



Експертн системи з додатками

домашня сторінка журналу: www.elsevier.com/locate/eswa



DLSF: система рекомендацій на основ глибокого навчання та семантичного злиття

Вандана Бхатія

Технологічний університет Нетадж Субхас, Дварка, Делі, Індія

ІНФОРМАЦІЯ ПРО СТАТТЮ

Ключові слова:
Система рекомендацій
Глибоке навчання
Спільна фільтрація
Злиття речень
Семантичний аналіз

АНОТАЦІЯ

Інтелектуальна система рекомендацій полегшує користувачам, рекомендуючи товари на основі вподобань користувача. Більшість сучасних систем рекомендацій використовують спільну фільтрацію, яка розглядає лише одні дані, не використовуючи знання. Так підходять до багатьох проблем, таких як проблема холодного запуску, розподіленість, масштабованість. Щоб подолати ці невдачі, у цьому документі пропонується нова система рекомендацій на основі глибокого навчання та семантичного злиття під назвою DLSF, яка об'єднує численні джерела знань, використовуючи три модулі для створення перших рекомендацій. Перший модуль запобігає розподіленості, використовуючи приховані представлення та глибоке навчання для розуміння взаємодій між користувачем і елементом. Другий модуль використовує семантичну інформацію, надану поданими елементами користувачами для вирішення проблеми холодного запуску. Третій модуль працює для нових користувачів і елементів, які не мають попередньої інформації. Комплексна експериментальна оцінка проводиться на основі контрольних наборів даних. Запропонована DLSF показала в середньому на 16,02% і 16,71% більшу ефективність порівняно з базовими системами рекомендацій, врахованими з точки зору MAE і RMSE відповідно. DLSF виявив значну ефективність з точки зору точності, запам'ятовування та загальної

1. Вступ

З удосконаленням ефективності моделей навчання знадобилося аналізувати велику та багатомірну дані, щоб отримати корисну інформацію. За останні десятиліття в Інтернеті доступна величезна кількість інформації, яка зростає експоненціально. Щоб уникнути виснажливого процесу розуміння всієї доступної інформації користувачем, системи рекомендацій виявилися надзвичайно важливими в останні роки (Fu et al., 2019; Lee & Lee, 2019a; Lin et al., 2019). Метою системи рекомендацій є аналіз різного типу інформації для допомоги користувачам у прийнятті правильного рішення. Рекомендацію можна зробити на основі багатьох факторів, таких як характеристики предмета, характеристики користувача, поведінка схожих користувачів (He, 2016; Wei et al., 2017; Zhao, 2017) тощо.

Спільна фільтрація та фільтрація вмісту є одними з успішних підходів до надання рекомендацій. Підходи на основі спільної фільтрації (CF) використовують дані користувача, дані об'єкта в даній проїхній взаємодії. Взаємодія може бути оцінка, яку користувач надає окремому елементу. З іншого боку, підхід, заснований на вмісті, збирає попереднє знання, надане користувачем про елемент, або інформацію про елемент для створення нових рекомендацій. Різниця між обома підходами показана на рис. 1.

Методи на основі спільної фільтрації більш популярні через кращу продуктивність прогнозування. Однак є обидва підходи

певні обмеження. Спільна фільтрація стикається з такими проблемами, як розподіленість даних, масштабованість зменшення популярності. Підходи на основі CF страждають від двох проблем «холодного старту» також через розподіленість даних (He, 2016), як показано на рис. 2. З іншого боку, підходи на основі контенту також мають різні недоліки, такі як втрата провідності, непомірбна специфічність та пригнічення. Щоб подолати ці проблеми, рекомендації можуть бути зроблені шляхом злиття обох підходів на рівні прийняття рішення. Це призведе до полегшення проблеми холодного запуску та провідності. Продаж також є головною проблемою при виборі відповідної системи рекомендацій. Останніми роками доведено, що методи глибокого навчання мають високу масштабованість і досягли величезного успіху в багатьох сферах, таких як обробка тексту (Nishit & Fatma, 2019; Wang, 2015), комп'ютерний зір (Voulodimos et al., 2018), зображення аналіз, мережевий аналіз (V. Bhatia & Rani, 2018b; Bhatia et al., 2017) тощо.

Глибоке навчання може ефективно вирішити проблему масштабованості в системі рекомендацій (Lee & Lee, 2019b; Sivaramakrishnan et al., 2020), оскільки воно може масштабуватися до великої кількості даних.

Щоб подолати вищезазначені проблеми, у цьому документі пропонується модель на основі синтезу речень під назвою DLSF, яка поєднує в собі три основні модулі рекомендацій. Перший модуль реалізує глибоку нейронну мережу прямого зв'язку для спільної фільтрації. Раніше в ньому виконують факторизацію, щоб мати справу з розподіленістю. В ниввічє вбудовування користувачів і елементів, які надаються як вхідні дані для глибокої нейронної мережі. Другий модуль використовує семантичну інформацію для кодування її в контекст

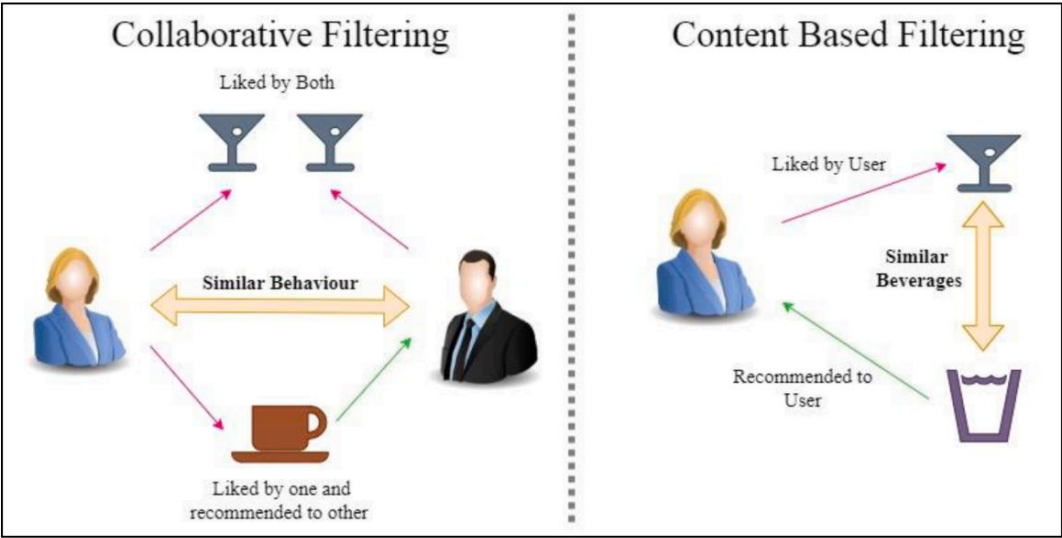


Рис. 1. Зображення рекомендаційних систем.

					New Item
	3	?	1	5	?
	4	5	?	2	?
	?	3	4	?	?
	?	2	?	5	?
New User	?	?	?	?	?

Рис. 2. Ілюстрація холодного запуску та проблеми розрідженості в системі рекомендацій.

базується на низьковимірному векторі та має справу з проблемою дефіцитності. Третій модуль виграє важливу роль у прийнятті рішення, коли користувач новий і не надав жодного рейтингу чи інформації, рекомендуючи найпопулярніші товари, як мають високі оцінки більшості користувачів. Модуль 3 допомагає вирішити проблему холодного запуску. Остаточне рішення щодо рекомендації приймається шляхом рейтингу рекомендацій у цьому модулі. Було проведено кілька експериментів, щоб продемонструвати ефективність запропонованої моделі DLSF у порівнянні з сучасними системами рекомендацій на чотирьох контрольних наборах даних. За всіма розглянутими параметрами DLSF перевершив компетенції інших алгоритмів. Запропоновану модель DLSF можна масштабувати до різних розмірів даних.

Решта паперу організована таким чином. В наступній роботі представлена розділ II. У розділі III представлено роботу запропонованої моделі DLSF. У розділі IV представлено експериментальні результати та оцінку продуктивності. Останній розділ завершує статтю.

2. Сродна робота

Ультратурбу були запропоновані різні підходи до рекомендацій для різних областей, таких як онлайн-магазини (Nishit & Fatma,

2019), пропозиції щодо фільмів (Katarya & Verma, 2017), вибір одягу (R. He, 2016), пропозиції щодо їжі (Mathur та ін., 2019) тощо.

