik} \cdot v_{kj} \tag{5.24}$$

Το επόμενο βήμα βάσει της μεθόδου είναι να επιλέξουμε έναν τρόπο ώστε να ορίσουμε τις αρχικές τιμές για τα U και V. Ένας τρόπος επίλυσης του προβλήματος είναι να αρχικοποιήσουμε τους δύο πίνακες με κάποιες τυχαίες τιμές και να υπολογίσουμε πόσο διαφέρει το γινόμενό τους σε σχέση με τον αρχικό πίνακα A. Εν συνέχεια, επαναλαμβάνοντας τη διαδικασία προσπαθούμε να ελαχιστοποιήσουμε αυτήν τη διαφορά. Μια τέτοια μέθοδος αριθμητικής προσέγγισης ονομάζεται **"μέθοδος Επικλινούς Καθόδου" (Gradient Descent method)** ή αλλιώς **"Κανόνας Δέλτα"**, και στοχεύει στο να ελαχιστοποιήσει το σφάλμα της διαφοράς μεταξύ της πραγματικής και της προβλεπόμενης βαθμολογίας ως προς κάθε ζεύγος χρήστη-στοιχείου χρησιμοποιώντας την Εξίσωση 5.25:

$$e_{ij}^2 = (a_{ij} - \hat{a}_{ij})^2 = (a_{ij} - \sum_{k=1}^{K} u_{ik} v_{kj})^2 \tag{5.25}$$

Προσέξτε ότι υπολογίζουμε το τετραγωνικό σφάλμα στην Εξίσωση 5.25, επειδή η προβλεπόμενη βαθμολογία μπορεί να είναι είτε υψηλότερη είτε χαμηλότερη από την πραγματική. Επιπλέον, υπολογίζοντας το τετράγωνο του σφάλματος "επιβραβεύουμε" τα μικρά σφάλματα και "τιμωρούμε" τα μεγαλύτερα. Το Σχήμα 5.6 απεικονίζει τη συνάρτηση $f(x) = x^2$. Όπως λοιπόν φαίνεται σ' αυτό, το τμήμα με την πράσινη γραμμή είναι η περιοχή όπου το τετραγωνικό σφάλμα είναι μικρό και μ' αυτόν τον τρόπο "επιδοκιμάζουμε" αυτές τις διαφορές έναντι της περιοχής με το κόκκινο χρώμα όπου το σφάλμα είναι σημαντικό και επομένως έτσι "αποδοκιμάζουμε" αυτές τις διαφορές.



Σχήμα 5.6: Γράφημα του εύρους των σφαλμάτων μεταξύ των πραγματικών και των προβλεπόμενων βαθμολογιών. Χρησιμοποιώντας τη συνάρτηση $f(x) = x^2$ "επιβραβεύουμε" τα μικρά λάθη και "τιμωρούμε" τα μεγάλα.

5.4.1 Ορισμός Αντικειμενικής Συνάρτησης

Σ' αυτήν την παράγραφο θα ορίσουμε την *αντικειμενική συνάρτηση*. Έχει ως εξής:

$$G = \|A - \hat{A}\|_F^2 = \|A - UV\|_F^2 = e_{ij}^2 = (a_{ij} - \sum_{k=1}^{K} u_{ik} v_{kj})^2 \tag{5.26}$$

Όπως φαίνεται από την Εξίσωση 5.26, η *αντικειμενική συνάρτηση* ισούται με τη συνάρτηση τετραγωνικού σφάλματος της Εξίσωσης 5.25. Έτσι, η πρόκληση που έχουμε να αντιμετωπίσουμε είναι να ελαχιστοποιήσουμε το σφάλμα μεταξύ των πραγματικών και των προβλεπόμενων τιμών του πίνακα βαθμολογίας χρήστη-στοιχείου A. Δεδομένου ότι δεν υπάρχει λύση κλειστής μορφής για τη συνάρτηση G, χρησιμοποιούμε μια μέθοδο αριθμητικής προσέγγισης, όπως η μέθοδος της **επικλινούς καθόδου** (*Gradient Descent*), για την επίλυσή της. Αναλυτικότερα, η μέθοδος *Gradient Descent* βρίσκει την κατεύθυνση προς την οποία θα πρέπει να αλλάξει η κάθε καταχώριση του πίνακα A έτσι ώστε να μειωθεί όσο το δυνατόν περισσότερο η *συνάρτηση σφάλματος*. Πιο συγκεκριμένα, για να ελαχιστοποιήσουμε τη *συνάρτηση σφάλματος*, πρέπει να γνωρίζουμε προς ποια κατεύθυνση πρέπει να τροποποιήσουμε τις τιμές των u_{ik} και v_{kj} της Εξίσωσης 5.26, οπότε υπολογίζουμε τις μερικές παραγώγους της G ως προς U και V ως εξής:

$$\frac{\partial G}{\partial U} = \frac{\partial}{\partial u_{ik}} e_{ij}^2 = 2(a_{ij} - \hat{a}_{ij}) \frac{\partial}{\partial u_{ik}} (a_{ij} - \hat{a}_{ij}) \iff$$

$$\frac{\partial G}{\partial U} = \frac{\partial}{\partial u_{ik}} e_{ij}^2 = 2(a_{ij} - \hat{a}_{ij})(-v_{kj}) \iff$$

$$\frac{\partial G}{\partial U} = \frac{\partial}{\partial u_{ik}} e_{ij}^2 = -2 \cdot e_{ij} \cdot v_{kj}$$

$$\wedge$$

$$\frac{\partial G}{\partial V} = \frac{\partial}{\partial v_{kj}} e_{ij}^2 = 2(a_{ij} - \hat{a}_{ij}) \frac{\partial}{\partial v_{kj}} (a_{ij} - \hat{a}_{ij}) \iff$$

$$\frac{\partial G}{\partial V} = \frac{\partial}{\partial v_{kj}} e_{ij}^2 = 2(a_{ij} - \hat{a}_{ij})(-u_{ik}) \iff$$

$$\frac{\partial G}{\partial V} = \frac{\partial}{\partial v_{kj}} e_{ij}^2 = -2 \cdot e_{ij} \cdot u_{ik}$$

Έχοντας υπολογίσει το *gradient* (δηλαδή το διάνυσμα που κρατά τις μερικές παραγώγους της συνάρτησης, επίσης γνωστό ως *ανάδελτα*), μπορούμε τώρα να ορίσουμε τους *κανόνες ενημέρωσης* τόσο για το u_{ik} όσο και για το v_{kj} ως ακουλούθως:

$$u'_{ik} = u_{ik} - \eta \cdot \frac{\partial}{\partial u_{ik}} \cdot e_{ij}^2 = u_{ik} + 2 \cdot \eta \cdot e_{ij} \cdot v_{kj}$$

$$\wedge$$

$$v'_{kj} = v_{kj} - \eta \cdot \frac{\partial}{\partial v_{kj}} \cdot e_{ij}^2 = v_{kj} + 2 \cdot \eta \cdot e_{ij} \cdot u_{ik}$$

Εδώ, η μεταβλητή η είναι μια σταθερά της οποίας η τιμή καθορίζει τον *ρυθμό προσέγγισης (learning rate)* του ελάχιστου. Συνήθως επιλέγουμε μια μικρή τιμή βήματος για το η, όπως 0.0002. Ο λόγος είναι ότι, εάν κάνουμε ένα μεγάλο βήμα προς το ελάχιστο, διατρέχουμε τον κίνδυνο να το χάσουμε και να καταλήξουμε να περιφερόμαστε γύρω από αυτό.

5.4.2 Αποφυγή Υπερπροσαρμογής μέσω Κανονικοποίησης

Ένα συνηθισμένο πρόβλημα στη μοντελοποίηση προβλέψεων είναι η *υπερπροσαρμογή* (*overfitting*): Συνήθως, ένα μοντέλο πρόβλεψης εκπαιδεύεται και μεγιστοποιεί την απόδοσή του με βάση κάποιο σύνολο *δεδομένων εκπαίδευσης*. Η αποτελεσματικότητά του ωστόσο δεν καθορίζεται από την απόδοσή του στα *δεδομένα εκπαίδευσης*, αλλά από την ικανότητά του να αποδίδει καλά σε *δεδομένα δοκιμής*. Το πρόβλημα της *υπερπροσαρμογής* εμφανίζεται όταν ένα μοντέλο αρχίζει να "απομνημονεύει" τα δεδομένα εκπαίδευσης αντί να "μαθαίνει" να γενικεύει από την τάση που επικρατεί στα *δεδομένα εκπαίδευσης*. Μπορούμε όμως να αποφύγουμε τον σκόπελο αυτόν προσθέτοντας στην *αντικειμενική* μας *συνάρτηση* μία παράμετρο β και τροποποιώντας το *τετραγωνικό σφάλμα* ως εξής:

$$G = e_{ij}^2 = (a_{ij} - \sum_{k=1}^{K} u_{ik} v_{kj})^2 + \frac{\beta}{2} \sum_{k=1}^{K} (\|U\|^2 + \|V\|^2) \tag{5.27}$$

Η νέα παράμετρος β χρησιμοποιείται για τον έλεγχο των μεγεθών των διανυσμάτων χρήστη και στοιχείου -της *νόρμας Frobenius* των διανυσμάτων χρήστη και στοιχείου ουσιαστικά- για την οποία μπορούμε να πούμε πως με όρους *Συναρτησιακής Ανάλυσης* είναι η *Ευκλείδεια νόρμα (L^2 νόρμα)*. Πιο συγκεκριμένα, μειώνει τις τυχόν μεγάλες αριθμητικές τιμές του ενός διανύσματος έτσι ώστε να μην μπορούν να κυριαρχήσουν στις πιθανά μικρότερες αριθμητικές τιμές του άλλου διανύσματος. Στην πράξη το β τίθεται σε πολύ μικρές τιμές, της τάξης του 0.02. Για την ελαχιστοποίηση της νέας *επαυξημένης συνάρτησης σφάλματος*, πρέπει να υπολογιστούν εκ νέου οι μερικές παράγωγοι της συνάρτησης G ως προς U και V ως ακολούθως:

$$\frac{\partial G}{\partial U} = \frac{\partial}{\partial u_{ik}} e_{ij}^2 = 2(a_{ij} - \hat{a}_{ij}) \frac{\partial}{\partial u_{ik}} (a_{ij} - \hat{a}_{ij}) + \beta u_{ik} \iff$$

$$\frac{\partial G}{\partial U} = \frac{\partial}{\partial u_{ik}} e_{ij}^2 = 2(a_{ij} - \hat{a}_{ij})(-v_{kj}) + \beta u_{ik} \iff$$

$$\frac{\partial G}{\partial U} = \frac{\partial}{\partial u_{ik}} e_{ij}^2 = -2 \cdot e_{ij} \cdot v_{kj} + \beta \cdot u_{ik}$$

$$\wedge$$

$$\frac{\partial G}{\partial V} = \frac{\partial}{\partial v_{kj}} e_{ij}^2 = 2(a_{ij} - \hat{a}_{ij}) \frac{\partial}{\partial v_{kj}} (a_{ij} - \hat{a}_{ij}) + \beta v_{kj} \iff$$

$$\frac{\partial G}{\partial V} = \frac{\partial}{\partial v_{kj}} e_{ij}^2 = 2(a_{ij} - \hat{a}_{ij})(-u_{ik}) + \beta v_{kj} \iff$$

$$\frac{\partial G}{\partial V} = \frac{\partial}{\partial v_{kj}} e_{ij}^2 = -2 \cdot e_{ij} \cdot u_{ik} + \beta \cdot v_{kj}$$

