Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)
Кафедра Інформаційних управляючих систем
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження методів оцінки рекомендаційних систем
.
(тема)

Виконав:
студент 2 курсу, групи ІУСТм-18-1
Щербаков Олександр Едуардович
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)
Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма Інформаційні управляючі системи та технології
(повна назва освітньої програми)

Керівник д.т.н, проф. каф. ІУС Чалий С.Ф.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ Петров К. Е.
(підпис) (прізвище, ініціали)

2019р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Інформаційних управляючих систем

Рівень вищої освіти другий (магістерський)

Спеціальність 122 – Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Спеціалізація Інформаційні управляючі системи та техгології
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 2019 р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

Студентові Щербакову Олександру Едуардовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів оцінки рекомендаційних систем
_____
_____
затверджена наказом по університету від 31.10.2029. № 1590СТ
2. Термін подання студентом роботи до екзаменаційної комісії _____.
3. Вихідні дані до роботи Науково-технічні публікації та інтернет джерела з тематики атестаційної роботи
_____
4. Перелік питань, що потрібно опрацювати в роботі: Аналіз методів оцінки рекомендаційних систем та постановка задачі; Дослідження характеристик рекомендаційних систем; Аналіз процесу роботи рекомендаційних систем; Аналі критеріїв та методів оцінювання рекомендаційних систем; Постановка задачі дослідження; Дослідження методів оцінювання рекомендацій з використанням явного та неявного зворотнього зв'язку; Загальний підхід до оцінювання рекомендаційних систем і їх порівняльний аналіз; Побудова рекомендаційних систем за допомогою колаборативної фільтрації; Удосконалення колаборативної фільтрації на основі пропущених даних з неявним зворотним зв'язком; Структурно-функціональне моделювання процесу оцінювання рекомендаційних систем; Система оцінювання рекомендаційних систем; Удосконалення оцінювання рекомендаційних систем на основі колаборативної фільтрації; Реалізація удосконаленого методу; Експериментальна перевірка удосконаленого методу

5. Перелік графічного матеріалу, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Рисунок 1 – Структура рекомендаційної системи; Рисунок 2 – Процес роботи рекомендаційних систем; Рисунок 3 – Класифікація рекомендаційних систем; Рисунок 4 – Фільтрація вмісту; Рисунок 5 - Колаборативна фільтрація; Рисунок 6 - Комбінування методів в рекомендаційних системах гібридного типу; Рисунок 7 – Модель рекомендаційної системи; Рисунок 8 – Схема класів рекомендаційної системи; Рисунок 9 – Порівняльний аналіз методів підбору рекомендацій; Рисунок 10 – Класифікація методів колаборативної фільтрації; Рисунок 11 – Приклад матриці оцінок; Рисунок 12 – Визначення відносного порядку між парами об'єктів для кожного користувача; Рисунок 13 – Технологія використання удосконаленого методу; Рисунок 14 –. Переваги користувача; Рисунок 15 - AUC при різних гіпермараметрах; Рисунок 16 - Результати похибок експерименту; Рисунок 17 - Результати експерименту; Рисунок 18 – Зміна параметру AUC; Рисунок 19 - Оцінка ефективності системи

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на атестаційну роботу	04 11 2019	
2	Попереднє дослідження галузі завдання	04.11.2019 – 11.11.2019	
3	Дослідження методів оцінювання рекомендацій з використанням явного та неявного зворотнього зв'язку	11.11.2019 – 16.11.2019	
4	Дослідження моделей і методів автоматичної побудови продукційної бази знань у системах управління ІТ-проектами	16.11.2019 – 25.11.2019	
5	Структурно-функціональне моделювання процесу оцінювання рекомендаційних систем	25.01.2019 – 02.12.2019	
6	Удосконалення оцінювання рекомендаційних систем на основі колаборативної фільтрації	02.12.2019 – 07.12.2019	
7	Оформлення пояснювальної записки та графічного матеріалу	07.12.2019 – 10.12.2019	
8	Подання студентом роботи для перевірки на плагіат	10.12.2019	
	Захист	17.12.2019	

Дата видачі завдання .04.11.2019 р.

Студент _____
(підпис)

Керівник роботи _____ д.т.н, проф. каф. ІУС Чалий С.Ф.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до магістерської атестаційної роботи містить: 77 с., 4 розділи, 12 рис., 7 табл., 22 джерела.

Об’єкт дослідження – процес побудови рекомендацій.

Мета дослідження – підвищення ефективності побудови формування рекомендацій на основі ітеративної оцінки результатів роботи рекомендаційних систем.

У роботі виконано дослідження методів оцінки рекомендаційних систем. Проаналізовано існуючі методи оцінки, на підставі проведеного аналізу запропоновано удосконалення оцінювання рекомендаційних систем на основі колаборативної фільтрації на основі пропущених даних з неявним зворотним зв’язком.