Традиційні системи рекомендацій розглядають лише один даний, тому не можуть спостерігати складні зв'язки між об'єктами. Для роботи з величезною різноманітною інформацією моделі, засновані на термодинамічному синтезі, виявилися надійними, точними та надійними (Hosseini & Chen, 2018; Liu et al., 2019; Strub et al., 2016). Популярні підходи, як використовуються в літературі для рекомендацій, включають методи на основі матриць факторизації (Salakhutdinov & Mnih, 2009), спільну фільтрацію (Katarya & Verma, 2017; Li et al., 2020), фільтрацію на основі контенту (Mathur et al., 2019; K. Wang та ін., 2019), машини факторизації та її варіанти (Kipf & Welling, 2017; Rendle, 2012; Salakhutdinov & Mnih, 2009), логістична регресія (Fang та ін., 2019; Pal & Chakrabarti, 2016), тобто. Деякі методи аналізу частотних шаблонів на основі графів (V. Bhatia & Rani, 2018a) також можна використовувати для системи рекомендацій (Sayeb et al., 2022). Останнім часом багато дослідників в також пропонують моделі на основі глибокого навчання (Fu et al., 2019; Nishit & Fatma, 2019; Wei et al., 2017). Підходи, що базуються на вмісті, надають рекомендації в повільно до подібності елементів. Однак за останні десятиліття найпопулярнішими системами рекомендацій стали методи, засновані на спільній фільтрації (CF) (Fu et al., 2019; Katarya & Verma, 2017; R. He, 2016). Методи на основі спільної фільтрації намагаються передбачити вподобання користувача на основі поведінки користувача в минулому та покладаються на зв'язок між користувачем і елементом, який кодується як матриця оцінок.

Однак підходи на основі CF стикаються з такими проблемами, як розрідженість проблеми холодного запуску. Проблема холодного запуску виникає, коли додаються нові користувачі або нові елементи (Lika та ін., 2014). Для ефективної рекомендації CF вимагає багато оцінок, наданих користувачами для товарів. Для вирішення проблеми холодного запуску в літературі були запропоновані різні гібридні підходи (Strub et al., 2016). Деякі з них використовували гетерогенну інформаційну мережу, формуючи меташляхи та мотогографі замість матричної факторизації (Fang та ін., 2019; Хоссейн та Чен, 2018; Ши та ін., 2018). Більшості з них покладаються на додаткову інформацію, щоб замінити відсутні оцінки. Однак нещодавно багато дослідників в запропонували алгоритми на основі глибокого навчання, які перевершили їх (Bhatia & Rani, 2019; K. Ван та ін., 2019). Алгоритми на основі глибокого навчання використовують спільну фільтрацію (Fu et al., 2019) за вбудовуваннями розглянутих користувачів і елементів. У літературі більшість рекомендаційних систем на основі глибокого навчання використовують неконтрольоване навчання за допомогою Autoencoder (Fu та ін., 2019; Лі та ін., 2020; K. Wang та ін., 2019). Більшість сучасних підходів, заснованих на глибокому навчанні, страждають від надмірного оснащення та мають високу помилку узагальнення. Також було надано деякі масштабовані рішення (Ji & Shen, 2015; Lee & Lee, 2019a, 2019b). Свармакроснавантливості використовували багаті

Таблиця

1 Опис символів.

символ	опис
U	Набір користувачів
\mathcal{Y}	Набір предметів
u_i	i -й користувач у множині U
t_j	j -й елемент у наборі \mathcal{I}
R	Рейтингова матриця, що дорівнює $U \times T$
\hat{r}_{ij}	Прогноз r_{ij} моделі для користувача u_i та елемента t_j
\mathcal{Y}_u	Вихідна матриця
sim_{ij}	Матриця подібності користувачів u_i та u_j
\mathcal{Y}	Глобальний впорядкований список рекомендацій \mathcal{Y}
$\text{mkm1}, \text{mkm2}, \text{mkm3}$	Масштаб рекомендацій для кожного модуля
t_i	Ранг пункту t_i
$s(t_i)$	Об'єднаний бал пункту t_i

нейронна мережа та запропонована масштабована система рекомендацій на основі спільної фільтрації (Sivaramakrishnan et al., 2020). Масштабовані рішення для систем рекомендацій на основі змагання також були запропоновані в літературі (Ji & Shen, 2015).

Запропонована модель DLSF використовує контрольоване глибоке навчання та унікає переобладнання шляхом додавання регуляризації. Вона також усуває проблему холодного запуску та розрідженість, використовуючи семантику та найпопулярніші елементи. Оскільки вона перетворює дані великої розмірності в вектор низької розмірності для ефективної обробки, тому є високомасштабованим.

3. Пропонована модель

Запропонована модель рекомендацій на основі злиття ршень агрегує рейтингові дані та семантичну інформацію. Модель використовує семантику для моделювання профілю користувача для всебічної схожості предметів. Вона також вирішує проблеми моделей спільної фільтрації, зосереджуючись на проблемі холодного запуску та забезпечуючи масштабованість.

У системах рекомендацій основними компонентами є набір користувачів U , набір елементів \mathcal{I} та взаємодія між U та \mathcal{I} . Взаємодія зазвичай представлена у вигляді рейтингової матриці:

$$R = U \cdot I = \begin{matrix} & t_1 & \dots & t_m \\ u_1 & r_{11} & \dots & r_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ u_n & r_{n1} & \dots & r_{nm} \end{matrix} \quad (1)$$

тут $u_i \in U$ та $t_j \in \mathcal{I}$. Кожна сутність u_{it} рейтингової матриці R лежить у певному діапазоні. Рядки R представляють користувачів, а стовпці представляють елементи. Символи описані в таблиці 1.

Нехай P — матриця прогнозування, отримана після застосування моделі, де $\hat{r}_{ij} \in P$. Метою системи рекомендацій є мінімізація різниці між P та початковою матрицею оцінок \mathcal{Y} . Постановку проблеми можна подати так:

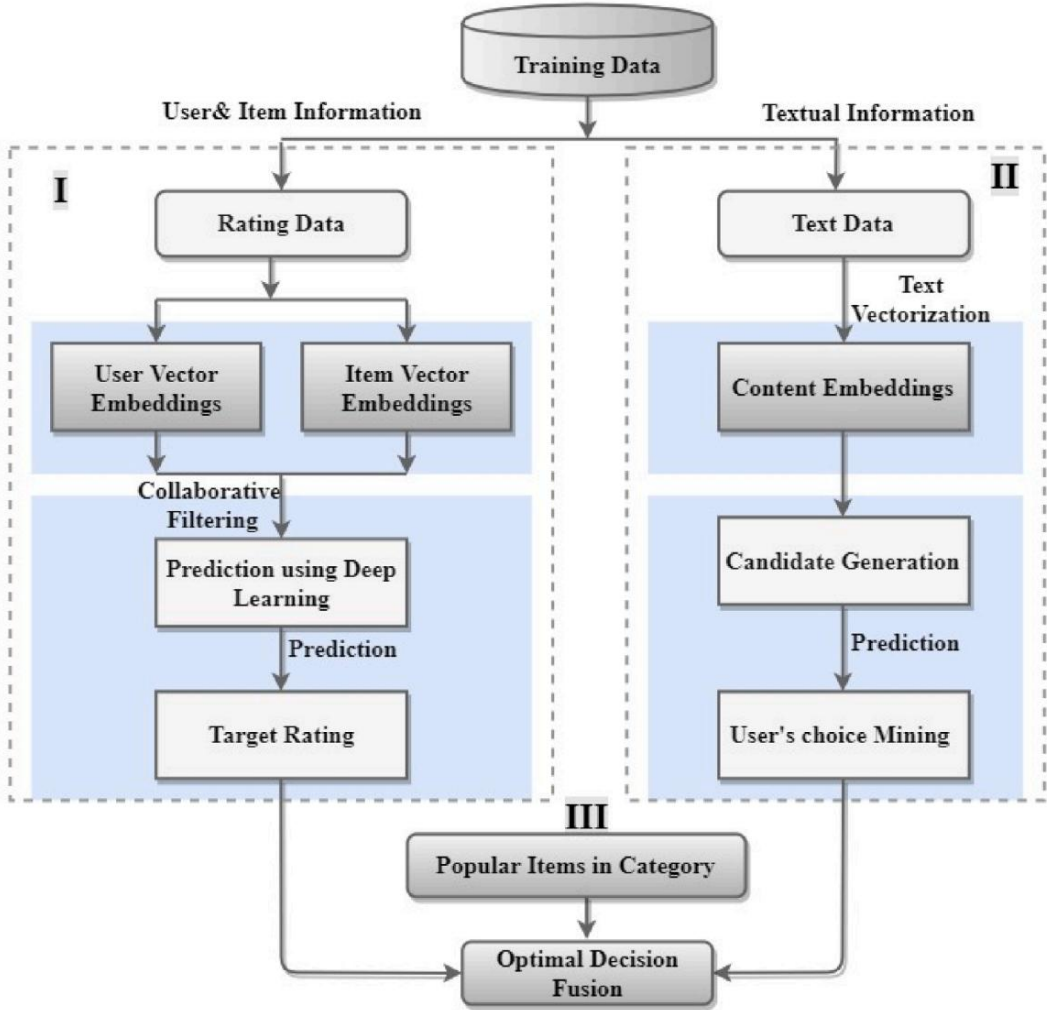


Рис. 3. Структурна схема запропонованої моделі.