Οι νέοι *κανόνες ενημέρωσης* τόσο για το διάνυσμα χρήστη u_{ik} όσο και για το διάνυσμα στοιχείου v_{kj} έχουν ως εξής:

$$u'_{ik} = u_{ik} - \eta \frac{\partial}{\partial u_{ik}} e_{ij}^2 = u_{ik} + \eta \cdot (2 \cdot e_{ij} \cdot v_{kj} - \beta \cdot u_{ik})$$

$$\wedge$$

$$v'_{kj} = v_{kj} - \eta \frac{\partial}{\partial v_{kj}} e_{ij}^2 = v_{kj} + \eta \cdot (2 \cdot e_{ij} \cdot u_{ik} - \beta \cdot v_{kj})$$

5.4.3 Ο Αλγόριθμος UV-decomposition

Στην παρούσα υποενότητα παρουσιάζουμε (σε μορφή ψευδοκώδικα) τον *αλγόριθμο UV-decomposition* [Bhargava et al., 2015], όπως αυτός φαίνεται στον Αλγόριθμο 1. Αναλυτικότερα, τα δεδομένα εισόδου στον Αλγόριθμο 1 είναι: (α) η *αντικειμενική συνάρτηση* $G = \|A - \hat{A}\|^2$, η οποία θα πρέπει όμως να ελαχιστοποιηθεί, και (β) ο πίνακας βαθμολογίας χρήστη-στοιχείου A ($A \in \mathbb{R}^{m \times n}$). Επιπλέον, η παράμετρος k, η οποία ελέγχει το μέγεθος των πινάκων U και V, καθώς και η παράμετρος η, η οποία εκφράζει τον *ρυθμό μάθησης (learning rate)* και ελέγχει το μέγεθος του *βήματος* προς την κατεύθυνση της απομείωσης της *αντικειμενικής συνάρτησης* G.

Με βάση την παράμετρο k στη γραμμή 1 του Αλγορίθμου 1 αρχικοποιούμε με τυχαίες τιμές τους πίνακες U και V. Ομοίως πράττουμε και για τη συνάρτηση G και υπολογίζουμε τις μερικές παραγώγους ($\frac{\partial G}{\partial U}$ και $\frac{\partial G}{\partial V}$) για U και V αντίστοιχα. Μετά τις παραπάνω ενέργειες ο αλγόριθμος ξεκινά τη διαδικασία σύγκλισης της πρόβλεψης της βαθμολογίας ενός χρήστη για ένα στοιχείο στις γραμμές 2 και 3. Συγκεκριμένα, χρησιμοποιούμε την παλιά τιμή των U_i και V_i αντίστοιχα, και κάνουμε ένα βήμα προς την κατεύθυνση της απομείωσης της συνάρτησης G. Αυτή η διαδικασία θα εκτελεστεί επανειλημμένα έως ότου το σφάλμα (δηλ. $\|A - \hat{A}\|^2$) μεταξύ των πραγματικών και των προβλεπόμενων βαθμολογιών των πινάκων A και $\hat{A}$ αντίστοιχα να πάψει να μειώνεται ή έως ότου επιτευχθεί ένας μέγιστος αριθμός επαναλήψεων (γραμμή 3). Στη γραμμή 4 λαμβάνουμε το γινόμενο των υπολογισμένων πινάκων U και V, και το οποίο παράγει τον τελικό πίνακα πρόβλεψης βαθμολογίας $\hat{A}$.

Αλγόριθμος 1 Αποσύνθεση κατά UV

Είσοδος: Η *αντικειμενική συνάρτηση* $G = \|A - \hat{A}\|^2$, ένας πίνακας βαθμολογίας χρηστών $A \in \mathbb{R}^{m \times n}$, η παράμετρος k που ελέγχει το μέγεθος των πινάκων U και V, και η παράμετρος η που ελέγχει το μέγεθος του βήματος προς την κατεύθυνση της απομείωσης της *αντικειμενικής συνάρτησης* G.

Έξοδος: Ένας πίνακας $\hat{A} \in \mathbb{R}^{m \times n}$, ο οποίος προβλέπει τη βαθμολογία των χρηστών επί στοιχείων.

1: Αρχικοποίηση των πινάκων U και V, της *αντικειμενικής συνάρτησης* G, και των μερικών παραγώγων $\frac{\partial G}{\partial U}$, $\frac{\partial G}{\partial V}$

2: **Επανάλαβε**:

$$U_i' = U_i - \eta \cdot \frac{\partial G}{\partial U},$$
$$V_i' = V_i - \eta \cdot \frac{\partial G}{\partial V}$$

3: **μέχρις ότου** η $\|A - \hat{A}\|^2$ πάψει να μειώνεται ή μέχρι ο μέγιστος αριθμός επαναλήψεων να έχει επιτευχθεί.

4: $\hat{A} \leftarrow U \cdot V$

5: **Επίστρεψε** $\hat{A}$

Παράδειγμα 5.3 Έστω ότι μας ζητείται ο υπολογισμός βήμα προς βήμα της μεθόδου *UV-Decomposition (αποσύνθεση κατά UV)* για το τρέχον παράδειγμά του Σχήματος 5.1. Σημειώστε ότι σ' αυτήν την περίπτωση συγχωνεύουμε το *σύνολο εκπαίδευσης* του Σχ. 5.1(α) με το *σύνολο δοκιμής* του Σχ. 5.1(β). Ο χρήστης των *δεδομένων δοκιμής* συμπεριλαμβάνεται δηλαδή στον υπολογισμό της *UV-decomposition* για λόγους συντομίας. Ο νέος συγχωνευμένος αρχικός πίνακας αξιολόγησης χρηστών A του τρέχοντος παραδείγματός μας παρουσιάζεται στο Σχήμα 5.7.

$$\begin{bmatrix} 4 & 1 & 1 & 4 \\ 1 & 4 & 2 & 0 \\ 2 & 1 & 4 & 5 \\ 1 & 4 & 1 & ? \end{bmatrix} = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \\ u_{31} & u_{32} \\ u_{41} & u_{42} \end{bmatrix} \times \begin{bmatrix} v_{11} & v_{12} & v_{13} & v_{14} \\ v_{21} & v_{22} & v_{23} & v_{24} \end{bmatrix}$$

$$A_{4\times 4} \qquad U_{4\times 2} \qquad V^{\top}_{2\times 4}$$

Σχήμα 5.7: Εφαρμόζουμε την *αποσύνθεση κατά UV* στο παράδειγμά μας με $k = 2$ και προσπαθούμε να υπολογίσουμε τους πίνακες U και V.

Όπως φαίνεται σ' αυτό, *αποσυνθέτουμε* τον A σ' έναν πίνακα U διαστάσεων 4 επί 2 και σ' έναν πίνακα V διαστάσεων 2 επί 4. Αυτό σημαίνει ότι δημιουργούμε έναν δισδιάστατο χώρο (k=2) διατηρώντας μόνο τις δύο βασικές διαστάσεις από τους U και V αντίστοιχα.
Η *αποσύνθεση κατά UV* ξεκινά με κάποιες τυχαίες αρχικές τιμές για τους πίνακες U και V και στη συνέχεια τους επαναπροσδιορίζει με διαδοχικές επαναλήψεις, για να μειώσει το *Root Mean Square Error (RMSE)* μεταξύ των πραγματικών και των προβλεπόμενων βαθμολογιών του πίνακα A. Ο ορισμός του *RMSE* που θέλουμε να ελαχιστοποιήσουμε φαίνεται στην Εξίσωση 9.3:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}\sum_{j=1}^{m}(a_{ij} - \hat{a}_{ij})^2}{D}}, \quad \forall i \in n \text{ και } j \in m : a_{ij} \neq 0 \tag{5.28}$$

όπου D ο αριθμός των μη μηδενικών βαθμολογιών του πίνακα βαθμολογίας χρήστη-στοιχείου A. Ας υποθέσουμε τώρα ότι στο τρέχον παράδειγμα θέτουμε όλα τα αρχικά στοιχεία των πινάκων U και V να ισούνται με 1, ενώ το γινόμενό τους δημιουργεί τον πίνακα $\hat{A}$, ο οποίος περιέχει τις προβλεπόμενες βαθμολογίες, όπως φαίνεται και στο Σχήμα 5.8. Έπειτα, μπορούμε με βάση την Εξίσωση 9.3 να υπολογίσουμε το *RMSE* για τους πίνακες A και $\hat{A}$ των Σχημάτων 5.7 και 5.8 αντίστοιχα. Στην περίπτωσή μας το *RMSE* ισούται με 1,581. Ο τρόπος υπολογισμού του *RMSE* για το παρόν παράδειγμα περιγράφεται αναλυτικά στην ενότητα 9.2 του αντίστοιχου κεφαλαίου.

$$\begin{bmatrix} 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\hat{A}_{4\times4} \qquad U_{4\times2} \qquad V^{\top}_{2\times4}$$

Σχήμα 5.8: Πίνακες U και V με όλα τα στοιχεία να ισούνται με 1.

Για να μειώσουμε, τώρα, το *RMSE* της *αντικειμενικής μας συνάρτησης* σταδιακά, μπορούμε να κάνουμε σχετικές βελτιώσεις τιμών σε κάθε ξεχωριστό στοιχείο των πινάκων U και V αντίστοιχα. Ορίζουμε επομένως τα στοιχεία των U και V του Σχήματος 5.8 με τέτοιον τρόπο ώστε όλες οι καταχωρίσεις τους να είναι αρχικά ίσες με 1, με στόχο να βρούμε εκείνες τις τιμές (για τις καταχωρίσεις αυτές) που θα δώσουν τη μεγαλύτερη δυνατή βελτίωση στο *RMSE*. Έχοντας ως σημείο εκκίνησης της *αποσύνθεσης κατά UV* τους πίνακες του Σχήματος 5.8, αλλάζουμε αρχικά την τιμή του στοιχείου u_{11} του πίνακα U, για να μειώσουμε όσο το δυνατόν περισσότερο το *RMSE*. Έστω, λοιπόν, ότι συμβολίζουμε το στοιχείο u_{11} ως μεταβλητή x, όπως φαίνεται στο Σχήμα 5.9:

$$\begin{bmatrix} x+1 & x+1 & x+1 & x+1 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \end{bmatrix} = \begin{bmatrix} x & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\hat{A}_{4\times4} \qquad U_{4\times2} \qquad V^{\top}_{2\times4}$$

Σχήμα 5.9: Το πρώτο βήμα της μεθόδου *UV-decomposition*. Θέτουμε το στοιχείο u_{11} του πίνακα U ως μεταβλητή x.