В ході дослідження отримані такі результати: визначені існуючі рекомендаційні системи, виконано аналіз процесу роботи рекомендаційних систем, аналіз критеріїв та методів оцінювання рекомендаційних систем, побудовано удосконалену рекомендаційну систему за на основі колаборативної фільтрації, побудовано контекстну діаграму моделі «Дослідження методів оцінки рекомендаційних систем», проведено експериментальну перевірку удосконаленого методу.

РЕКОМЕНДАЦІЙНІ СИСТЕМИ, МЕТОДИ ОЦІНЮВАННЯ, ПРОЦЕС ОЦІНЮВАННЯ, КОЛАБОРАТИВНА ФІЛЬТРАЦІЯ, ВЛАСТИВОСТІ ОЦІНЮВАННЯ

ABSTRACT

Explanatory Note to master certification work contains: 77 pages, 4 sections, 12 pictures, 7 tables, 22 sources.

The aim of the study is to increase the efficiency of building recommendations on the basis of iterative evaluation of the results of recommendation systems.

In this work, the methods of evaluation of the recommendation systems are investigated. Existing evaluation methods have been analyzed, and based on the analysis, improvements in the evaluation of recommendation systems based on collaborative filtering based on missing data with implicit feedback by sorting negative results in the output matrix are proposed.

The study obtained the following results: identified existing recommender systems, the analysis of the process of recommender systems, analysis of criteria and methods of evaluation of recommendation systems based advanced recommendation system based on collaborative filtering, built contextual model diagram "Study of assessment methods of recommender systems", carried out experimental verification of the improved method.

RECOMMENDATION SYSTEMS, EVALUATION METHODS, EVALUATION PROCESS, COLLABORATIVE FILTRATION, EVALUATION PROPERTIES

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ ТА ТЕРМІНІВ

IDF – Inverse Document Frequency;

LSA – Latent semantic analysis;

RMSE - Root Mean Square Error;

MAE – Mean Absolute Error;

RMS – Root Mean Squared Error;

SADT – Structured Analysis and Design Technique;

RDD — Resilient Distributed Dataset (стійкий розподілений набір даних)

IDEF0 — Function Modeling, методологія функціонального моделювання, яка за допомогою наочної графічної мови представляється у вигляді набору взаємозалежних функцій;

ІС – інформаційна система;

КФ – колаборативна фільтрація

ЗМІСТ

Вступ 8

1. Аналіз методів оцінки рекомендаційних систем і постановка задачі ... 10

1.1 Дослідження характеристик рекомендаційних систем 10

1.2 Аналіз процесу роботи рекомендаційних систем 13

1.3 Аналіз критеріїв та методів оцінювання рекомендаційних систем 22

1.4 Постановка задачі дослідження. 27

2. Дослідження методів оцінювання рекомендацій з використанням явного та неявного зворотнього зв'язку 29

3 Структурно-функціональне моделювання процесу оцінювання рекомендаційних систем 46

3.1. Система оцінювання рекомендаційних систем 46

4 Удосконалення оцінювання рекомендаційних систем на основі колаборативної фільтрації 49

4.1 Реалізація удосконаленого методу 49

4.2 Експериментальна перевірка удосконаленого методу 54

Висновки 60

Список літератури 61

Додаток А. Графічний матеріал 64

ВСТУП

З появленням інтернету кількість інформації, з якою люди щодня стикаються, значно зросла. Це означає, що люди повинні орієнтуватися серед надзвичайно великої кількості доступних альтернатив, коли хочуть щось знайти. Але з іншого боку виступають власники інтернет-магазинів і сервісів: вони зацікавлені в персональній рекламі і рекомендаціях кожному конкретному користувачеві, тому що такий підхід може значно збільшити прибуток компанії. Як результат, в останні роки інтерес до розробки і поліпшення існуючих рекомендаційних систем значно виріс.

Рекомендаційні системи – це програми, головна мета яких полягає у формуванні рекомендацій різних продуктів або сервісів для користувачів на основі їх переваг.

Сервіси збирають інформацію про переваги користувачів і намагаються запропонувати їм корисні товари. На даний момент існує множина методів для формування рекомендацій, але всі вони мають свої переваги і недоліки. Саме тому дослідження в даній області актуальні. Проблема особливо актуальна для нових, щойно створених систем. Проблема релевантності оцінок часто виникає в разі холодного старту, оскільки нові об'єкти або користувачі ускладнюють створення релевантних рекомендацій.

Мета роботи – підвищення ефективності оцінки рекомендаційних систем шляхом удосконалення методу оцінювання рекомендацій.

У першому розділі було проведено аналіз методів оцінки рекомендаційних систем, описана предметна область, виконаний огляд існуючих методів побудови рекомендаційних систем. Також було проведено аналіз критеріїв та властивостей оцінювання рекомендаційних систем.