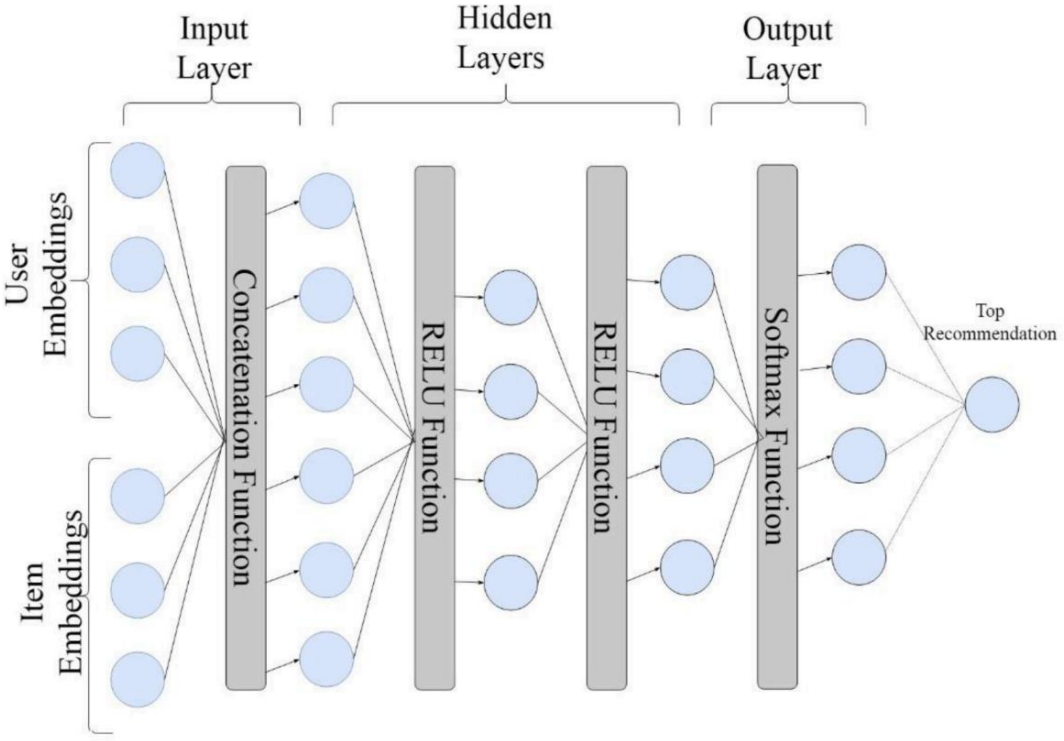


Рис. 4. Рівень глибокого навчання в модулі І.

$$C_{U,I} = \arg \min_{U,I} \sum_{(i,j) \in P} [p_{ij} - v_i]^2 \tag{2}$$

Блок-схема запропонованої моделі показана на рис. 3. Запропонована модель DLDF складається з трьох модулів: перший модуль виконує спільну функцію злиття рейтингів та характеристик користувача та елементу, другий модуль виконує обробку тексту для обчислення подібності між ними, а третій модуль надає рекомендації щодо популярних товарів, якщо користувач новий, не надавши жодного рейтингу. Результати цих трьох модулів будуть використані для прийняття остаточного рішення.

3.1. Модуль 1

У першому модулі рейтингова матриця R розкладається на менші приховані фактори U та I розміру k для користувачів та елементів відповідно таким чином, щоб $k \leq \min(n, m)$. Кожен рядок u_i в U , де $i = (1, \dots, n)$, є вектором розміром k , а кожен стовпець t_j в I , де $j = (1, \dots, m)$, є вектором розміром k . Тут рядки є вбудованими векторами дискретних множин U та I , які в подальшому використовуються для утворення векторів безперервних значень. Враховуючи розподіл рейтингів, ми хочемо розкласти R так, щоб $R \approx U \cdot I^T$. Це зробить подальші обчислення ефективними з точки зору складності. Оскільки U та I окремо значно менші за рейтингову матрицю, їхній продукт не є розподілом, як матриця взаємодії.

Основною метою модуля 1 у запропонованій моделі є мінімізація квадратичної помилки між вхідною рейтинговою матрицею R та прогнозованою матрицею \hat{R} шляхом знаходження вихідних U та I . Їх можна подати як:

$$C_{U,I} = \arg \min_{U,I} \sum_{(i,j) \in R} [(U_i)(I_j)^T - R_{ij}]^2 \tag{8}$$

Пакетна оптимізація градієнтного спуску використовується для отримання матриць U та I , щоб градієнт не потрапив би в нуль для кожного значення в R розглядаючи одну матрицю за раз. Це буде виконано шляхом додавання регуляризованих членів до функції вартості, щоб уникнути переобчислення. Це допоможе розглянути винятки, наприклад, деякі елементи надто хороші, тому майже кожен користувач дає їм високу оцінку. Після додавання членів регуляризації рівняння

стає:

$$(U, I) = \arg \min_{U,I} \sum_{(i,j) \in R} [(U_i)(I_j)^T - R_{ij}]^2 + \frac{\alpha}{2} \sum_{i,k} (U_{i,k})^2 + \frac{\beta}{2} \sum_{j,k} (I_{j,k})^2 \tag{9}$$

де α — розмір швидкості навчання, а k визначає розмір прихованих факторів.

Приховані функції користувача та елемента надаються як вхідні дані для вхідного рівня нейронної мережі прямого зв'язку. Робота нейронної мережі Feed forward показана на рис. 4. Для будь-якого рейтингу R_{ij} вхідні дані для першого рівня будуть:

$$x_{in} = f(U_i, I_j) \tag{10}$$

де x_{in} — вхідні дані для першого рівня, а $f(\cdot)$ — функція конкатенації. Результати вхідного шару будуть введені в перший прихований шар. Вхідні дані для першого прихованого шару x_{h1} можна задати як:

$$x_{h1} = f(W_1 x_{in} + b_1) \tag{11}$$

тут W_1 та b_1 — вагова матриця та зміщення вхідного шару відповідно. Функція $f(\cdot)$ є функцією активації, якою в даному випадку є ReLU. Нехай I — кількість прихованих шарів. Вихід останнього прихованого шару буде вхідом для вихідного шару, який можна подати як:

$$y = f(W_{out} x_{out} + b_{out}) \tag{12}$$

де функція $f(\cdot)$ є функцією активації SoftMax, а W_{out} , x_{out} та b_{out} — вагова матриця, вхідні дані до вихідного шару та зміщення. Ми припускаємо, що оцінки знаходяться в діапазоні від 1 до 5, функція активації SoftMax можна обчислити за допомогою $\sigma_i = \frac{e^{y_i}}{\sum_{j=1}^5 e^{y_j}}$. Товару було присвоєно рейтинг з

максимальною ймовірністю. Рівняння між прогнозованим рейтингом \hat{r}_{ij} фактичним рейтингом обчислюється шляхом застосування регуляризації L2 до функції перехресної ентропії. Регуляризація L2 допоможе зменшити перенавчання тренувальних даних, контролюючи величини призначених ваг.