Παρατηρήστε στο Σχήμα 5.9 ότι τα μόνα στοιχεία του πίνακα $\hat{A}$ που αλλάζουν είναι αυτά της πρώτης γραμμής. Έτσι, όταν συγκρίνουμε τον πίνακα $\hat{A}$ του Σχήματος 5.9 με τον πίνακα A του Σχήματος 5.7, η μόνη αλλαγή στο *RMSE* προέρχεται από την πρώτη γραμμή. Το άθροισμα λοιπόν της διαφοράς των τετραγώνων από την πρώτη γραμμή των πινάκων $\hat{A}$ και A του Σχήματος 5.10 είναι το ακόλουθο:

$$C = (4-(x+1))^2 + (1-(x+1))^2 + (1-(x+1))^2 + (4-(x+1))^2 \iff$$
$$C = (16 - 2(4(x+1)) + (x+1)^2) + (1 - 2(x+1) + (x+1)^2) +$$
$$+(1 - 2(x+1) + (x+1)^2) + (16 - 2(4(x+1)) + (x+1)^2) \iff$$

$$C = (16 - 8x - 8 + x^2 + 2x + 1) + (1 - 2x - 2 + x^2 + 2x + 1) +$$
$$+(1 - 2x - 2 + x^2 + 2x + 1) + (16 - 8x - 8 + x^2 + 2x + 1) \iff$$
$$C = (9 - 6x + x^2) + x^2 + x^2 + (9 - 6x + x^2) \iff$$
$$C = (3 - x)^2 + x^2 + x^2 + (3 - x)^2 = 2((3 - x)^2 + x^2) \iff$$
$$C = 2(9 - 6x + 2x^2)$$

$$\begin{bmatrix} 4 & 1 & 1 & 4 \\ 1 & 4 & 2 & 0 \\ 2 & 1 & 4 & 5 \\ 1 & 4 & 1 & 0 \end{bmatrix} \quad \& \quad \begin{bmatrix} x+1 & x+1 & x+1 & x+1 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \end{bmatrix}$$

$\hat{A}_{4\times4}$ $A_{4\times4}$

Σχήμα 5.10: Η μόνη αλλαγή στο *RMSE* προέρχεται από την πρώτη γραμμή των πινάκων $\hat{A}$ και A αντίστοιχα.

Στη συνέχεια, θέλουμε να βρούμε την τιμή του x που ελαχιστοποιεί το άθροισμα της διαφοράς των τετραγώνων. Από την παραπάνω εξίσωση παίρνουμε την πρώτη παράγωγό της και τη θέτουμε ίση με 0 ως εξής:

$$\frac{\partial C}{\partial x} = -6 + 4x = 0 \iff x = 1.5 \tag{5.29}$$

Μετά τον υπολογισμό της τιμής της μεταβλητής x μπορούμε να λογαριάσουμε εκ νέου τις προβλεπόμενες τιμές των στοιχείων της πρώτης γραμμής του $\hat{A}$, οι οποίες επικαιροποιημένες φαίνονται τώρα στο Σχήμα 5.11. Προσέξτε ότι στην πρώτη γραμμή των πινάκων A και $\hat{A}$ το άθροισμα των τετραγώνων έχει μειωθεί τώρα από 10 σε 9. Έτσι, το συνολικό *RMSE* μειώθηκε από 1,581 σε 1,558. Ας υποθέσουμε τώρα ότι θέλουμε να αλλάξουμε την τιμή του στοιχείου v_{11} του πίνακα V. Έστω λοιπόν ότι συμβολίζουμε το στοιχείο v_{11} ως μεταβλητή y, όπως φαίνεται στο Σχήμα 5.12.

$$\begin{bmatrix} 2.5 & 2.5 & 2.5 & 2.5 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \end{bmatrix} = \begin{bmatrix} 1.5 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$\hat{A}_{4\times4}$ $U_{4\times2}$ $V^{\top}_{2\times4}$

Σχήμα 5.11: Τα αποτελέσματα μετά την εκτέλεση του πρώτου βήματος του *αλγορίθμου UV-decomposition.*

$$\begin{bmatrix} 1.5y+1 & 2.5 & 2.5 & 2.5 \\ y+1 & 2 & 2 & 2 \\ y+1 & 2 & 2 & 2 \\ y+1 & 2 & 2 & 2 \end{bmatrix} = \begin{bmatrix} 1.5 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \times \begin{bmatrix} y & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\hat{A}_{4\times4} \qquad U_{4\times2} \qquad V^{\top}_{2\times4}$$

Σχήμα 5.12: Το δεύτερο βήμα της *μεθόδου UV-decomposition*. Θέτουμε το στοιχείο v_{11} του πίνακα V ως τη μεταβλητή y.

Με παρόμοιο με αυτόν του Σχήματος 5.10 τρόπο, μπορούμε να παρατηρήσουμε -όπως φαίνεται στο Σχήμα 5.13- ότι μόνο η πρώτη στήλη του πίνακα $\hat{A}$ επηρεάζεται από το y. Ως εκ τούτου, όταν συγκρίνουμε τον πίνακα $\hat{A}$ με τον πίνακα A, η μόνη αλλαγή στο *RMSE* προέρχεται από την πρώτη στήλη:

$$\begin{bmatrix} 4 & 1 & 1 & 4 \\ 1 & 4 & 2 & 0 \\ 2 & 1 & 4 & 5 \\ 1 & 4 & 1 & 0 \end{bmatrix} \quad \& \quad \begin{bmatrix} 1.5y+1 & 2.5 & 2.5 & 2.5 \\ y+1 & 2 & 2 & 2 \\ y+1 & 2 & 2 & 2 \\ y+1 & 2 & 2 & 2 \end{bmatrix}$$

$$\hat{A}_{4\times4} \qquad A_{4\times4}$$

Σχήμα 5.13: Η μόνη αλλαγή στο *RMSE* προέρχεται από την πρώτη στήλη.

Η συνεισφορά στο άθροισμα των τετραγώνων από την πρώτη στήλη είναι:

$$D = (4-(1.5y+1))^2 + (1-(y+1))^2 + (2-(y+1))^2 + (1-(y+1))^2 \iff$$
$$D = (16 - 2(4(1.5y+1)) + (1.5y+1)^2 + (1-2(y+1)) + (y+1)^2 +$$
$$+(4-2(2(y+1)) + (y+1)^2 + (1-2(y+1)) + (y+1)^2 \iff$$
$$D = (16 - 12y - 8 + 2.25y^2 + 3y + 1) + (1 - 2y - 2 + y^2 + 2y + 1) +$$
$$+(4 - 4y - 4 + y^2 + 2y + 1) + (1 - 2y - 2 + y^2 + 2y + 1) \iff$$
$$D = (9 - 9y + 2.25y^2) + y^2 + (1 - 2y + y^2) + y^2 \iff$$
$$D = 10 - 13y + 5.25y^2$$

Όπως και νωρίτερα (βλ. την Εξίσωση 5.29), θέλουμε να βρούμε την τιμή της μεταβλητής y που ελαχιστοποιεί το άθροισμα των τετραγώνων. Παίρνουμε λοιπόν από την παραπάνω Εξίσωση την πρώτη παράγωγό της και τη θέτουμε ίση με 0 ως εξής:

$$\frac{\partial D}{\partial y} = -13 + 10.5y = 0 \iff$$

$$y = 1.238 \tag{5.30}$$

Μετά τον υπολογισμό της τιμής της μεταβλητής y, μπορούμε να λογαριάσουμε εκ νέου τις αρχικές προβλεπόμενες τιμές των στοιχείων της πρώτης στήλης του $\hat{A}$, οι οποίες επικαιροποιημένες τώρα φαίνονται στο Σχήμα 5.14:

$$\begin{bmatrix} 2.571 & 2.5 & 2.5 & 2.5 \\ 2.047 & 2 & 2 & 2 \\ 2.047 & 2 & 2 & 2 \\ 2.047 & 2 & 2 & 2 \end{bmatrix} = \begin{bmatrix} 1.5 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \times \begin{bmatrix} 1.047 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\hat{A}_{4\times4} \qquad U_{4\times2} \qquad V^{\top}_{2\times4}$$

Σχήμα 5.14: Τα αποτελέσματα μετά το δεύτερο βήμα του *αλγορίθμου της κατά UV αποσύνθεσης*.

Επαναλαμβάνοντας τα παραπάνω βήματα για κάθε στοιχείο των πινάκων U και V, θα προσπαθήσουμε να ελαχιστοποιήσουμε περαιτέρω το *RMSE* και να παράξουμε έναν προβλεπόμενο πίνακα $\hat{A}$ που θα προσεγγίζει τις πραγματικές τιμές του A. Στο Σχήμα 5.15, που ακολουθεί, παρουσιάζεται ο τελικός προβλεπόμενος πίνακας $\hat{A}$ ($\eta = 0.002$, $\beta = 0.02$) για το τρέχον παράδειγμα:

$$\begin{bmatrix} 2.56 & 0.82 & 2.36 & 4.08 \\ 1.20 & 3.99 & 1.71 & 0.02 \\ 3.10 & 1.13 & 2.06 & 4.86 \\ 0.74 & 3.89 & 1.31 & -0.71 \end{bmatrix} = \begin{bmatrix} 0.67 & 1.67 \\ 1.77 & 0.22 \\ 0.86 & 1.97 \\ 1.68 & -0.52 \end{bmatrix} \times \begin{bmatrix} 0.82 & 2.20 & 1.08 & 0.30 \\ 1.21 & -0.39 & 0.97 & 2.33 \end{bmatrix}$$

$$\hat{A} \qquad U \qquad V$$

Σχήμα 5.15: Το γινόμενο των πινάκων U και V με το οποίο προβλέπουμε τον πίνακα $\hat{A}$.

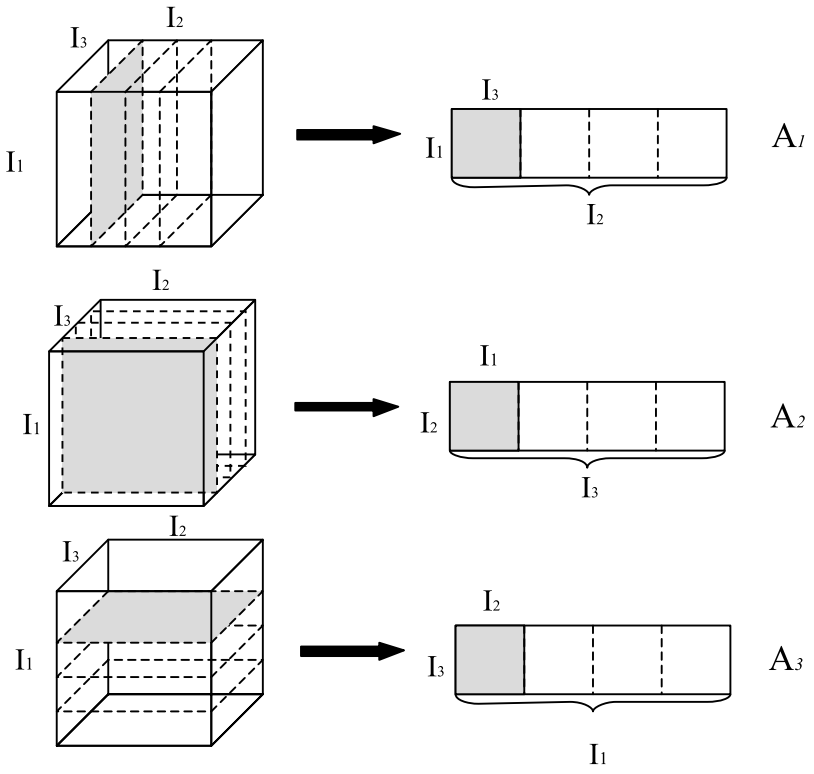
Σημειώστε ότι η προβλεπόμενη βαθμολογία του χρήστη U_4 στο στοιχείο I_4 είναι κοντά στο 0. Αυτό είναι αναμενόμενο, επειδή οι χρήστες U_2 και U_4 έχουν παρόμοιες βαθμολογίες στα στοιχεία I_1, I_2 και I_3. Έτσι, ο χρήστης U_4 προβλέπεται να έχει την ίδια βαθμολογία με τον U_2 στο στοιχείο I_4.