Другий розділ включає в себе методи оцінювання рекомендацій з використанням явного і неявного зворотнього зв’язку. Було описано принцип роботи рекомендаційних систем та проведен їх порівняльний аналіз. Також був досліджений метод колаборативної фільтрації на основі подібності елементів

Третій розділ присвячений структурно-функціональному моделюванню процесу оцінювання рекомендаційних систем, було побудовано контекстну діаграму моделі «Дослідження методів оцінки рекомендаційних систем» та її декомпозицію.

У четвертому розділі було удосконалено оцінювання рекомендаційних систем на основі колаборативної фільтрації за допомогою сортування негативних результатів в матриці вихідних даних та пропущених даних з неявним зворотним зв’язком.

1. АНАЛІЗ МЕТОДІВ ОЦІНКИ РЕКОМЕНДАЦІЙНИХ СИСТЕМ І ПОСТАНОВКА ЗАДАЧІ

1.1 Дослідження характеристик рекомендаційних систем

Перш за все, необхідно дати визначення рекомендаційної системи. Рекомендаційні системи – це програми, головна мета яких полягає у формуванні рекомендацій різних продуктів або сервісів для користувачів на основі їх переваг.

З появленням інтернету кількість інформації, з якою люди щодня стикаються, значно зросла. Це означає, що люди повинні орієнтуватися серед надзвичайно великої кількості доступних альтернатив, коли хочуть щось знайти. Але з іншого боку виступають власники інтернет-магазинів і сервісів: вони зацікавлені в персональній рекламі і рекомендаціях кожному конкретному користувачеві, тому що такий підхід може значно збільшити прибуток компанії. Як результат, в останні роки інтерес до розробки і поліпшення існуючих рекомендаційних систем значно виріс.

В наші дні рекомендаційні системи вже досить поширені і мають велику кількість застосувань. В першу чергу, рекомендаційні системи використовуються в інтернет-комерції для того, щоб допомогти користувачам вибрати відповідні товари. Такі сервіси збирають інформацію про переваги користувачів і намагаються запропонувати їм корисні товари. На даний момент існує множина методів для формування рекомендацій, але всі вони мають свої переваги і недоліки. Саме тому дослідження в даній області актуальні.

Якість рекомендаційної системи залежить від багатьох властивостей вироблюваних нею рекомендацій, які зазвичай оцінюються з допомогою показників продуктивності, що відображають вузьке уявлення про поведінку системи. Найчастіше розробники оцінюють точність рекомендацій за допомогою метрик, які вимірюють здатність рекомендаційної системи точно представляти набір відомих переваг. Точність вимірюється шляхом

резервування частини набору даних рейтингів у вигляді набору відомих уподобань і використання інших рейтингів для навчання рекомендаційної системи і вироблення рекомендацій. Відомі переваги можуть використовуватися деякими метриками, такими як точність або відгук, для обчислення оцінки, що представляє точність алгоритму рекомендації для цього набору даних.