3.2. Модуль 2

Другий модуль виконує фільтрацію на основі мети цього модуля є надання рекомендацій щодо елементів, що мають подібний вміст до елемента, вибраного користувачем у минулому. Це допоможе у вирішенні проблеми холодного запуску, яка є основною проблемою спільної фільтрації. Дані у вигляді тексту розглядаються тут для обчислення подібності. Текстом може бути відгук, наданий користувачем, жанр фільму або специфікація товару. Попередня обробка виконується над текстовими даними, щоб видалити знаки пунктуації, пробіли, статті та інші не потрібні дані. Вбудовані вмісту формуються шляхом поєднання всіх атрибутів у Bag_of_words. Однак модель може лише порівнювати вектор (матрицю) з іншим вектором. Таким чином, Bag_of_words перетворюється на векторне представлення за допомогою частотного лемматизатора для кожного ключового слова, наявного в Bag_of_words. Після отримання матриці подібності функція подібності використовується для обчислення подібності між елементами. Оскільки кожен елемент є вектором у n-вимірному просторі, для обчислення подібності між векторами обчислюються кути між ними. У цьому статті для обчислення кутів між векторами використовується подібність косинусів, яка може бути задана як:

$$\text{sim}(p, q) = \frac{pq}{\|p\| \|q\|} = \frac{\sum_{i=0}^n p_i q_i}{\sqrt{\sum_{i=0}^n p_i^2} \sqrt{\sum_{i=0}^n q_i^2}} \tag{13}$$

де p і q — елементи, подібність яких обчислюється. Матриця подібності складається з схожості кожного елемента з іншим елементом у діапазоні $[0,1]$ може бути представлена як:

	item1	item2	...	itemm
пункт1	1	sim12	...	sim1m
...
См = пункт 2	sim21	1	...	sim2m

(14)

Після обчислення подібності матриця оцінок найкращих рекомендацій надаються як вихідні дані. Рейтинги сумарних пар користувач-предмет обчислюються шляхом обчислення зваженої суми. Усі рейтинги, надані подібними елементами до цільового елемента, значення подібності враховуються для обчислення зваженої суми, яка може бути подана як:

$$p_{u,t} = \frac{\sum_{t=0}^m (\text{sim}_{tm} * P_{ym})}{\sum_{t=0}^m (\text{sim}_{tm})} \tag{15}$$

Після отримання рейтингів вони передаються на рівень остаточного прийняття рішення для створення остаточних прогнозів.

3.3. Модуль 3

Для нових користувачів, як раніше не оцінювали жодного елемента в категорії, давати рекомендації за допомогою спільної фільтрації та фільтрації на основі вмісту є дуже недоречним. Для таких користувачів у пошуковій категорії надаються рекомендації на основі популярності. Популярними будуть ті, що найбільше сподобалися або оцінили користувачі в подібній категорії. Для елемента, який шукав користувач, будуть рекомендовані інші елементи в тій же категорії з максимальною кількістю оцінок.

$$\text{популярний_item} = \max_{i=0}^m U[\text{подібні_товари}(ti)] \tag{16}$$

3.4. Злиття рішення

Для прийняття остаточного рішення щодо пункту, як слід рекомендувати, збирається середнє зважене нормалізоване бальне значення з усіх модулів, які підлягають на 5 частин однакового розміру випадковим чином. Одночасно 4 частини

готуються остаточний рейтинг. Допоможе впоратися з проблемою холодного запуску, розподіленості та дефіциту. Для нових предметів, оцінки яких ще не доступні, рекомендації складаються виключно за модулем 3, який вирішить проблему холодного запуску та розподіленості. Для боротьби з дефіцитом семантичної інформації використовується в модулі 2. Якщо елемент популярний, тоді буде використано спільне фільтрування з модуля 1, тоді як семантична інформація буде використана в контентному фільтруванні з модуля 2, якщо елемент не популярний дуже популярний.

Нехай набір рекомендацій μ_1, μ_2, μ_3 є отриманими рекомендаціями з модуля 1, модуля 2 та модуля 3 відповідно. Для кожного елемента ti з μ_1, μ_2, μ_3 буде оцінено релевантність вибору користувача. Центр об'єднання ретельно об'єднає локальні рекомендації всіх модулів для надання глобального впорядкованого списку рекомендацій Y . До всіх елементів, рекомендованих модулями, буде застосовано нормалізацію для отримання рівномірного розподілу балів. Враховуючи ранг ti для кожного елемента ti кожним модулем локально в μ_1, μ_2, μ_3 , нормалізована вага оцінки буде призначена кожному елементу. Бали присвоюються предметам, які належать до Модуля 1, Модуля 2 та Модуля 3 окремо. Вага показника нормалізації w_{ti} для елемента ti може бути розрахована за такою формулою:

$$w_{ti} = \frac{1}{\max(\mu_j) - \min(\mu_j)} \tag{17}$$

де $\max(\mu_j)$ і $\min(\mu_j)$ представляють максимум і мінімум рангів відповідно з локального набору рекомендацій μ_j . Оскільки $ti = 0$, якщо предмет не рекомендований модулем.

Об'єднання ретельно надасть глобальний список рекомендацій Y , враховуючи об'єднання елементів у μ_1, μ_2, μ_3 . Нехай $\delta = [\mu_1 \cup \mu_2 \cup \mu_3]$, глобальний список рекомендацій Y впорядковано відповідно до зменшення значення об'єднаного балу $s(ti)$ для кожного пункту ti . Об'єднаний бал $s(ti)$ буде розраховано за допомогою лнійної комбінації нормалізованих ваг балів, подано як:

$$s(ti) = \sum_{j=1}^3 \rho(\mu_j) \cdot w_{ti} \tag{18}$$

де $\rho(\mu_j)$ – пріоритет, призначений таким модулям, що $\rho(\mu_j)$

$\rho_j = 1$ для $\mu_j = 1$. Якщо товар оцінюють менше 2 користувачів, то рекомендаціям, наданим модулем 3, буде надано високий пріоритет, але якщо елемент оцінено менше ніж 10 користувачами, рекомендаціям, наданим модулем 2, буде надано високий пріоритет. Для елемента, який оцінюють багато користувачів, рекомендації модуля 1 мають найвищу вагу.

Необхідні рекомендаційні елементи будуть збережені в Y . Чим вищий бал, тим вищим буде ранг продукту в глобальному списку рекомендацій.

Алгоритм 1. Злиття рішення		
Вхід: Рекомендований набір μ_1, μ_2, μ_3 .		
Вихід: Глобальний впорядкований список рекомендацій Y .		
1. $\delta = \mu_1 \cup \mu_2 \cup \mu_3$		
2. Для кожного $ti \in \delta$ обчислити w_{ti} за допомогою (17)		
3. Обчислити $s(ti)$ за допомогою (18)		
4. $Y = \delta$		
5. Сортувати Y за зменшенням $s(ti)$		
6. Вивести Y		

4. Експериментальна оцінка

Для оцінки точності та ефективності запропонованої моделі було проведено масштабні експерименти. Для проведення всіх експериментів була використана п'ятикратна перехресна перевірка. У цьому випадку всі набори даних поділяються на 5 частин однакового розміру випадковим чином. Одночасно 4 частини

Таблиця

2 Опис набору даних.

Набір даних	Рейтинги	Користувач	Предмети
Yelp	8 М	1,9 М	209 393
MovieLens100K	0,1 М	0,01 М	0,017
MovieLens25M	25 М	0,162 М	0,062 М
Amazon	82,83 М	20,98 М	9,35 М

розглядається для навчання моделі, а решта використовується для тестування. Використовуючи п'ятикратну перехресну перевірку, щоразу співвідношення навчання та тестування становитиме 80:20.

4.1. Набір даних

Ми використали чотири реальні набори даних: Yelp ([Yelp Open Dataset, 2021](#)), MovieLens100K ([MovieLens 100K Dataset, 2016](#)), MovieLens25M ([MovieLens 25M Dataset, 2010](#)) Amazon (R. He, 2016), взяті з загальнодоступних репозиторіїв. Докладно про набори даних показано в таблиці 2. Усі вони мають неоднорідну інформацію. Yelp — це веб-сайт для рейтингів користувачів, на якому користувачі можуть оцінювати місцеві компанії в діапазоні від 1 до 5. Користувачі також можуть публікувати фотографії та додавати відео. Компанії рекомендуються на основі вподобань користувачів, наданих користувачем. Розглянуто два набори даних, зібрані з веб-сайту MovieLens, один з оцінками 100 K (набір даних MovieLens 100K, 2016), а інший — 25 M (набір даних MovieLens 25M, 2010). Набори даних надані GroupLens Research з Університету Міннесоти. Четвертий використаний набір даних — Amazon. Набір даних складається з 82,83 мільйона унікальних вступів приблизно в 20 мільйонів користувачів (R. He, 2016).

4.2. Конкурентні алгоритми

Продуктивність запропонованої моделі DLSF порівнюється з базовими методами.

о FMR: підхід, заснований на факторизації, який використовує оцінку за елементами користувача матрицю (Rendle, 2012).