5.5 Παραγοντοποίηση Τανυστή (Tensor decomposition)

Ένας *Τανυστής* είναι ένας *πολυδιάστατος πίνακας N-τάξης*. Εφεξής, ένας *τανυστής* θα συμβολίζεται με το καλλιγραφικό γράμμα $\mathcal{A} \in \mathbb{R}^{I_1 \dots I_N}$, με στοιχεία $a_{i_1,\dots,i_N}$. Η εφαρμογή της *ανάλυσης ιδιαζουσών τιμών (SVD)* για την *παραγοντοποίηση τανυστών* [Lathauwer et al., 2000] γενικεύει την εφαρμογή της *SVD* σε απλούς δισδιάστατους πίνακες. Η δε εφαρμογή της *Higher Order SVD* σ' έναν *τανυστή* $\mathcal{A}$ 3ης-τάξης δημιουργεί τρία *matrix unfoldings*[1] αυτού. Οι πράξεις ορίζονται ως εξής [Lathauwer et al., 2000]:

$$A_1 \in \mathbb{R}^{I_1 \times (I_2 I_3)}, \qquad A_2 \in \mathbb{R}^{I_2 \times (I_1 I_3)}, \qquad A_3 \in \mathbb{R}^{(I_1 I_2) \times I_3} \tag{5.31}$$

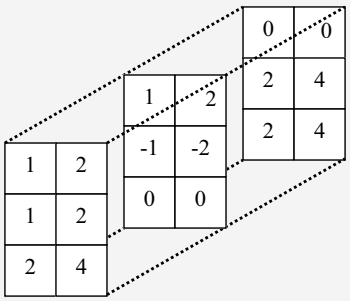
όπου A_1, A_2, A_3 ονομάζεται ο mode-1, ο mode-2, και ο mode-3 τρόπος ξεδιπλώματος σε δισδιάστατο πίνακα του *τανυστή* $\mathcal{A}$ αντίστοιχα - και ο οποίος τρόπος απεικονίζεται στο Σχήμα 5.16. Στη συνέχεια θα παρουσιάσουμε ένα παράδειγμα παραγοντοποίησης *τανυστή* που υιοθετήθηκε από την ακόλουθη ερευνητική εργασία, [Lathauwer et al., 2000]:



Σχήμα 5.16: Οπτική απεικόνιση των τριών τρόπων ξεδιπλώματος ενός *τανυστή* 3ης τάξης.

[1] Ορίζουμε ως *matrix unfolding ενός τανυστή* την αναπαράστασή του υπό τη μορφή απλού δισδιάστατου πίνακα όπου όλα τα διανύσματα του *τανυστή* (είτε στήλης είτε γραμμής) στοιβάζονται το ένα μετά το άλλο με κατάλληλο τρόπο ώστε να σχηματίσουν έναν δισδιάστατο πίνακα.

Παράδειγμα 5.4 Έστω ότι έχουμε έναν *τανυστή* $\mathcal{A} \in \mathbb{R}^{3\times2\times3}$ με $a_{1,1,1} = a_{1,1,2} = a_{2,1,1} = -a_{2,1,2} = 1, a_{2,1,3} = a_{3,1,1} = a_{3,1,3} = a_{1,2,1} = a_{1,2,2} = a_{2,2,1} = -a_{2,2,2} = 2, a_{2,2,3} = a_{3,2,1} = a_{3,2,3} = 4, a_{1,1,3} = a_{3,1,2} = a_{1,2,3} = a_{3,2,2} = 0$. Ο *τανυστής* και το "ξεδίπλωμά" του σε mode-1 πίνακα $A_1 \in \mathbb{R}^{I_1\times I_2 I_3}$ απεικονίζονται στο Σχήμα 5.17:



$$A_1 = \begin{bmatrix} 1 & 1 & 0 & 2 & 2 & 0 \\ 1 & -1 & 2 & 2 & -2 & 4 \\ 2 & 0 & 2 & 4 & 0 & 4 \end{bmatrix}$$

Σχήμα 5.17: Οπτική απεικόνιση του *τανυστή* $\mathcal{A}$ $\mathbb{R}^{3\times2\times3}$ και του mode-1 *ξεδιπλώματός* του σε δισδιάστατο πίνακα.

Ακολούθως, ορίζουμε την πράξη του γινόμενου mode-n ενός *τανυστή* N-τάξης $\mathcal{A} \in \mathbb{R}^{I_1\times\cdots\times I_N}$ μ' έναν πίνακα $U \in \mathbb{R}^{J_n\times I_n}$, η οποία συμβολίζεται ως $\mathcal{A} \times_n U$. Το αποτέλεσμα του γινόμενου mode-n είναι ένας *τανυστής* διαστάσεων $(I_1 \times I_2 \times \cdots \times I_{n-1} \times J_n \times I_{n+1} \times \cdots \times I_N)$, τα στοιχεία του οποίου ορίζονται ως εξής:

$$(\mathcal{A} \times_n U)_{i_1 i_2 \ldots i_{n-1} j_n i_{n+1} \ldots i_N} = \sum_{i_n} a_{i_1 i_2 \ldots i_{n-1} i_n i_{n+1} \ldots i_N} u_{j_n, i_n} \tag{5.32}$$

Δεδομένου ότι εστιάζουμε σε *τανυστές* 3ης τάξης όπου $n \in \{1, 2, 3\}$, χρησιμοποιούμε πράξεις γινόμενου τύπου mode-1, -2 και -3. Όσον δε αφορά στις πράξεις γινόμενου mode-n, η *SVD* σ' έναν κανονικό δισδιάστατο πίνακα (*τανυστής* δευτέρας τάξης) μπορεί να ξαναγραφεί ως εξής [Lathauwer et al., 2000]:

$$F = S \times_1 U^{(1)} \times_2 U^{(2)} \tag{5.33}$$

όπου $U^{(1)} = (u_1^{(1)} u_2^{(1)} \ldots u_{I_1}^{(1)})$ ένας *μοναδιαίος* $(I_1 \times I_1)$-πίνακας [2], $U^{(2)} = (u_1^{(2)} u_2^{(2)} \ldots u_{I_1}^{(2)})$ ένας *μοναδιαίος* $(I_2 \times I_2)$-matrix, και S ένας $(I_1 \times I_2)$-πίνακας με τις παρακάτω ιδιότητες:

1. Ψευδοδιαγωνιότητα: $S = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_{\min\{I_1, I_2\}})$,
2. Διάταξη στοιχείων: $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min\{I_1, I_2\}} \geq 0$.

[2] Ένας πίνακας $n \times n$ U λέγεται *μοναδιαίος* όταν τα διανύσματα στήλης του σχηματίζουν ένα ορθοκανονικό σύνολο στον μιγαδικό χώρο εσωτερικού γινόμενου $\mathbb{C}^n$. Δηλαδή, $U^T U = I_n$.

Επεκτείνοντας την εφαρμογή της *SVD* της προηγούμενης εξίσωσης, η *HOSVD* ενός *τανυστή* τρίτης τάξης $\mathcal{A}$ μπορεί να γραφεί ως εξής [Lathauwer et al., 2000]:

$$\mathcal{A} = S \times_1 U^{(1)} \times_2 U^{(2)} \times_3 U^{(3)} \tag{5.34}$$

όπου $U^{(1)}$, $U^{(2)}$, $U^{(3)}$ περιέχουν τα ορθοκανονικά διανύσματα (τα οποία ονομάζονται *ιδιάζοντα διανύσματα* mode-1, mode-2 και mode-3 αντίστοιχα) και σχετίζονται με το "ξεδίπλωμα" του *τανυστή* στους τρεις δισδιάστατους πίνακες A_1, A_2, A_3 αντίστοιχα. Ο S καλείται *πυρήνας-τανυστής* και έχει την ιδιότητα της "πλήρους ορθογωνιότητας" (για περισσότερες πληροφορίες βλ. [Lathauwer et al., 2000]). Αυτή η *αποσύνθεση τανυστή* αναφέρεται επίσης σ' ένα γενικό μοντέλο παραγοντοποίησης, είναι γνωστή ως *Tucker decomposition* [Tucker, 1966], και θα την αναλύσουμε περαιτέρω στη συνέχεια.

5.6 Αποσύνθεση Tucker και HOSVD

Η *Αποσύνθεση Tucker (TD)* προτάθηκε από τον Tucker το 1966 (βλ. [Tucker, 1966]). Η δε *μέθοδος αποσύνθεσης Tucker I* είναι μια σημαντική παραλλαγή της, η οποία αργότερα έγινε γνωστή ως *HOSVD* [Lathauwer et al., 2000]. Η *HOSVD* αποσυνθέτει έναν *τανυστή* σε (α) ένα σύνολο πινάκων και (β) έναν μικρό *πυρήνα (core) τανυστή*. Εδώ αναλύουμε τον τρόπο με τον οποίο η μέθοδος *HOSVD* μπορεί να χρησιμοποιηθεί για την *παραγοντοποίηση τανυστών* σε *συστήματα κοινωνικής επισήμανσης* (*Social Tagging Systems*), τα οποία χρησιμοποιούνται προκειμένου οι χρήστες να κάνουν επικόλληση ετικετών πάνω σε πληροφοριακά στοιχεία (π.χ. φωτογραφίες, βίντεο κ.λπ.), ώστε επισημαίνοντάς τα να βρίσκουν εύκολα στη συνέχεια αυτά τα στοιχεία, λαμβάνοντας πάντα υπόψη ότι οι φωτογραφίες και τα βίντεο δεν διαθέτουν επαρκή *meta data* (π.χ. κείμενο) που να τα περιγράφουν.