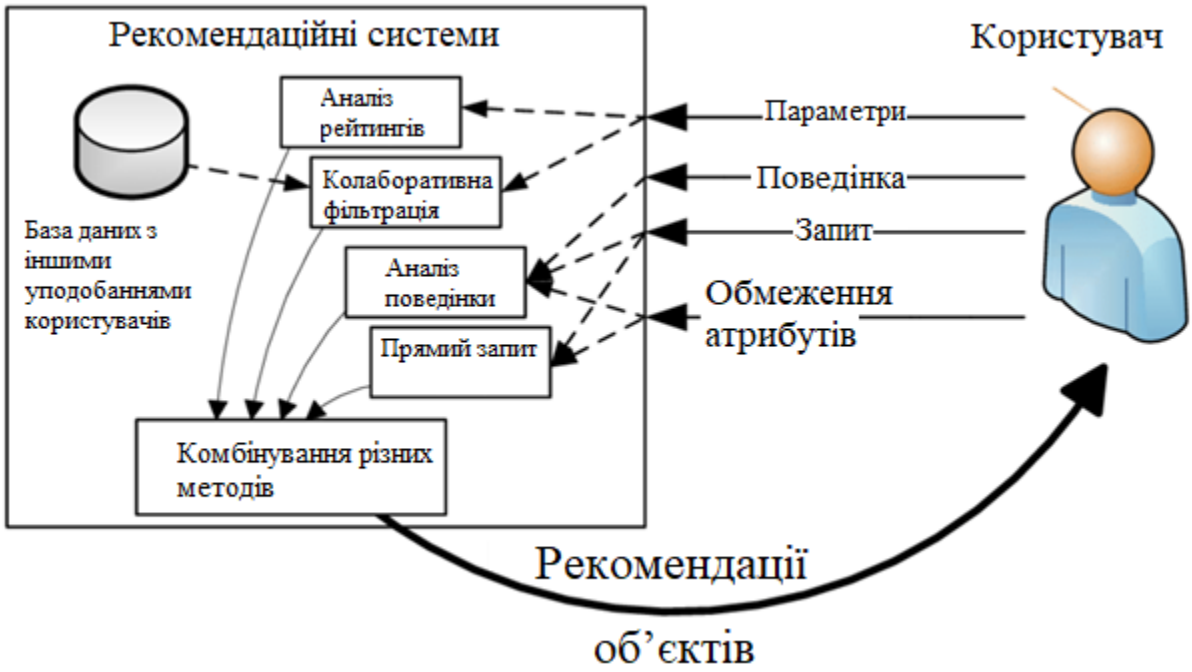

Рис.1.1 – Структура рекомендаційної системи

Однак метрики точності дають тільки грубе, одномірне уявлення про продуктивність рекомендаційної системи. Зокрема, точність і відгук обмежені декількома способами. По-перше, вони дають мало інформації про поведінку предметів з невідомими уподобаннями. По-друге, коли набір відомих вподобаних елементів малий, значення зазвичай низькі через малу ймовірність включення рекомендованих елементів в набір відомих переваг. Оскільки точність і відгук не передають ніякої інтуїції про поведінку предметів, які не входять в набір відомих уподобань, висновки про відносну перевагу однієї системи над іншою, можуть вводити в оману.

Рекомендації формуються окремо для кожної людини, спираючись на його попередні дії на конкретному веб-ресурсі чи на основі минулої активності. Крім того, значення має і поведінка попередніх учасників процесу.

Для інтернет-магазинів – це важлива функція, а для таких великих каталогів типу Amazon – один з небагатьох способів якісно працювати. Спосіб рекомендації в даному випадку не є звичайною додатковою опцією, вона забезпечує зручність навігації користувача по веб-ресурсу. Якщо електронний каталог містить понад 20 000 найменувань продукції, орієнтація вже видається непомірно важкою, що говорити, якщо товарів мільйони?

Наскільки стомлює потенційного покупця взаємодія з подібним сайтом? Відповідь очевидна. І на допомогу приходить віджет пошуку товарів, візуально схожих на шуканий, або належить одній групі виробів, або компліментарна продукція (коли до парі туфель пропонують вибрати сумочку, наприклад). Таке рішення збільшує не тільки число переглядів, це позитивно впливає на конверсію.

Як показує практика, не тільки онлайн-магазини використовують подібний прийом. Також подібні прийоми легко можна побачити на різних соціальних платформах, порталах, присвячених літературі, подорожам, на новинних ресурсах, в інтернет-магазинах, словом – майже скрізь. Ця методика дійсно дуже популярна. Рекомендаційні сервіси збирають різну інформацію про людину, використовуючи кілька методів, за якими і поділяють всі системи.

Отже, перший тип – явний збір даних. Як можна було здогадатися з назви, користувач сам надає необхідні для роботи матеріали. Наприклад, коли рекомендаційні системи Google або інших пошуковиків просять людину дати оцінки різним елементам, скласти список фаворитів певної сфери або ж відповісти на кілька питань. Якщо ж людина відмовляється дати інформацію самостійно, актуальною буде наступна методика.

Другий тип – неявний збір даних. Це шпигунська місія, згідно з якою дії учасника процесу фіксуються програмою для подальшої обробки і застосування. Програма розпізнає покупки, оцінки на сайтах, збирає інформацію про перегляди, коментарі. Звичайно, вибір такої методики веде за

собою деякі етичні проблеми, адже захист персональних даних – одна з головних вимог, користувачів до пошукових сайтів. Але поки факт залишається фактом – своєрідне стеження можливо, і відвідувачі сайтів перевірити, чи дійсно ведуться подібні заходи, не можуть.

Існують також види рекомендаційних систем, що визначаються за підходами, які вони застосовують.

1.2 Аналіз процесу роботи рекомендаційних систем

Дані про відгуки збираються з різних причин, не тільки для машинного навчання. Кожна навчальна система має специфічні вимоги до того, як дані повинні бути представлені для аналізу. Вибір конкретних даних для вивчення може сильно вплинути на модель навчання, з цих причин підготовка даних є важливою частиною для будь-яких цілей машинного навчання. Підготовка даних часто є найбільш трудомісткою частиною будь-якого проекту машинного навчання. Вибір конкретного типу фільтрації або комбінації з декількох методів безпосередньо залежить від двох факторів – складності проекту і розміру його фінансування. Наприклад, створити алгоритм для системи з тематичних, пересічних один з одним блогів-завдання відносно проста і помірно витратна. Більш масштабні і неоднорідні проекти, такі як онлайн-магазини, вимагають великих витрат, особливо в тому випадку, якщо стоїть завдання збільшити конверсію на дійсно значущі величини. Як правило, в таких проектах не виходить обмежуватися одним видом рекомендаційного алгоритму і доводиться використовувати гібридну фільтрацію, в результаті чого вартість і складність розробки збільшується на порядки.