о PMF: підхід до декомпозиції матриць на основі SVD, який розглядає лише ненульові елементи (Salakhutdinov & Mnih, 2009). о SVD++:

поєднує переваги моделі суцільності та латентної моделі (Koren, 2008). о Підхід на основі автокодера:

реалізація спільної фільтрації за допомогою автокодера на основі глибокого навчання (Strubell et al., 2016). о DLM1: Рекомендаційний модуль 1 запропонованої моделі

4.3. Показники ефективності

4.3.1. Помилка

Для аналізу продуктивності використовуються середня абсолютна помилка (MAE), середньоквадратична помилка (RMSE), оскільки вони є популярними показниками та використовуються в багатьох областях для оцінки ефективності моделі. У MAE — величина похибки, обчислена за абсолютними значеннями, враховуючи середні значення всіх абсолютних значень похибки. Його можна подати як:

$$MAE = \frac{1}{n} \sum_{i,j} |p_{i,j} - y_{i,j}| \tag{3}$$

тут $p_{i,j}$ — фактична оцінка елемента i,j , а $y_{i,j}$ визначають прогнозований рейтинг

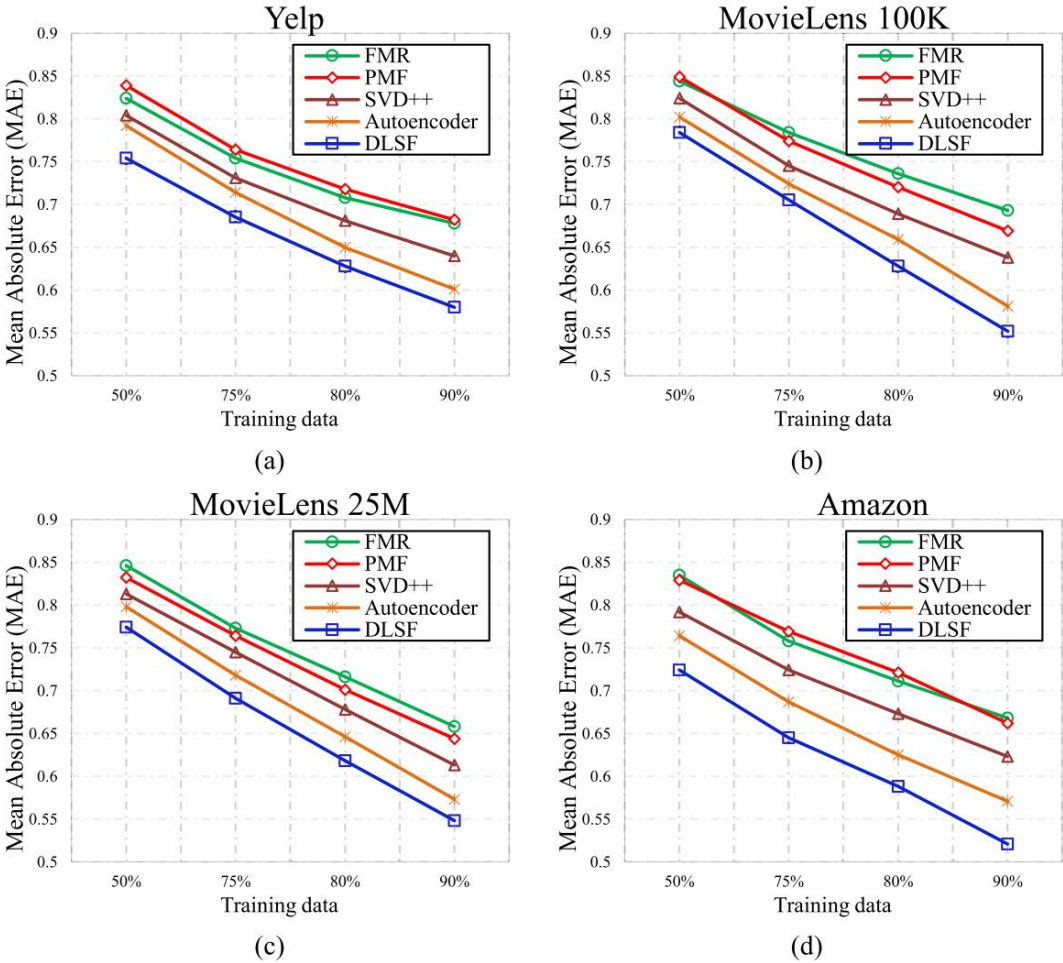


Рис. 5. Показник продуктивності з точки зору MAE для: (a) Yelp, (b) MovieLens100K, (c) MovieLens25M, (d) Amazon.

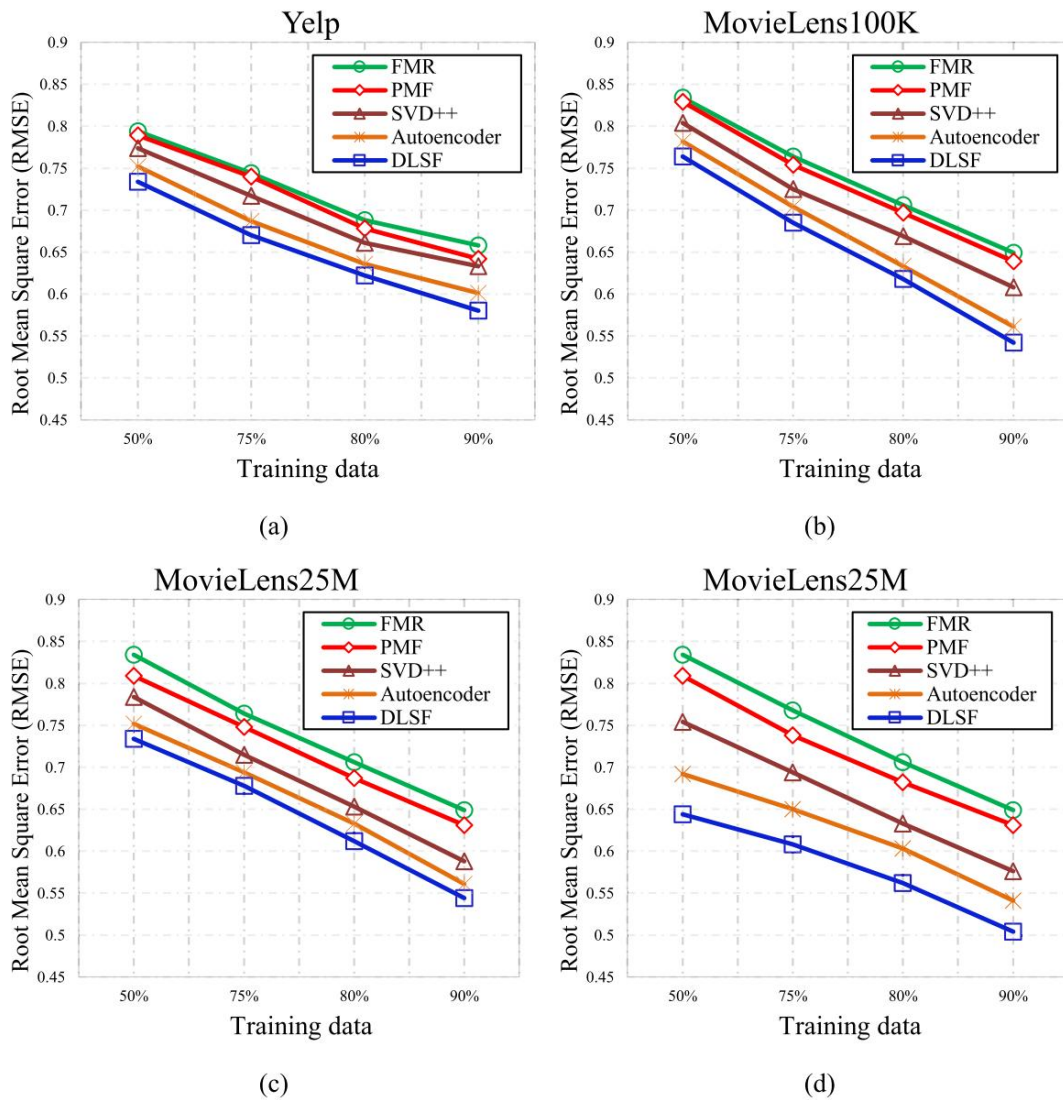


Рис. 6. Вимрювання продуктивності в термінах RMSE для: (a) Yelp, (b) MovieLens100K, (c) MovieLens25M, (d) Amazon.

фактичний рейтинг користувача i для пункту j в даних. RMSE обчислює середнє значення квадратів помилок. Його можна подати як:

$$RMSE = \frac{1}{M} \sum_{ij} (p_{ij} - y_{ij})^2 \quad (4)$$

І в MAE, в RMSE значення помилок обчислюються шляхом знаходження різниці рейтингу, передбаченого моделлю, з уже в даних значеннями рейтингу тестового набору даних. Чим менше значення MAE RMSE, тим краще прогнозна точність.