Η τριμερής σχέση των χρηστών, των στοιχείων και των ετικετών στα *Social Tagging Systems* μπορεί να αναπαρασταθεί μ' έναν τρίτης τάξης *τανυστή* $\mathcal{A}$. Στη συνέχεια, είναι δυνατόν να εφαρμοστούν επί του *τανυστή* τεχνικές παραγοντοποίησης και ανάλυσής του για την αξιοποίηση της *λανθάνουσας σημασιολογικής πληροφορίας* που βρίσκεται στον $\mathcal{A}$ αλλά δεν είναι εξαρχής εμφανής. Η ιδέα του υπολογισμού χαμηλής τάξης *τανυστών* έχει ήδη χρησιμοποιηθεί για πολλούς διαφορετικούς σκοπούς ([Lathauwer et al., 2000], [Shashua and Hazan, 2005], [Kolda and Sun, 2008], [Wang and Ahuja, 2007], [Chen et al., 2007], [Sun et al., 2005]). Το σκεπτικό πίσω απ' αυτήν αφορά στη μετατροπή του προβλήματος της σύστασης στοιχείων σε πρόβλημα συμπλήρωσης των στοιχείων του *τανυστή* τρίτης τάξης με στόχο τη συμπλήρωση των *μη παρατηρούμενων (non-observable)* καταχωρίσεών του ή αλλιώς τη συμπλήρωση των *ελλειπουσών τιμών (missing values)* του.

Τυπικά, ένα *σύστημα κοινωνικής επισήμανσης* ορίζεται ως μια *σχεσιακή δομή* $\mathbb{F} := (U, I, T, Y)$ όπου

- U, I και T είναι ξεχωριστά, μη κενά πεπερασμένα σύνολα, τα μέλη των οποίων ονομάζονται *χρήστες, στοιχεία* και *ετικέτες* αντίστοιχα, και
- Y είναι το σύνολο των παρατηρούμενων τριμερών σχέσεων μεταξύ τους, $Y \subseteq U \times I \times T$, τα μέλη του οποίου ονομάζονται *αναθέσεις ετικετών επί των στοιχείων.*
- Το σύνολο των αναθέσεων *ετικετών* ενός *χρήστη* για ένα συγκεκριμένο *στοιχείο* ορίζεται με τη βοήθεια τριπλετών της μορφής $(u, i, T_{u,i})$, όπου ο *χρήστης* $u \in U$, το *στοιχείο* $i \in I$, και υπάρχει και ένα μη κενό σύνολο $T_{u,i} := \{t \in T \mid (u, i, t) \in Y\}$.

Το Y που εκφράζει την τριμερή σχέση *χρηστών, στοιχείων* και *ετικετών* μπορεί να απεικονιστεί από έναν τρίτης τάξεως δυαδικό *τανυστή* $\mathcal{A} = (a_{u,i,t}) \in \mathbb{R}^{|U|\times|I|\times|T|}$ όπου το 1 υποδηλώνει παρατηρημένες αναθέσεις *ετικετών* από *χρήστες* επί των *στοιχείων* και το 0 *ελλείπουσες τιμές*:

$$a_{u,i,t} := \begin{cases} 1, & (u,i,t) \in Y \\ 0, & \text{αλλιώς} \end{cases}$$

Εν συνεχεία, εκφράζουμε την *αποσύνθεση* ενός *τρίτης τάξης τανυστή* $\mathcal{A}$ ως εξής:

$$\hat{\mathcal{A}} := \hat{C} \times_u \hat{U} \times_i \hat{I} \times_t \hat{T} \tag{5.35}$$

όπου $\hat{U}$, $\hat{I}$, και $\hat{T}$ δισδιάστατοι πίνακες που αναπαριστούν έναν διαφορετικό τρόπο "ξεδιπλώματος" του τανυστή (*χρήστης, στοιχεία,* και *ετικέτες* αντίστοιχα) ως προς ένα συγκεκριμένο μέγεθος του λανθάνοντος διανυσματικού τους χώρου k_U, k_I, k_T αντίστοιχα, ενώ $\hat{C} \in \mathbb{R}^{k_U \times k_I \times k_T}$ *τανυστής* (δηλαδή ένας τρισδιάστατος πίνακας) που κρατά τις αλληλεπιδράσεις μεταξύ των δισδιάστατων πινάκων. Οι προς βελτιστοποίηση παράμετροι του μοντέλου αναπαρίστανται από την τετράδα $\hat{\theta} := (\hat{C}, \hat{U}, \hat{I}, \hat{T})$, (βλ. Σχήμα 5.18).

Η βασική ιδέα του *αλγορίθμου HOSVD* είναι η ελαχιστοποίηση της διαφοράς μεταξύ των στοιχείων του *προβλεπόμενου τανυστή* $\hat{\mathcal{A}}$ και του *αρχικού τανυστή* $\mathcal{A}$:

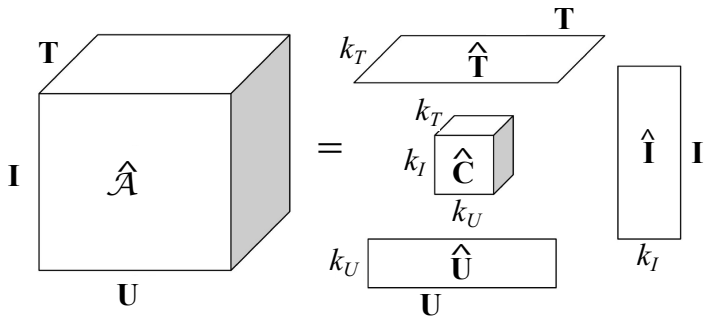
$$\arg\min_{\hat{\theta}} \sum_{(u,i,t)\in Y} (\hat{a}_{u,i,t} - a_{u,i,t})^2$$

Αφού λοιπόν βελτιστοποιηθούν οι παράμετροι, οι προβλέψεις μπορούν να γίνουν ως εξής:

$$\hat{a}(u,i,t) := \sum_{\tilde{u}=1}^{k_U} \sum_{\tilde{i}=1}^{k_I} \sum_{\tilde{t}=1}^{k_T} \hat{c}_{\tilde{u},\tilde{i},\tilde{t}} \cdot \hat{u}_{u,\tilde{u}} \cdot \hat{i}_{i,\tilde{i}} \cdot \hat{t}_{t,\tilde{t}} \tag{5.36}$$

όπου $\hat{U} = [\hat{u}_{u,\tilde{u}}]^{u=1,\ldots,U}_{\tilde{u}=1,\ldots,k_U}$, $\hat{I} = [\hat{i}_{i,\tilde{i}}]^{i=1,\ldots,I}_{\tilde{i}=1,\ldots,k_I}$, $\hat{T} = [\hat{t}_{t,\tilde{t}}]^{t=1,\ldots,T}_{\tilde{t}=1,\ldots,k_T}$, και όπου ο δείκτης πάνω από την κάθε λανθάνουσα διάσταση ενός δισδιάστατου *πίνακα/τανυστή* αναπαρίσταται με το *σύμβολο tilde* $\tilde{\cdot}$.

Παρατηρήστε στο Σχήμα 5.18 ότι ο βασικός λόγος για την υψηλή (κυβική) πολυπλοκότητα ($O(k^3)$ με $k := \min(k_U, k_I, k_T)$) της *μεθόδου HOSVD* είναι ο *πυρήνας (core) τανυστής* $\hat{C}$.



Σχήμα 5.18: *Αποσύνθεση τανυστή* στα *Social Tagging Systems*. Το Σχήμα έχει προσαρμοστεί βάσει της ακόλουθης μελέτης, [Rendle et al., 2009b].

5.7 Αλγόριθμος AlsHOSVD

Ο *προβλεπόμενος τανυστής* της προηγούμενης υποενότητας (γνωστός και ως *truncated HOSVD*) δεν είναι ο βέλτιστος, αλλά αποτελεί ένα καλό σημείο εκκίνησης για έναν *επαναληπτικό αλγόριθμο εναλλασσόμενων ελαχίστων τετραγώνων (ALS)* με στόχο την καλύτερη δυνατή σύγκλιση του *προβλεπόμενου τανυστή* έναντι του *αρχικού τανυστή* [Kolda and Bader, 2009]. Η βασική ιδέα του *αλγορίθμου AlsHOSVD* συνίσταται στο ότι προσπαθεί να ελαχιστοποιήσει το σφάλμα μεταξύ των αρχικών και των προβλεπόμενων τιμών του *τανυστή*. Ο ψευδοκώδικας της *μεθόδου AlsHOSVD* απεικονίζεται παρακάτω, στον Αλγόριθμο 2.

Αλγόριθμος 2 AlsHOSVD

Είσοδος: Ο *αρχικός τανυστής* $\mathcal{A}$ με τις διαστάσεις *χρήστη*, *στοιχείου*, και *ετικέτας*.

Έξοδος: Ο *ανακατασκευασμένος τανυστής* $\hat{\mathcal{A}}$ με k_U, k_I και k_T τα αριστερά κορυφαία *ιδιοδιανύσματα* κάθε διάστασης αντίστοιχα.

1: Αρχικοποίηση του *τανυστή πυρήνα (core tensor)* $\mathcal{C}$ και των αριστερών *ιδιοδιανυσμάτων* $U^{(1)}, U^{(2)}, U^{(3)}$ των δισδιάστατων πινάκων A_1, A_2 και A_3 αντίστοιχα.

2: **Επανάλαβε**:

$$\mathcal{C} = \mathcal{A} \times_1 {U_{k_U}}^{(1)^T} \times_2 U_{k_I}^{(2)^T} \times_3 {U_{k_T}}^{(3)^T}$$

$$\hat{\mathcal{A}} = \mathcal{C} \times_1 {U_{k_U}}^{(1)} \times_2 {U_{k_I}}^{(2)} \times_3 {U_{k_T}}^{(3)}$$

$$U_{k_U}^{(1)} \leftarrow \text{κορυφαία } k_U \text{ αριστερά } \textit{ιδιοδιανύσματα} \text{ του } A_1$$

$$U_{k_I}^{(2)} \leftarrow \text{κορυφαία } k_I \text{ αριστερά } \textit{ιδιοδιανύσματα} \text{ του } A_2$$

$$U_{k_T}^{(3)} \leftarrow \text{κορυφαία } k_T \text{ αριστερά } \textit{ιδιοδιανύσματα} \text{ του } A_3$$

3: **μέχρις ότου** το $\|\mathcal{A} - \hat{\mathcal{A}}\|^{\in}$ παύσει να βελτιώνεται **ή** μέχρι να φτάσουμε έναν max (μέγιστο) αριθμό επαναλήψεων.

4: **Επίστρεψε** $\mathcal{C}$, ${U_{k_U}}^{(1)}$, ${U_{k_I}}^{(2)}$, και ${U_{k_T}}^{(3)}$

Όπως φαίνεται στη γραμμή 3 του αλγορίθμου 2, ο *AlsHOSVD* ελαχιστοποιεί μια *αντικειμενική συνάρτηση* που υπολογίζει το σφάλμα μεταξύ των πραγματικών και των προβλεπόμενων τιμών του *αρχικού* και του *ανακατασκευασμένου τανυστή*. Αυτό γίνεται επαναληπτικά μέχρι η *αντικειμενική συνάρτηση* να πάψει να βελτιώνεται ή να επιτευχθεί ο μέγιστος αριθμός των επαναλήψεων που έχει οριστεί. Τονίζουμε ότι οι τιμές των κορυφαίων k αριστερών μοναδιαίων *διανυσμάτων* των τριών πινάκων (γραμμή 2) αυξάνονται σταδιακά σε κάθε επανάληψη του αλγορίθμου 2.