Так чи інакше, розробляючи проект, що пропонує користувачеві можливість вибирати конкретні об'єкти із загальної множини, необхідно враховувати стрімкий прогрес юзабіліті абсолютно у всіх сферах людського життя – від оптимізації сну за допомогою приладів, які аналізують всі що протікають у сні процеси і видають рекомендації щодо його поліпшення, до

автоматичного підбору повсякденних товарів виходячи з поточних потреб користувача.

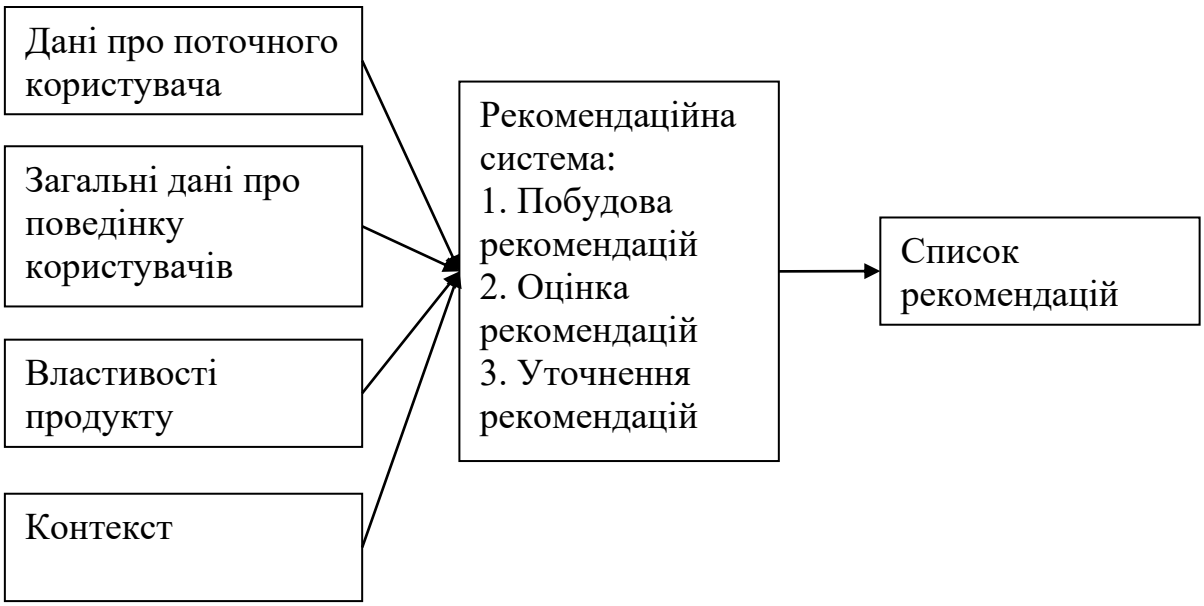

Рис. 1.2 – Процес роботи рекомендаційних систем

На сьогоднішній день при створенні рекомендаційних систем використовуються дві основні стратегії: фільтрація вмісту та колаборативна фільтрація. Варто відзначити, що на практиці зазвичай використовуються гібридні методи, що поєднують в собі переваги приведених нижче підходів.