4.3.2. Точність

Точність може бути обчислена в термінах точності, в даних або комбінації обох. Точність можна визначити як в даних

правильно рекомендовані предмети та загальна кількість рекомендованих предметів, тоді як запам'ятовування є співвідношенням в даних рекомендованих предметів до всіх рекомендованих предметів. В рекомендаційних системах, точність в даних розраховується щодо найкращих s рекомендацій, де s – кількість найкращих рекомендацій, наданих користувачеві. Обидва обчислюються за допомогою позитивними значеннями (TP), справді негативними значеннями (TN), помилково позитивними значеннями (FP) та помилково негативними значеннями (FN). Точність можна подати як:

$$\text{Точність} = \frac{TP}{TP + FP} = \frac{\text{Правильно рекомендовані предмети}}{\text{Рекомендовані предмети}} \quad (5)$$

Чим вище Precision@ s , тим краща точність. В даних можна обчислити як:

$$\text{Згадати} = \frac{TP}{TP + TN} = \frac{\text{Правильно рекомендовані предмети}}{\text{Правильно рекомендовані та не рекомендовані товари}} \quad (6)$$

Чим вище значення Recall@ s , тим краща точність. Загалом, точність можна обчислити як в даних всього правильно рекомендованих елементів до всього рекомендованих можна надати як:

$$\text{Точність} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{\text{Вс. Правильно складені Рекомендації}}{\text{Вс. рекомендації}} \quad (7)$$

4.4. Результати та аналіз

Ефективність запропонованої моделі DLSF оцінюється з точки зору середньої абсолютної похибки (MAE), середньої квадратичної похибки (RMSE), точності, повторюваності та точності. Далі модель тестується шляхом зміни кількості епох для вищезгаданих параметрів. Ефективність

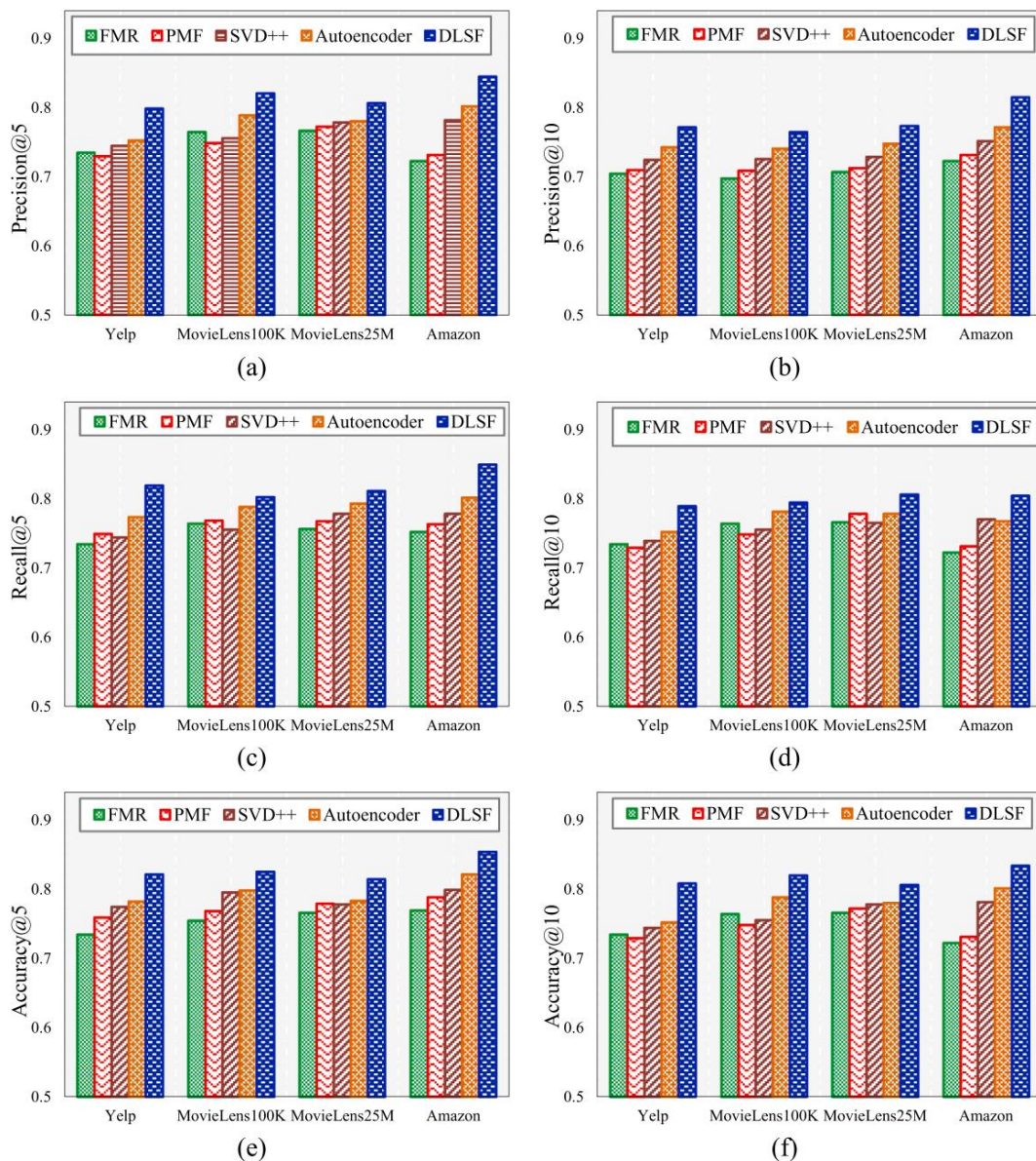


Рис. 7. Продуктивність з точки зору (а)-(б) точності, (с)-(д) - в дкликання, (е)-(ф) точності.

Показано використання моделі на основі прийняття рішення над моделлю спільної фільтрації на основі глибокого навчання Module1. Для обчислення MAE, RMSE та показник в точності швидкість навчання залишається рівною 0,02, а кількість епох – 300.

4.4.1. MAE, RMSE

Для оцінки продуктивності використовуються середня абсолютна похибка (MAE), середньоквадратична похибка (RMSE), оскільки вони є популярними показниками та використовуються в багатьох областях для оцінки продуктивності. Навіть незначне покращення показників в MAE та RMSE має помітний вплив на ефективність рекомендацій.

Обидва враховують прогнозовані рейтинги та фактичні рейтинги для обчислювальної помилки. Таким чином, під час об'єднання рішень усі елементи розглядалися замість основних рекомендацій для обчислення MAE та RMSE. Як показано на рис. 5, MAE та RMSE розраховуються для набору даних YELP, MovieLens100K, MovieLens25M та Amazon для порівняння продуктивності запропонованого DLSF з компетентними підходами на основі FMR, PMF, SVD++ та Autoencoder. Ефективність оцінюється шляхом зміни сповільнення тестових навчальних даних. К-перехресна складка використовується для оцінки, де k вважається рівним

2, 4, 5, 10. Для всіх наборів даних, як показано на рис. 5, запропонований DLSF має нижчу середню абсолютну похибку порівняно з компетентними алгоритмами. Запропонований DLSF працює найкраще з точки зору MAE для набору даних Amazon, оскільки він містить як рейтинги, так і дуги в достатній кількості. DLSF ефективно працює з проблемою холодного запуску та розрідженості, оскільки можна помітити, що DLSF є ефективним з точки зору як RMSE, так і MAE. При k = 5 80 % даних є розглядається для навчання за раз, запропонований DLSF є 6,2 %, 14,4 %, 22,6 % моделі 20,9 % ефективнішим з точки зору MAE, ніж на основі автокодерів, SVD++, PMF та FMR в даних.

Ефективність запропонованого DLSF разом з іншими компетентними підходами з точки зору RMSE показана на рис. 6. Як показано, усі підходи не забезпечують хороших результатів з точки зору RMSE, коли k = 2 (50 %). Для Amazon, коли k = 10 (90 % тренувальних даних), запропонований DLSF на 7,3 %, 14,28 %, 25,19 %, 28,7 % ефективніший з точки зору RMSE, ніж на основі Autoencoder, SVD++, PMF та FMR в даних. У той час як для набору даних Amazon, коли k = 10 (80 % тренувальних даних), DLSF на 7,29 %, 12,63 %, 21,35 %, 25,6 % ефективніший з точки зору RMSE, ніж Autoencoder-based, SVD++, PMF та FMR в даних.