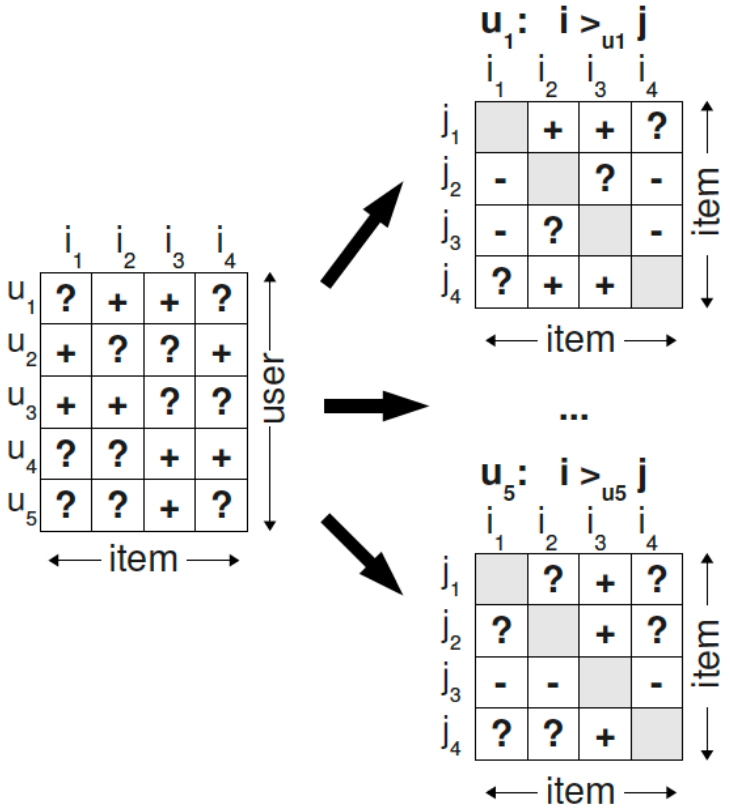
5.8 Αλγόριθμος Bayesian Pairwise Ranking

Στις προηγούμενες ενότητες του παρόντος κεφαλαίου περιγράψαμε την τεχνική συμπλήρωσης των *ελλειπουσών τιμών* του πίνακα βαθμολογιών χρήστη-στοιχείου. Παρόλο που η τεχνική της μείωσης της διαφοράς μεταξύ προβλεπόμενων και πραγματικών τιμών του πίνακα βαθμολογιών χρήστη-στοιχείου είναι σημαντική, δεν βελτιστοποιεί άμεσα τις παραμέτρους του μοντέλου για την τελική κατάταξη των στοιχείων που θα προταθούν στον χρήστη. Αντ' αυτού, η μέθοδος αυτή εκπαιδεύει ένα μοντέλο για να *προβλέπει σωστά τη βαθμολογία ενός χρήστη για ένα στοιχείο* (*rating prediction task*). Θα εξετάσουμε, λοιπόν, τη δημιουργία ενός συστήματος συστάσεων χρησιμοποιώντας απευθείας τη μέθοδο της *εξατομικευμένης κατάταξης κατά Bayes* (*Bayesian Personalized Ranking*) [Rendle et al., 2009a], η οποία εκπαιδεύει ένα μοντέλο -το οποίο *μαθαίνει- να κατατάσσει σωστά τα στοιχεία* (*learning to rank task*) βάσει των προτιμήσεων του υπό εξέταση χρήστη.

Υπογραμμίζουμε εδώ ότι συχνά δεν διαθέτουμε τις βαθμολογίες των χρηστών επί των στοιχείων (για παράδειγμα, μπορεί να έχουμε μόνο την πληροφορία ότι ένας χρήστης διάβασε ή αλληλεπίδρασε μ' ένα άρθρο μιας ηλεκτρονικής εφημερίδας). Έτσι, δεν μπορούμε να γνωρίζουμε εάν το συγκεκριμένο στοιχείο άρεσε στον χρήστη ή όχι. Με άλλα λόγια, εάν υποθέσουμε ότι τα στοιχεία με τα οποία αλληλεπίδρασε ένας χρήστης είναι "θετικά" παραδείγματα (του αρέσουν δηλαδή), δεν μπορούμε να ισχυριστούμε ότι όλα τα υπόλοιπα στοιχεία της βάσης δεδομένων μας, με τα οποία δεν αλληλεπίδρασε ο υπό εξέταση χρήστης, αποτελούν "αρνητικά" παραδείγματα (δηλαδή δεν του αρέσουν). Για την επίλυση αυτού του προβλήματος η μέθοδος του *Bayesian Personalized Ranking (BPR)* επιτρέπει την κατάταξη ενός στοιχείου με το οποίο έχει αλληλεπιδράσει ένας χρήστης σε καλύτερη θέση στη λίστα συστάσεων από ό,τι ένα στοιχείο με το οποίο δεν έχει αλληλεπιδράσει.

Για να εκπαιδεύσουμε επομένως ένα μοντέλο βασισμένο στην *εξατομικευμένη κατάταξη κατά Bayes*, πρέπει να ορίσουμε *τριπλέτες* (*χρήστης, θετικό στοιχείο, αρνητικό στοιχείο*) που θα καταγράφουν τις προτιμήσεις ενός χρήστη με το σκεπτικό που αναλύσαμε παραπάνω. Λόγου χάρη, στην αριστερή πλευρά του Σχήματος 5.19 εμφανίζονται τα *παρατηρούμενα (observable)* δεδομένα του πίνακα βαθμολογιών χρήστη-στοιχείου S, όπου το συν (+) σημαίνει ότι ο χρήστης αλληλεπίδρασε μ' ένα στοιχείο, ενώ το αγγλικό ερωτηματικό (?) σημαίνει ότι ο χρήστης δεν αλληλεπίδρασε μ' αυτό. Η *μέθοδος BPR* δημιουργεί για κάθε υπό εξέταση χρήστη u ζεύγη προτιμήσεων της μορφής $i >_u j$ μεταξύ ενός ζεύγους στοιχείων, όπου το στοιχείο i προτιμάται από τον χρήστη u έναντι του στοι-

χείου j. Το συν (+) στο δεξιό μέρος του Σχήματος 5.19 υποδηλώνει ότι ένας χρήστης προτιμά το στοιχείο i έναντι του στοιχείου j, ενώ το μείον (-) υποδηλώνει ότι προτιμά το j έναντι του i.



Σχήμα 5.19: Το Σχήμα προσαρμόστηκε βάσει της εργασίας του Rendle και των συνεργατών του [Rendle et al., 2009a].

Επειδή στο πρόβλημά μας αυτό της βελτιστοποίησης έχουμε *τριπλέτες (χρήστης, θετικό στοιχείο i, αρνητικό στοιχείο j)* της μορφής $(u,i,j) \in D_S$, πρώτα αναλύουμε την προβλεπόμενη βαθμολογία $\hat{x}_{uij}$ και την ορίζουμε ως εξής:

$$\hat{x}_{uij} := \hat{x}_{ui} - \hat{x}_{uj}$$

όπου $\hat{x}_{ui}$ η προβλεπόμενη βαθμολογία του χρήστη u για το αντικείμενο i με το οποίο έχει αλληλεπιδράσει στο παρελθόν - και θα πρέπει να το προβλέπουμε με υψηλότερη βαθμολογία έναντι του στοιχείου j με το οποίο ο χρήστης δεν έχει αλληλεπιδράσει στο παρελθόν. Μ' άλλα λόγια, θα πρέπει να προβλέπουμε πάντα το $\hat{x}_{uij}$ να έχει τιμή θετική (μεγαλύτερη του μηδενός).

Έπειτα, ορίζουμε την πιθανότητα ένας χρήστης να προτιμά πραγματικά το στοιχείο i από το στοιχείο j ως εξής:

$$p(i >_u j \mid \Theta) := \sigma\left(\hat{x}_{uij}(\Theta)\right)$$

όπου σ η *logistic sigmoid function*:

$$\sigma(x) := \frac{1}{1+e^{-x}}$$

Επίσης, η $\hat{x}_{uij}(\Theta)$ είναι μια συνάρτηση του διανύσματος παραμέτρου ενός μοντέλου Θ που αποτυπώνει την ειδική σχέση μεταξύ του χρήστη u, του στοιχείου i και του στοιχείου j. Η τελική *αντικειμενική συνάρτηση* που θέλουμε να βελτιστοποιήσουμε είναι η παρακάτω:

$$\sum_{(u,i,j)\in D_s} \sigma\left(x_{uij}, \hat{x}_{uij}\right) + \lambda \cdot \|\Theta\|$$

όπου $x_{uij} = x_{ui} - x_{uj}, \hat{x}_{uij} = \hat{x}_{ui} - \hat{x}_{uj}$, και D_s ένα σύνολο από τριπλέτες, καθεμία εκ των οποίων περιλαμβάνει έναν χρήστη u και ένα ζεύγος στοιχείων i και j, όπου συνήθως i είναι ένα θετικό στοιχείο και j ένα αρνητικό.

Τόσο για τις *αντικειμενικές συναρτήσεις κατά ζεύγη (pairwise)* όσο και για τις *αντικειμενικές συναρτήσεις κατά σημεία (point-wise)*, η επιλογή της l*oss function* είναι σημαντική (π.χ *square loss, cross-entropy loss* κ.λπ.). Παρακάτω παραθέτουμε τη *συνάρτηση cross-entropy* για την τεχνική *BPR*:

$$\sigma(x, \hat{x}) = -x\log(p) - (1-x)\log(1-p), \text{ όπου } p = \sigma(\hat{x}) = 1/(1+\exp(-\hat{x}))$$

όπου $\hat{x}$ η προβλεπόμενη βαθμολογία, ενώ x η πραγματική βαθμολογία που παρατηρούμε στα *δεδομένα εκπαίδευσης*.

5.9 Αλγόριθμος List-wise Ranking

Οι αλγόριθμοι κατάταξης των προτεινόμενων στοιχείων μιας λίστας συστάσεων διακρίνονται σε *point-wise* [Bhargava et al., 2015], *pair-wise* [Rendle et al., 2009a], και *list-wise* [Wu et al., 2016].

Οι αλγόριθμοι κατάταξης βάσει της πρόβλεψης της βαθμολογίας ενός στοιχείου (point-wise ranking) περιγράφηκαν εκτενώς στις πρώτες ενότητες του παρόντος κεφαλαίου. Πιο συγκεκριμένα, οι *point-wise* προσεγγίσεις προβλέπουν βαθμολογίες για μεμονωμένα στοιχεία και μπορούν έτσι να θεωρηθούν εννοιολογικά ισοδύναμες με το *πρόβλημα της πρόβλεψης της βαθμολογίας του χρήστη για ένα στοιχείο* όπως αυτό ορίζεται στο *συνεργατικό φιλτράρισμα*. Κατόπιν, τα στοιχεία κατατάσσονται σύμφωνα μ' αυτές τις προβλεπόμενες βαθμολογίες, με αποτέλεσμα να δημιουργείται μια *διατεταγμένη λίστα προτεινόμενων στοιχείων*.