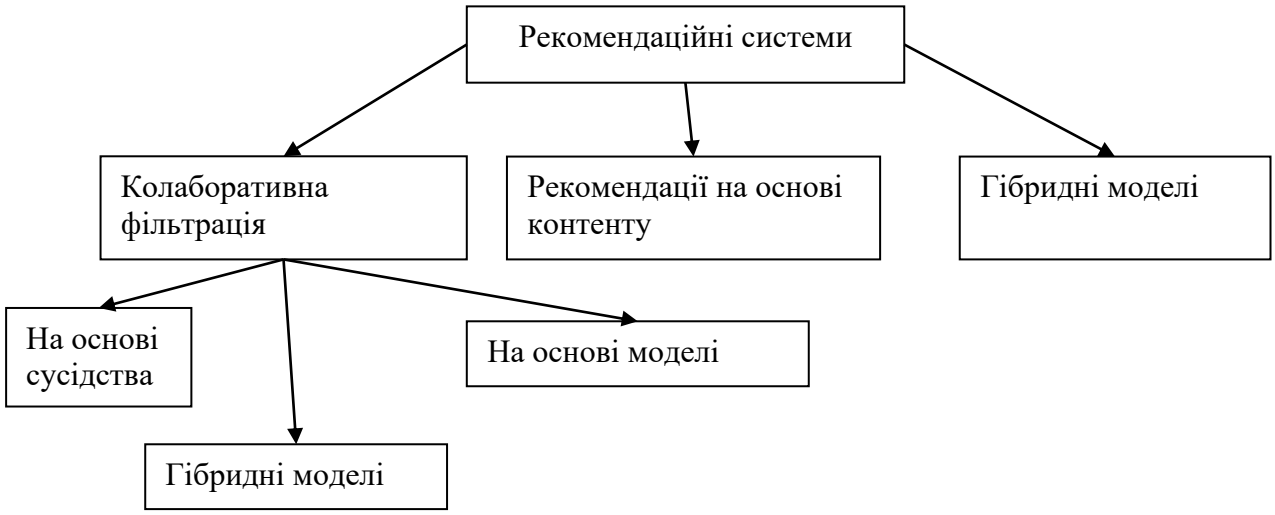

Рис.1.3 – Класифікація рекомендаційних систем

Фільтрація вмісту (Рис. 1.4) ґрунтується на створенні профілю користувача і профілю об'єкта. Необхідно враховувати параметри об'єктів і їх відповідність перевагам користувача. Для цього в рекомендаційних системах використовуються теги (ключові слова), щоб описати об'єкти, а профіль користувача відображав оцінку певних тегів або їх сукупність.

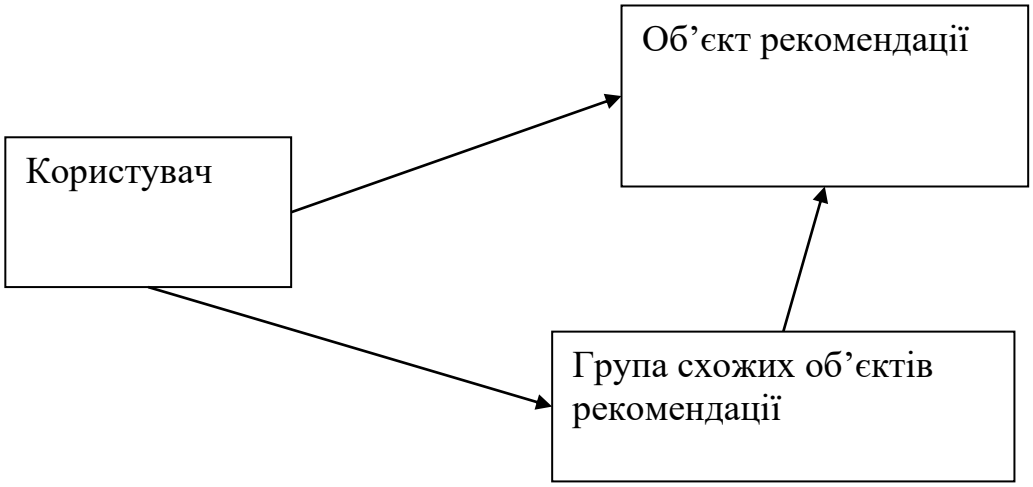

Рис. 1.4 – Фільтрація вмісту

У рекомендаційних системах з фільтрацією вмісту функція задоволеності h(u, s) користувача u деяким об'єктом s визначається, ґрунтуючись на відомостях про задоволеність користувача об'єктами $s_i \in S$, які мають схожість з s. Тобто, якщо рекомендаційну систему використовувати в веб-додатку музичного аудіо-стрімінга, то для того щоб рекомендувати користувачеві u музичні композиції, система повинна визначити, що пов'язує різні музичні композиції, яким користувач раніше дав високу оцінку. Потім система буде рекомендувати користувачеві музичні композиції з максимальною відповідністю тим, які були високо оцінені користувачем раніше. Призначений для користувача профіль в системі формується із споживаного користувачем контенту у вигляді множини параметрів, які визначають об'єкт s. Такими параметрами зазвичай є ключові слова і відповідні їм ваги для кожного об'єкта. Отже, необхідно визначити ваги цих параметрів. В інформаційному пошуку

одним з найбільш відомих способів визначення ваги ключових слів є TF-IDF міра. Припустимо, що N – підсумкова кількість об'єктів, які мають ймовірність бути рекомендованими користувачеві, а також, що ключове слово k_j зустрічається в n_i об'єктах, а $f_{i,j}$ - кількість входжень цього слова в об'єкт d_j. Тоді $TF_{i,j}$ (term frequency) – це відношення числа входжень тега до загальної кількості тегів об'єкта, тобто:

$$TF_{i.j} = \frac{f_{i,j}}{max_z f_{z,j}}. \tag{1.1}$$

Але якщо враховувати тільки частоту входження тега, то в більшості об'єктів максимальна вага буде у найпоширеніших тегів, що, найімовірніше, призведе до неправильного оцінювання переваг користувача. Для уникнення подібного застосовується IDF_i (inverse document frequency) – величина, зворотна частоті входження тега в об'єкт колекції. Визначаємо її як:

$$IDF_i = log \frac{N}{n_i}. \tag{1.2}$$

Таким чином, вага $w_{i,j}$ у ключового слова k_i в об'єкті d_j позначається як добуток частоти входження тега на зворотну частоту об'єкта:

$$w_{i.j} = TF_{i,j} \times IDF_i, \tag{1.3}$$

В такому випадку контент об'єкту d_j можна визначити так:

$$Content(d_{j,i}) = (w_{1,j}, \dots, w_{k,j}). \tag{1.4}$$

Як зазначалося вище, рекомендаційні системи з фільтрацією вмісту пропонують об'єкти з урахуванням тих, які користувач високо оцінив раніше. Різні об'єкти порівнюються з тими, які сподобалися користувачеві і з них рекомендуються ті, які мають максимальну схожість. Сукупність об'єктів, які користувач оцінив раніше, утворює призначений для користувача профіль ContentBasedProfile(u) або, інакше кажучи, вектор ваг ($w_{u,1}$, ..., $w_{u,k}$), де кожна вага $w_{u,i}$ визначає важливість тега k_i для користувача u. Отже, ContentBasedProfile (u) і Content (s) можна уявити як TF-IDF вектори $\vec{wu}$ і $\vec{ws}$, при цьому функція задоволеності користувача h(u, s) може бути представлена

як косінусний коефіцієнт векторів $\vec{w}u$ і $\vec{w}s$, де K – спільна кількість тегів в системі:

$$h(u,s) = \cos(\vec{w}_u, \vec{w}_s) = \frac{\vec{w}_u \vec{w}_s}{\|\vec{w}_u\| \|\vec{w}_s\|} = \frac{\sum_{i=1}^{K} w_{i,u} w_{i,s}}{\sqrt{\sum_{i=1}^{K} w^2_{i,u}} \sqrt{\sum_{i=1}^{K} w^2_{i,s}}}, \quad (1.5)$$

Крім евристики, заснованої здебільшого на методах інформаційного пошуку, достатньо часто використовуються і інші техніки рекомендацій з фільтрацією вмісту, наприклад, простий байесовский класифікатор, різні техніки машинного навчання, в тому числі нейронні мережі, дерева рішень і кластеризація. Але на відміну від методів інформаційного пошуку, ці техніки намагаються передбачити задоволеність користувача, використовуючи в основі евристику косинусного коефіцієнта. Наприклад, у музичному аудіо-стрімінзі можна, використовуючи інформацію про музичні композиції які сподобались та ні, через наївний байесовский класифікатор виділити композиції, які користувач ще не чув і оцінити ймовірність належності композиції p_j до певного класу C_i (вподобані та ні).

Колаборативна фільтрація (Рис. 1.5) використовує відомі переваги групи користувачів, до якої входить той користувач, для якого потрібно спрогнозувати рекомендації. Головна підстава такого підходу в тому, що користувачі однаково оцінили деякі об'єкти в минулому, найімовірніше, однаково оцінять інші об'єкти в майбутньому.