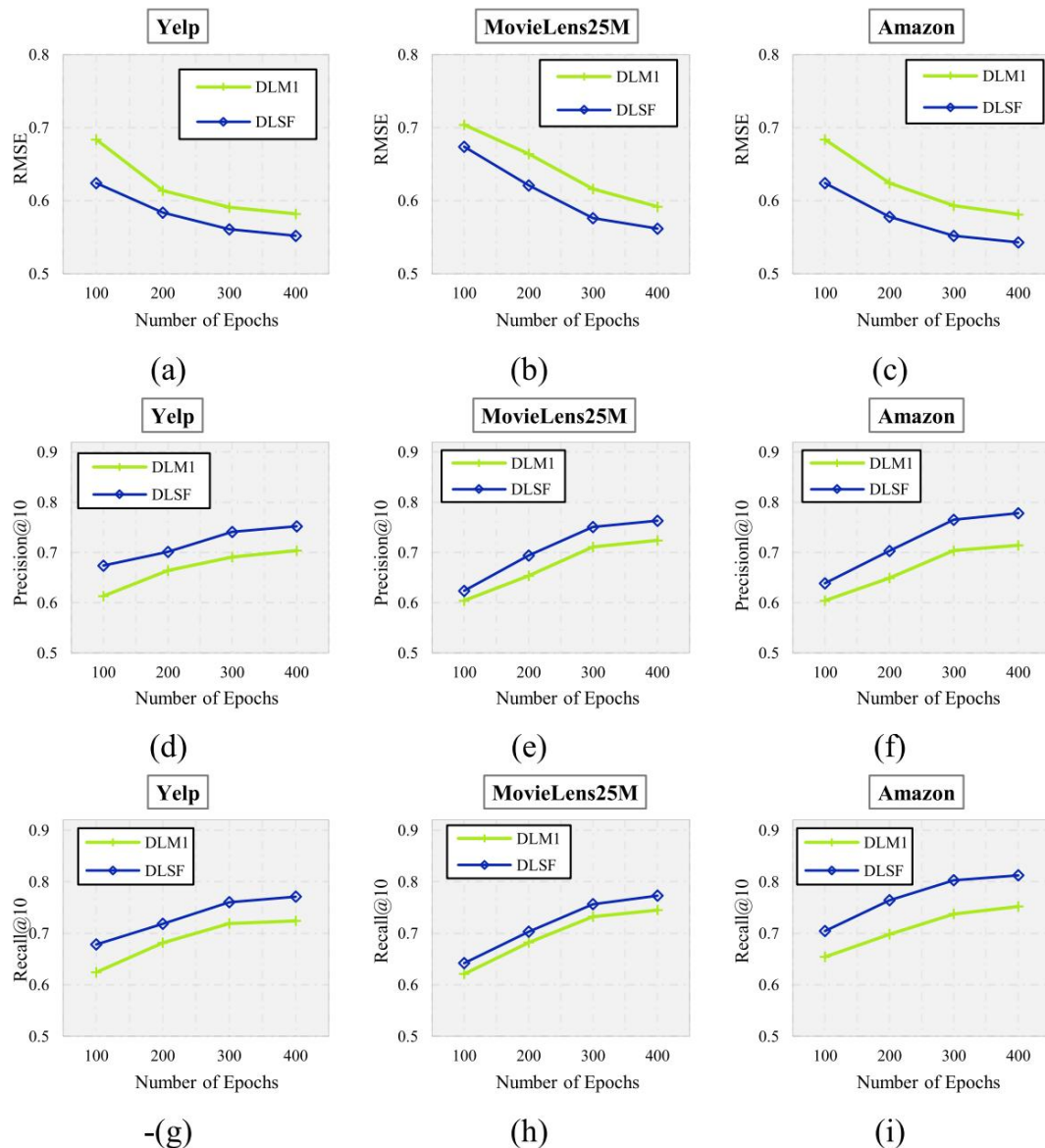


Рис. 8. Продуктивність за змінною кількістю епох (a)-(c) точність, (d)-(f) в дкликання, (g)-(h) точність.

Для набору даних MovieLens25M, коли $k = 5$, DLSF є на 3,4 %, 6,6 %, 12,2 % 15,3 % ефективнішим, ніж Autoencoder-based, SVD++, PFM FMR в дповдню з точки зору RMSE.

4.4.2. Точність

Для оцінки продуктивності розглядається 5-кратна перехресна перевірка, а s розглядається як 5 10. Продуктивність DLSF порівнюється з компетентними підходами до набору в даних Yelp, MovieLens100K, MovieLens25M Amazon, як показано на рис. 7. Для Precision@5 запропонований DLSF працює краще порівняно з Precision@10. Те ж саме з пригадуванням точності. Для Precision@5, розрахованого для Amazon, DLSF є на 5,09 %, 7,4 %, 13,3 % 14,5 % ефективнішим, ніж Autoencoder-based, SVD++, PFM FMR в дповдню. У той час як для Yelp запропонований DLSF на 5,7%, 6,7%, 8,6% 8,02% ефективніший з точки зору RMSE, ніж на основі Autoencoder, SVD++, PFM FMR в дповдню. Оскільки Amazon має більше даних для навчання, глибоке навчання використовує переваги навчання на більшому наборі даних забезпечує кращі результати порівняно з іншими підходами. Для Precision@10, розрахованого для Amazon, DLSF на 5,28 %, 7,7 %, 10,19 % 11,30 % ефективніший, ніж Autoencoder-based, SVD++, PFM FMR в дповдню.

Продуктивність для обчислення Recall показано на рис. 7(c) рис. 7(d). У середньому DLSF на 9 % ефективніший за Recall@10 на 7,02 % ефективніший за Recall@5. Усі алгоритми дають кращі результати при $s = 5$ порівняно з $s = 10$.

Точність також спостерігається для $s = 5$ $s = 10$, як показано на рис. 7(e) рис. 7(f). Запропонований DLSF має на 6,99 % 9,022 % більшу точність для accuracy@5 accuracy@10, якщо середні значення обчислюються для всіх наборів даних компетентних підходів.

4.4.3. Аналіз параметрів

Ефективність DLSF порівнюється з модулем 1 DLSF (DLM1), щоб проаналізувати переваги використання підходу на основі машинного навчання разом з системою рекомендацій на основі глибокого навчання й спільної фільтрації. Це також допоможе проаналізувати продуктивність моделі на основі синтезу та те, як вона вирішує проблему холодного запуску. Продуктивність оцінюється шляхом змінної кількості епох у багатоваріантній нейронній мережі прямого зв'язку. Аналіз з точки зору RMSE, Precision@10 Recall@10 виконується за допомогою набору в даних Yelp, MovieLens25M Amazon, як показано на рис. 8.

За всіма параметрами можна спостерігати, що DLSF забезпечує кращі результати для набору даних Amazon, ніж DLM1. DLSF краще справляється з проблемою холодного запуску та забезпечує ефективні результати. Для Yelp Movie-Lens25M також можна спостерігати ефективність DLSF DLM1. Немає великої різниці між DLSF DLM1 для 300 епох 400 епох, тому в цьому статті весь аналіз виконується для 300 епох.

Як показано на рис. 8(а), RMSE для запропонованого DLSF нижчий, ніж DLM1. DLSF є найефективнішим, ніж DLM1, для набору даних Amazon через наявність достатньої кількості витрат на угляд огляду. Те ж саме стосується Precision@10 Accuracy@10, DLSF вважається ефективнішим, ніж DLM1.

Пропонований DLSF на 6,09%, 3,6% 7,38% точніший за Yelp, MovieLens100K Amazon в дивовижно, коли $s = 10$.

Запропонований DLSF має покращену продуктивність порівняно з DLM1, оскільки він має справу з проблемою холодного старту та розривності, які виникають у спільній фільтрації, яка використовується в модулі 1. Крім того, можна помітити, що точність запропонованого DLSF становить половину Amazon yelp. для набору даних MovieLens100K. Саме завдяки наявності більшій семантичній інформації в наборі даних Yelp Amazon вирішується проблема дефіциту.

5. Висновок

У цьому документі пропонується система рекомендацій на основі злиття рейтингів, яка використовує як рейтингів дані, так і семантичну інформацію. Для аналізу зв'язку між користувачем і елементом пропонується п'ять спільних фільтрів на основі глибокого навчання, який має справу з проблемами холодного запуску та розривності. Крім того, для усунення обох проблем використовується п'ять даних, заснованих на семантичному навчанні, для визначення персоналізованих результатів користувача. Крім того, для користувачів, які є новими та не оцінювали жоден товар, надається жодного в дугу, враховуються популярні товари в цій конкретній категорії. Наприклад, розглядається п'ять даних на основі злиття рейтингів шляхом призначення рейтингів ваг кожному модулю для надання остаточних рекомендацій.