Από την άλλη, οι *τεχνικές κατάταξης κατά ζεύγη στοιχείων (pair-wise ranking)* κατατάσσουν ένα πλήθος ζευγών στοιχείων κάθε φορά, και παρουσιάστηκαν λεπτομερώς στην προηγούμενη ενότητα αυτού του κεφαλαίου με αντιπροσωπευτική αυτών μέθοδο τον *αλγόριθμο BPR*.

Όσον δε αφορά στους *listwise αλγορίθμους* ([Weimer et al., 2007], [Shi et al., 2010]), η ποιότητα ολόκληρης της λίστας προτεινόμενων στοιχείων αξιολογείται συνολικά με τη χρήση μιας *αντικειμενικής συνάρτησης μείωσης σφάλματος (loss function)* που λαμβάνει υπόψη τη θέση κατάταξης των προτεινόμενων στοιχείων μέσα στη λίστα συστάσεων. Οι *αντικειμενικές συναρτήσεις μείωσης σφάλματος* ορίζονται με τέτοιον τρόπο ώστε να αντικατοπτρίζουν την απόσταση μεταξύ της *λίστας των προτεινόμενων στοιχείων βάσει των πραγματικών προτιμήσεων του χρήστη* -όπως αποτυπώνονται στα

δεδομένα εκπαίδευσης- και της *λίστας των προτεινόμενων στοιχείων που προβλέπει το μοντέλο πρόβλεψης της κατάταξής τους.*

Ένα παράδειγμα τέτοιων *αντικειμενικών συναρτήσεων* είναι η *nDCG* (*normalized Discounted Cumulative Gain*), η οποία αποτελεί μέτρο αποτίμησης της ποιότητας της λίστας των προτεινόμενων στοιχείων, και θα παρουσιαστεί εκτενώς στο 9ο κεφάλαιο. Ειδικότερα, μπορούμε να επαναπροσδιορίσουμε τη διατεταγμένη λίστα προτεινόμενων στοιχείων βάσει των τιμών του *μέτρου αποτίμησης (nDCG)* της λίστας αυτής. Επομένως, η βασική ιδέα του *list-wise ranking* στηρίζεται στο να επινοηθεί ένα μοντέλο βελτιστοποίησης που να μπορεί να επαναπροσδιορίσει σε κάθε βήμα του αλγορίθμου την κατάλληλη μετάθεση των στοιχείων της λίστας συστάσεων βάσει των τιμών του *μέτρου αποτίμησης (nDCG).*

Οι *αλγόριθμοι listwise ranking* είναι οι πιο αποτελεσματικοί ως προς την *ακρίβεια* πρόβλεψης των προτιμήσεων του υπό εξέταση χρήστη συγκριτικά με τους *pairwise* και *pointwise* αλγορίθμους. Ο κύριος λόγος οφείλεται στο γεγονός ότι εξετάζουν συνολικά ολόκληρη τη λίστα των στοιχείων που κατατάσσονται ως προς την *αντικειμενική συνάρτηση μείωσης σφάλματος (loss function).* Ωστόσο, επειδή οι *αλγόριθμοι κατάταξης κατά λίστα (listwise)* είναι πολύ *υψηλών υπολογιστικών απαιτήσεων (high time complexity),* οι προσεγγίσεις *κατάταξης κατά ζεύγη* (π.χ. *αλγόριθμος BPR*) υιοθετούνται ευρέως από τα συστήματα συστάσεων, καθώς αποτελούν την καλύτερη *αντιστάθμιση* μεταξύ του *υπολογιστικού κόστους* και της *αποτελεσματικότητας* ως προς τη σωστή κατάταξη των προτεινόμενων στοιχείων σε σχέση με τις προτιμήσεις του υπό εξέταση χρήστη [Ying et al., 2016].

5.10 Ερωτήσεις Κεφαλαίου

1. Σε τί διαφέρει η *αποσύνθεση ιδιοτιμών πίνακα* από την *αποσύνθεση ιδιαζουσών τιμών πίνακα*; Επίσης, σε τί διαφέρει η *αποσύνθεση ιδιαζουσών τιμών πίνακα* από την *αποσύνθεση πίνακα κατά UV*;

2. Τί γνωρίζετε για το φαινόμενο της *υπερπροσαρμογής (overfitting)* για το *μοντέλο πρόβλεψης βάσει της αποσύνθεσης πίνακα κατά UV*; Δώστε ένα παράδειγμα.

3. Στο πρόβλημα *πρόβλεψης/συμπλήρωσης βαθμολογιών ενός πίνακα* χρήστη–στοιχείου *(matrix completion problem)* ορίζεται μια *αντικειμενική συνάρτηση* με την οποία επιδιώκουμε:

   Α) Να μεγιστοποιήσουμε τη διαφορά μεταξύ της *πραγματικής* και της *προβλεπόμενης* βαθμολογίας των χρηστών για τα στοιχεία.

   Β) Να μεγιστοποιήσουμε τη διαφορά (ως προς την κατάταξη των συστηνόμενων στοιχείων που προτείνονται στον υπό εξέταση χρήστη) μεταξύ της *προβλεπόμενης* και της *πραγματικής* λίστας στοιχείων που είναι συναφή με τον υπό εξέταση χρήστη.

   Γ) Να ελαχιστοποιήσουμε τη διαφορά μεταξύ της *πραγματικής* και της *προβλεπόμενης* βαθμολογίας των χρηστών για τα στοιχεία.

Δ) Να ελαχιστοποιήσουμε τη διαφορά (ως προς την κατάταξη των συστηνόμενων στοιχείων που προτείνονται στον υπό εξέταση χρήστη) μεταξύ της *προβλεπόμενης* και της *πραγματικής* λίστας στοιχείων που είναι συναφή με τον υπό εξέταση χρήστη.

4. *Η μέθοδος Matrix Factorization (MF)* επιτρέπει την παραγοντοποίηση ενός πίνακα A [m × n] που περιέχει m χρήστες και n στοιχεία σε δύο άλλους πίνακες μικρότερων διαστάσεων (U $[m \times k]$ και V $[n \times k]$), έτσι ώστε $A = U * V^T$. Ποια από τις παρακάτω παραδοχές είναι εσφαλμένη:

   Α) Στον πίνακα V, για κάθε στοιχείο του αρχικού πίνακα A έχουμε μια νέα, μικρότερη αναπαράσταση k διαστάσεων.

   Β) Στον πίνακα U, για κάθε χρήστη του αρχικού πίνακα A έχουμε μια νέα, μικρότερη αναπαράσταση k διαστάσεων.

   Γ) Ο πολλαπλασιασμός του $U * V^T$ δημιουργεί τον αρχικό πίνακα Α.

   Δ) Ο πολλαπλασιασμός του $U * V^T$ δημιουργεί έναν πίνακα που προσεγγίζει τον αρχικό πίνακα A.

5. Εφαρμόστε τη *μέθοδο SVD* στους τρεις χρήστες του παραδείγματος του Σχήματος 5.1(α) κρατώντας μόνο μία διάσταση (rank-1) από τα δεδομένα των παραγοντοποιημένων πινάκων. Με βάση την εφαρμογή της *μεθόδου SVD*, ποιες είναι οι προβλεπόμενες βαθμολογίες για την *ελλείπουσα τιμή* του τέταρτου χρήστη; Πώς συγκρίνεται αυτό το αποτέλεσμα με τα αποτελέσματα της παραγοντοποίησης που παρουσιάζονται στο Σχήμα 5.3;

6. Ποιες αλλαγές θα πρέπει να γίνουν στην *αντικειμενική συνάρτηση* για την παραγοντοποίηση του πίνακα βαθμολογιών χρήστη-στοιχείου ώστε να αποφύγουμε την *υπερπροσαρμογή* του μοντέλου πρόβλεψης στα *δεδομένα εκπαίδευσης*;

7. Έστω ότι εφαρμόζουμε τη *μέθοδο SVD* για να *αποσυνθέσουμε* έναν πίνακα βαθμολογιών χρήστη-στοιχείου R. Στη συνέχεια, χρησιμοποιούμε την *αποσύνθεση* του ($\hat{R} = U \cdot \Sigma \cdot V^T$) για την πρόβλεψη των βαθμολογιών των χρηστών επί των στοιχείων μετά τη μείωση της διάστασης των παραγοντοποιημένων πινάκων, όπου $\hat{R}$ ο ανακατασκευασμένος πίνακας που προκύπτει μετά τη μείωση της *διαστατικότητας* σε k μοναδιαίες (singular) τιμές. Σχεδιάστε την *καμπύλη σφάλματος πρόβλεψης* Err μεταξύ πραγματικών και προβλεπόμενων βαθμολογιών ως συνάρτηση του k για το πώς πιστεύετε ότι θα μεταβληθεί το Err με τη χρήση του ανασκευασμένου $\hat{R}$ αντί του αρχικού R.

8. Χρησιμοποιώντας την *αποσύνθεση SVD* του Σχήματος 5.15, υποθέστε ότι ένας νέος χρήστης (user U_5) δίνει βαθμολογία 3 στο στοιχείο I_2 και βαθμολογία 4 στο στοιχείο I_4, δίνοντάς μας μια αναπαράσταση του U_5 στον διανυσματικό χώρο των στοιχείων ως ακολούθως: $[0, 3, 0, 4]$. Βρείτε την αναπαράσταση του U_5 στον 2-σδιάστατο διανυσματικό χώρο. Τί προβλέπει αυτή

η αναπαράσταση σχετικά με το πόσο πολύ θα αρέσουν στον U_5 τα άλλα 2 στοιχεία (I_1 και I_3) που εμφανίζονται στα δεδομένα του παραδείγματός μας;

9. Ποια η διαφορά μεταξύ των *point-wise* και *pair-wise* αλγορίθμων πρόβλεψης;

5.11 Ασκήσεις Προγραμματισμού

Άσκηση 1: Παραγωγή λίστας συστάσεων που προκύπτει από την παραγοντοποίηση του πίνακα βαθμολογιών χρήστη-στοιχείου.

Δίνεται ο πίνακας βαθμολογίας χρήστη-στοιχείου A του Σχήματος 5.20. Όπως απεικονίζεται στο Σχήμα 5.20, I_{1-4} είναι τα στοιχεία και U_{1-4} είναι οι χρήστες. Τα κελιά χωρίς βαθμολογία παρουσιάζονται ως μηδενικά. Εκτελέστε τον *αλγόριθμο παραγοντοποίησης πινάκων UV-decomposition* και βρείτε τον πλησιέστερο *πίνακα προσέγγισής* $\hat{A}$ του αρχικού πίνακα A.

	I_1	I_2	I_3	I_4
U_1	4	1	1	4
U_2	1	4	2	0
U_3	2	1	4	5
U_4	1	4	1	?

Σχήμα 5.20: Πίνακας βαθμολογιών Χρήστη-Στοιχείου A (4×4).