Ефективність запропонованого DLSF спостерігалася за чотирма реальними наборами даних з урахуванням різних показників. Доведено, що DLSF є ефективнішим, ніж компетентні алгоритми з точки зору середньої абсолютної похибки, середньої квадратичної похибки, точності, в диктанні та точності. Продуктивність частини, заснованої на глибокому навчанні, також порівнюється з DLSF для аналізу точності. DLSF вважається більш ефективним з точки зору RMSE, точності та в диктанні, ніж його нижча версія. У майбутньому запропонована модель може бути розширена для розгляду рейтингової та семантичної інформації разом в одному конвеєрі, щоб зробити модель обчислювально ефективною. Крім того, модель буде змінено, щоб обробляти більш повну інформацію.

Декларація про конкуруючий інтерес

Автори заявляють, що у них немає в діючих конкуруючих фінансових інтересів або особистих стосунків, які могли б вплинути на роботу, про яку йдеться в цій статті.

Доступність даних

Дані, використані в статті, взяті з публічних сховищ. Посилання надано в рукописі.

Список літератури

Bhatia, V., & Rani, R. (2018a). Ap-FSM: паралельний алгоритм для наближеного аналізу частини графа в за допомогою Pregel. *Експертні системи з додатками*, 106, 217–232. <https://doi.org/10.1016/j.eswa.2018.04.010>

Bhatia, V., & Rani, R. (2018b). DFuzzy: модель нечіткої кластеризації на основі глибокого навчання для великих графів. *Системи знань та інформації*, 57 (1), 159–181. <https://doi.org/10.1007/s10115-018-1156-3>

Bhatia, V., & Rani, R. (2019). Розподілена модель виявлення спільного товариства, що перекивається, для великих графів в за допомогою автокодувальника. *Комп'ютерні системи майбутнього покоління*, 94, 16–26. <https://doi.org/10.1016/j.future.2018.10.045>

Bhatia, V., Saneja, B., & Rani, R. (2017). INGC: кластеризація графів в виявленні викидів в алгоритм, що використовує поширення м'якого міжнародна конференція з машинного навчання та науки про дані INGC, 2017, 68–74. <https://doi.org/10.1109/MLDS.2017.14>

Бу М., Ши Дж., Чанг К. С. та Лі Х. (2019). Навчання на основі метаграфів в гетерогенних графах. *IEEE Transactions on Knowledge and Data Engineering*, 1–15. <https://doi.org/10.1109/TKDE.2019.2922956>

Фу, М., Ку, Х. та І, З. (2019). Нова модель спільної фільтрації на основі глибокого навчання для системи рекомендацій. *IEEE Transactions On Cybernetics*, 49(3), 1084–1096.

Хоссейн, А., Чен, Т. (2018). HeteroMed: гетерогенна інформаційна мережа для медичної діагностики. *У матеріалах 27-ї міжнародної конференції ACM з управління інформацією та знаннями (стор. 763–772)*.

Джі К. та Шен Х. (2015). Звернення до холодного запуску: масштабована рекомендація з тегами та ключовими словами. *Системи на основі знань*, 83, 42–50.

Катарія Р. та Верма О.П. (2017). Ефективний спільний рекомендаційний алгоритм в системі з пошуком зсувів. *Єгипетський журнал інформатики*, 18 (2), 105–112. <https://doi.org/10.1016/j.eij.2016.10.002>

Kirp, TN, & Welling, M. (2017). Заповнення згорткової матриці графа. *Препринт ArXiv ArXiv:1706.02263*, 1–9.

Корень Ю. (2008). Розкладання на множники зустрічається з оцінками. 426. <https://doi.org/10.1145/1401890.1401944>.

Лі Х. та Лі Дж. (2019). Масштабована система рекомендацій на основі глибокого навчання. *ICT Express*, 5 (2), 84–88. <https://doi.org/10.1016/j.icte.2018.05.003>

Лі Х., Ван Ю., Лю З. та Ши Дж. (2020). Багатозадачне навчання для рекомендацій через різноманітну інформаційну мережу. *IEEE Transactions on Knowledge and Data Engineering*, 4347(c), 1. <https://doi.org/10.1109/tkde.2020.2983409>

Lika, B., Kolomvatsos, K., & Hadjiefthymiades, S. (2014). Зміщення з проблемою холодного запуску в системах рекомендацій. *Експертні системи з додатками*, 41 (4 ЧАСТИНА 2), 2065–2073. <https://doi.org/10.1016/j.eswa.2013.09.005>

Лі Дж., Лі Ю. та Лі Дж. (2019). Нова система рекомендацій через L0-регуляризовану систему $\text{convex optimization true positive}$. *0123456789 Neural Computing and Applications*, 1–15. <https://doi.org/10.1007/s00521-019-04213-w>

Лю, Ю., Чжан, Г., Цзін, Х., Юань, Х. (2019). Взаємодія на основі мультиінформаційного синтезу системи рекомендацій. *Журнал фізики: Серія конференції*, 1229 (1). <https://doi.org/10.1088/1742-6596/1229/1/012010>

Матур, А., Джугур, С.К., Ейрбак, М. (2019). Система рекомендацій на основі графів в для харчових продуктів. В Першому міжнародній конференції з графових обчислень (GC), 83–87. <https://doi.org/10.1109/GC46384.2019.00020>

Набір даних MovieLens 100K. (2016). <https://grouplens.org/datasets/movielens/100k/>.

Набір даних MovieLens 25M. (2010). <https://grouplens.org/datasets/movielens/25m/>.

Ніші, Т. С. та Фатма, Н. (2019). Глибинний аналіз застроїв Amazon.Com в дуги та рейтинги. *Міжнародний журнал з програмних обчислень, штучного інтелекту та програм (IJSCAI)*, 8(1), 1–15.

Пал, А., Чакрабарті, Д. (2016). Розповсюдження м'якого за допомогою нейронних мереж. На 13-му міжнародному симпозіумі IEEE з б'юмедичної візуалізації (ISBI) (стор. 1265–1268).

Ван, Р., Дж.М. (2016). Злетіти та падіння: моделювання взуальної еволюції модних тенденцій за допомогою однокласної спільної фільтрації. У матеріалах 25-ї міжнародної конференції Всесвітньої мережі (стор. 507–517). <https://cseweb.ucsd.edu/~jmcauley/datasets.html>.

Ренді, С. (2012). Машини факторизації з libFM. Оперативна система інтелектуальних систем технологій, 3(3). <https://doi.org/10.1145/2168752.2168771>

Салахутдінов Р. та Мінх А. (2009). Імовірність матрична факторизація. *Досягнення в нейронних системах обробки інформації 20 - Матеріали конференції 2007 року* (стор. 1–8).

Сабб, Ю., Джебрі, М., Гезала, Х.Б. (2022). Система рекомендацій на основі графів в для управління кризою Covid-19. *Procedia Computer Science*, 196, 348–355.

Ши К., Чжан З., Цзін Ю., Ван В., Ю П. С. та Ши З. (2018). SemRec: персоналізований семантичний метод рекомендацій на основі зв'язаних різноманітних інформаційних мереж. *World Wide Web*, 22 (1), 153–184.

Сварамкаршан, Н., Субраманьясам, В., Велор, Я.А., Ваджаякумар, В. та Senthilselvan, N. (2020). Гбридна модель на основі глибокого навчання для створення рекомендацій рейтингів. *Нейронні обчислення та додатки*.

Страб, Ф., Гаудель, Р., Мер, Дж. (2016). Гбридна система рекомендацій на основі автокодерів. У серії матеріалів міжнародної конференції ACM 15 вересня, 11–16. <https://doi.org/10.1145/2988450.2988456>

Вулдмос, А., Дулам, С.Н., Дулам, С.А. та Протопапакас, Е. (2018). Глибоке навчання для комп'ютерного зору: короткий огляд. *Computational Intelligence and Neuroscience*, 2018. <https://doi.org/10.1155/2018/7068349>

Ван, Х. (2015). Семантичне глибоке навчання, 1–42.

Ван, К., Сю, Л., Хуан, Л., Ван, К., Лай, Дж. (2019). SDDR: складена система рекомендацій на основі дискримінаційного усунення шуму в на основі автоматичного кодувальника. *Дослідження когнітивних систем*, 55, 164–174. <https://doi.org/10.1016/j.cogsys.2019.01.011>

Вей, Дж., Хе, Дж., Чень, К., Чжоу, Ю., Тан, З. (2017). Спільна фільтрація та система рекомендацій на основі глибокого навчання для елементів холодного запуску. *Експертні системи з додатками*, 69, 29–39. <https://doi.org/10.1016/j.eswa.2016.09.040>

Yelp Open Dataset. (2021). Yelp, Inc. <https://www.kaggle.com/yelp-dataset/yelp-dataset>.

Чжао, Х. (2017). Об'єднання рекомендацій на основі метаграфів у гетерогенних інформаційних мережах. У матеріалах 23-ї міжнародної конференції ACM SIGKDD з відкриття знань та аналізу даних (стор. 635–644).