Λύση

Ακολουθεί ο κώδικας που υλοποιεί την *αποσύνθεση κατά UV* με τις παρακάτω τιμές των παραμέτρων εισόδου ($\eta = 0.0002$, $\beta = 0.02$, $steps \approx 2750$).

```
## Μια υλοποίηση της παραγοντοποίησης πινάκων
"""
@INPUT:
    R : ο προς παραγοντοποίηση πίνακας βαθμολογιών χρηστών επί των στοιχείων διάστασης N x M
    U : ένας πίνακας διάστασης N x K με τυχαίες αρχικές τιμές
    V : ένας πίνακας διάστασης M x K με τυχαίες αρχικές τιμές
    K : ο αριθμός των λανθανόντων χαρακτηριστικών
    steps : ο μέγιστος αριθμός βημάτων για την εκτέλεση της βελτιστοποίησης
```

```
    eta : ο ρυθμός μάθησης
    beta : η παράμετρος κανονικοποίησης για την αποφυγή της
   υπερπροσαρμογής του μοντέλου πρόβλεψης στα δεδομένα εκπαίδευσης
@OUTPUT:
    οι τελικοί πίνακες U και V
"""
import numpy

def matrix_factorization(R, U, V, K, steps=2750, eta=0.0002,
   beta=0.02):
    V = V.T
    for step in range(steps):
        for i in range(len(R)):
            for j in range(len(R[i])):
                if R[i][j] >= 0:
                    eij = R[i][j] - numpy.dot(U[i,:],V[:,j])
                    for k in range(K):
                        U[i][k] = U[i][k] + eta * (2 * eij * V[
   k][j] - beta * U[i][k])
                        V[k][j] = V[k][j] + eta * (2 * eij * U[
   i][k] - beta * V[k][j])
        eR = numpy.dot(U,V)
        e = 0
        for i in range(len(R)):
            for j in range(len(R[i])):
                if R[i][j] > 0:
                    e = e + pow(R[i][j] - numpy.dot(U[i,:],V[:,
   j]), 2)
                    for k in range(K):
                        e = e + (beta/2) * ( pow(U[i][k],2) +
   pow(V[k][j],2) )
        if e < 0.001:
            break
    return U, V.T

##################################################
```

```
if __name__ == "__main__":
    R = [
          [4,1,1,4],
          [1,4,2,0],
          [2,1,4,5],
          [4,1,1,0],
         ]

    R = numpy.array(R)

    N = len(R)
    M = len(R[0])
    K = 2

    U= numpy.random.rand(N,K)
    V = numpy.random.rand(M,K)

    nU, nV = matrix_factorization(R, U, V, K)

    print('Εκτύπωση του Αρχικού Πίνακα Βαθμολογίας χρηστών επί των στοιχείων ')
    print (" ")
    print (R)
    print (" ")

    print('Εκτύπωση του Πίνακα U που κρατά τα διανύσματα των χρηστών ')
    print (" ")
    print (P)
    print (" ")

    print('Εκτύπωση του Πίνακα Q^T που κρατά τα διανύσματα των στοιχείων ')
    print (" ")
    print (Q.T)
    print (" ")
```

```
    P = numpy.asmatrix(P)
    l = Q.T
    l = numpy.asmatrix(l)
    R = P * l

    print('Εκτύπωση του Προσεγγιστικού Πίνακα Βαθμολογίας χρηστών επί των στοιχείων ')
    print (" ")
    print (R)
```

Η εκτύπωση των αποτελεσμάτων του παραπάνω κώδικα είναι η ακόλουθη:

```

 Εκτύπωση του Αρχικού Πίνακα Βαθμολογίας χρηστών επί των στοιχείων

[[4 1 1 4]
 [1 4 2 0]
 [2 1 4 5]
 [4 1 1 0]]

 Εκτύπωση του Πίνακα U που κρατά τα διανύσματα των χρηστών
[[ 1.58612214  0.938007  ]
 [-0.19515714  1.92430056]
 [ 2.15426592  0.56946985]
 [ 0.27163292  1.43364238]]

 Εκτύπωση του Πίνακα Q^T που κρατά τα διανύσματα των στοιχείων
 [[ 1.06483829 -0.14716005  1.10333664  2.31918693]
 [ 1.37763736  1.57570441  0.76995763  0.04534256]]

  Εκτύπωση του Προσεγγιστικού Πίνακα Βαθμολογίας χρηστών επί των στοιχείων
[[ 2.98119706  1.24460796  2.47225232  3.72104536]
 [ 2.44317756  3.06084822  1.26630588 -0.36535317]
 [ 3.07846777  0.58029428  2.81534819  5.02196657]
 [ 2.26428443  2.21902311  1.40354644  0.69497253]]


```

Παρατηρήστε ότι η προβλεπόμενη βαθμολογία του χρήστη U_4 στο αντικείμενο I_4 είναι κοντά στο 0 (0.6949). Αυτό είναι αναμενόμενο, επειδή οι χρήστες U_2 και U_4 έχουν παρόμοιες βαθμολογίες στα στοιχεία I_1, I_2 και I_3. Έτσι, ο χρήστης U_4 προβλέπεται να έχει παρόμοια βαθμολογία με τον U_2 στο αντικείμενο I_4. Τέλος, τονίζουμε ότι σε κάθε διαφορετική εκτέλεση του αλγορίθμου ο παραπάνω πίνακας που προβλέπουμε κατά προσέγγιση θα έχει συνακόλουθα ελαφρώς διαφορετικές τιμές.

Βιβλιογραφία

H. Bensmail and G. Celeux. Regularized gaussian discriminant analysis through eigenvalue decomposition. *Journal of the American Statistical Association*, 91(436):1743--1748, 1996. 10 .1080/01621459.1996.10476746. URL `http://www.tandfonline.com/doi/abs/10.1080/01621459.1996.10476746`.

M. Berry, S. Dumais, and G. O'Brien. Using linear algebra for intelligent information retrieval. *SIAM Review*, 37(4):573--595, 1994.

P. Bhargava, T. Phan, J. Zhou, and J. Lee. Who, what, when, and where: Multi-dimensional collaborative recommendations using tensor factorization on sparse user-generated data. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, pages 130--140, Republic and Canton of Geneva, Switzerland, 2015. International World Wide Web Conferences Steering Committee. ISBN 978-1-4503-3469-3. URL `http://dl.acm.org/citation.cfm?id=2736277.2741077`.

M. Brand. Incremental singular value decomposition of uncertain data with missing values. In *Proceedings of the 7th European Conference on Computer Vision (ECCV)*, pages 707--720, Copenhagen, Denmark, 2002.

L. Cao. Singular value decomposition applied to digital image processing. *Division of Computing Studies Arizona State University Polytechnic Campus, Mesa, Arizona 85212*, 2006.

S. Chen, F. Wang, and C. Zhang. Simultaneous heterogeneous data clustering based on higher order relationships. In *Proceedings of the Seventh IEEE International Conference on Data Mining Workshops*, ICDMW '07, pages 387--392, Washington, DC, USA, 2007. IEEE Computer Society. ISBN 0-7695-3033-8. http://dx.doi.org/10.1109/ICDMW.2007.94. URL `http://dx.doi.org/10.1109/ICDMW.2007.94`.

T.-J. Chin, K. Schindler, and D. Suter. Incremental kernel svd for face recognition with image sets. In *Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition (FGR 2006)*, pages 461--466. IEEE, 2006.

G. W. Furnas, S. Deerwester, S. T. Durnais, T. K. Landauer, R. A. Harshman, L. A. Streeter, and K. E. Lochbaum. Information retrieval using a singular value decomposition model of latent semantic structure. In *Proceedings ACM SIGIR Conference*, pages 465--480, 1988.

G. H. Golub and C. F. Van Loan. *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996. ISBN 0-8018-5414-8.

T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Rev.*, 51(3):455--500, Aug. 2009. ISSN 0036-1445.

T. G. Kolda and J. Sun. Scalable tensor decompositions for multi-aspect data mining. In *ICDM '08: Proceedings of the 8th IEEE International Conference on Data Mining*, pages 363--372. IEEE Computer Society, December 2008.

L. d. Lathauwer, B. d. Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM Journal of Matrix Analysis and Applications*, 21(4):1253--1278, 2000.

C. B. Moler and G. W. Stewart. An algorithm for generalized matrix eigenvalue problems. *SIAM Journal on Numerical Analysis*, 10(2):241--256, 1973. 10.1137/0710024. URL `http://dx.doi.org/10.1137/0710024`.

A. Quarteroni, R. Sacco, and F. Saleri. *Numerical Mathematics*. Texts in applied mathematics. Springer, 2000. ISBN 9780387989594. URL `https://books.google.gr/books?id=YVpyyi1M7vUC`.

A. Rajaraman and J. D. Ullman. *Mining of Massive Datasets*. Cambridge University Press, New York, NY, USA, 2011. ISBN 1107015359, 9781107015357.

S. Rendle, C. Freudenthaler, Z. Gantner, and S.-T. Lars. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 452--461, Arlington, Virginia, United States, 2009a. AUAI Press. ISBN 978-0-9749039-5-8. URL `http://portal.acm.org/citation.cfm?id=1795114.1795167`.

S. Rendle, L. B. Marinho, A. Nanopoulos, and L. Schimdt-Thieme. Learning optimal ranking with tensor factorization for tag recommendation. In *KDD '09: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 727--736. ACM, 2009b. ISBN 978-1-60558-495-9. 10.1145/1557019.1557100.

B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Application of dimensionality reduction in recommender system-a case study. In *ACM WebKDD Workshop*, 2000.

B. Sarwar, J. Konstan, and J. Riedl. Incremental singular value decomposition algorithms for highly scalable recommender systems. In *International Conference on Computer and Information Science*, 2002.

A. Shashua and T. Hazan. Non-negative tensor factorization with applications to statistics and computer vision. In *ICML '05: Proceedings of the 22nd International Conference on Machine Learning*, pages 792--799. ACM, 2005. ISBN 1-59593-180-5. http://doi.acm.org/10.1145/1102351.1102451.

Y. Shi, M. Larson, and A. Hanjalic. List-wise learning to rank with matrix factorization for collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 269--272, 2010.

J. Sun, D. Shen, H. Zeng, Q. Yang, Y. Lu, and Z. Chen. Cubesvd: a novel approach to personalized web search. In *World Wide Web Conference*, pages 382--390, 2005.

L. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, pages 279--311, 1966.

H. Wang and N. Ahuja. A tensor approximation approach to dimensionality reduction. *International Journal of Computer Vision*, pages 217--229, 2007.

M. Weimer, A. Karatzoglou, Q. Le, and A. Smola. Cofirank-maximum margin matrix factorization for collaborative ranking. In *Advances in Neural Information Processing Systems, 21st Annual Conference on Neural Information Processing Systems 2007*, pages 222--230, 2007.

O. Wu, Q. You, X. Mao, F. Xia, F. Yuan, and W. Hu. Listwise learning to rank by exploring structure of objects. *IEEE Transactions on Knowledge and Data Engineering*, 28(7):1934--1939, 2016.

H. Ying, L. Chen, Y. Xiong, and J. Wu. Collaborative deep ranking: A hybrid pair-wise recommendation algorithm with implicit feedback. In *Pacific-asia conference on knowledge discovery and data mining*, pages 555--567. Springer, 2016.