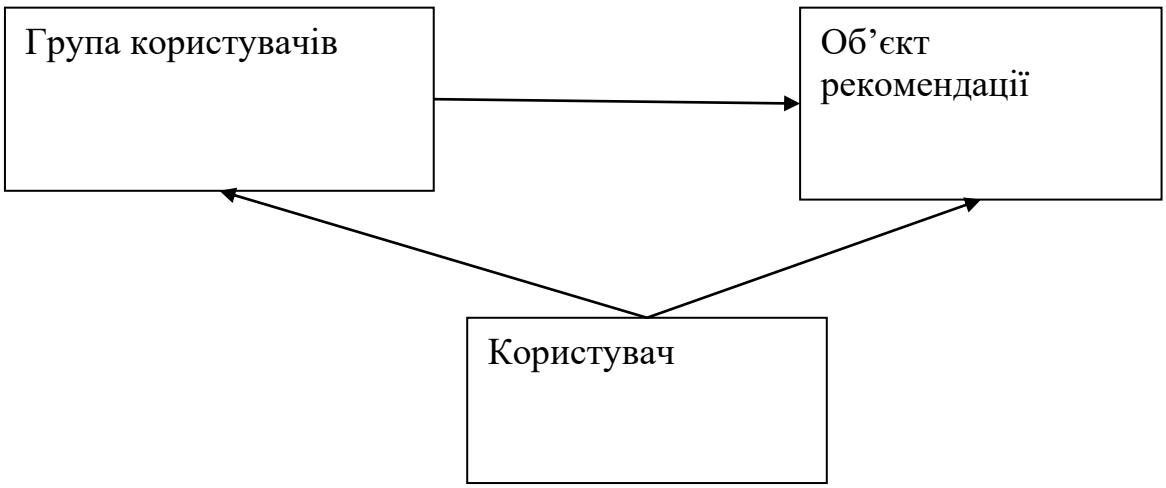

Рис. 1.5 – Колаборативна фільтрація

На відміну від фільтрації вмісту, методи колаборативної фільтрації намагаються передбачити чи буде відповідати перевагам користувача той чи інший об'єкт, грунтуючись на оцінках інших користувачів однієї групи, тобто отримуємо оцінку задоволеності h (u, s) деяким об'єктом s для користувача u, яка обчислюється залежно від задоволеності h (u_j, s) тим же об'єктом s користувачів u_j є U, які мають схожі ознаки з користувачем u. Варто відзначити важливу особливість даного методу: ознаки схожості одного користувача на інших не завжди є частиною даної системи. Тобто, в системі музичного аудіо-стрімінга призначені для користувача групи можуть визначатися не тільки на підставі музичних уподобань користувачів, але і грунтуючись на більш загальних ознаках, таких як географічне положення, профілі в соціальних мережах та інших параметрах. Подібного роду інформація може служити для первинної оцінки схожості нового користувача на вже зареєстрованих в системі і може застосовуватися для вирішення поширеною для колаборативної фільтрації проблеми «холодного старту».

Існує досить велика кількість user-based рекомендаційних систем, розроблених за участю як вчених, так і комерційних структур різних сфер діяльності. Системою такого роду є «Grundy system» - програм бібліотекар задає користувачам питання на різну тематику, складаючи таким чином призначений для користувача профіль і рекомендуючи книги, які могли б припасти до смаку користувачеві. Якщо користувач відповідав, що запропонована книга йому не цікава, то система починала з'ясовувати причину, поглиблюючи тим самим інформацію про профілі. Таким чином, «Grundy system» встановила для кожного користувача персональну модель, завдяки якій могла вельми точно радити цікаву літературу. В той самий час інша система, звана «Tapestry», замість визначення схожості через питання, пропонувала користувачеві самому перерахувати користувачів з відповідним вибором переваг.

Також, існує думка, що методи колаборативної фільтрації вмісту діляться на дві основні категорії: засновані на пам'яті і засновані на моделях.

Перша категорія алгоритмів намагається передбачити рейтинг об'єкта, використовуючи знання про всі об'єкти, які користувач встиг оцінити раніше. Величина рейтингу $r_{u,s}$ для користувача u і об'єкта s обчислюється, як агреговане значення оцінок інших найбільш схожих N користувачів для цього самого об'єкта s:

$$r_{u,s} = \text{aggr}_{u\epsilon\hat{U}}\, r_{u',s}, \tag{1.6}$$

де U позначає множина N користувачів, які оцінювали об’єкт s і найбільш схожі на користувача u. В якості найпростішого прикладу функції агрегування можна привести обчислення середнього:

$$r_{u,s} = \frac{1}{N}\sum_{u\epsilon\hat{U}} r_{u',s}. \tag{1.7}$$

Найпоширенішим випадком агрегування є:

$$r_{u,s} = k\sum_{u\epsilon\hat{U}} \text{sim}(u,u') \times r_{u',s}, \tag{1.8}$$

$$r_{u,s} = \bar{r}_u + k\sum_{u\epsilon\hat{U}} \text{sim}(u,u') \times (r_{u',s} - \bar{r}_{u'}), \tag{1.9}$$

де k застосовується для нормалізації і, як правило, a sim (u, u') – міра «схожості», яка є зворотною відстані і в більшості випадків береться в якості ваги. Таким чином, чим більше користувачі u і u' схожі, тим більший буде врахована вага при обчисленні $r_{u,s}$. Ми можемо бачити, що в різних рекомендаційних системах, цілком можливо, можуть бути застосовані різні заходи схожості. Давайте розберемо дві з них, які зустрічаються найбільш часто. У випадку із застосуванням зважених сум є кілька складнощів, наприклад, нездатність враховувати, що різні користувачі не завжди оцінюють об'єкти з однаковим рівнем строгості.

Існує величезна кількість модифікацій, які вдосконалюють ефективність роботи даних методів. Наприклад, зумовлені рейтинги (default voting) представляють одну з таких модифікацій згаданого вище методу, який заснований на пам'яті.

В той же час, коли розібрані вище підходи застосовувалися для обчислення схожості користувачів, B. Sarwar застосували ці ж самі підходи для

визначення кореляції між об'єктами, щоб обчислювати оцінки для них. Пізніше ця думка була вдосконалена до top-N алгоритму. Також, існує емпіричне підтвердження того факту, що алгоритми з фільтрацією вмісту здатні працювати з більшою продуктивністю щодо витрачених обчислювальних ресурсів, при цьому забезпечуючи не менше точні рекомендації, ніж більшість алгоритмів з колаборативною фільтрацією.

Деякі рекомендаційні системи використовують комбінації підходів фільтрації вмісту і колаборативної фільтрації, звані гібридними методами. Вони дозволяють в тій чи іншій мірі уникнути недоліки обох підходів. Є кілька основних варіантів комбінування різних методів в рекомендаційних системах гібридного типу, вони указані на рисунку 1.6.

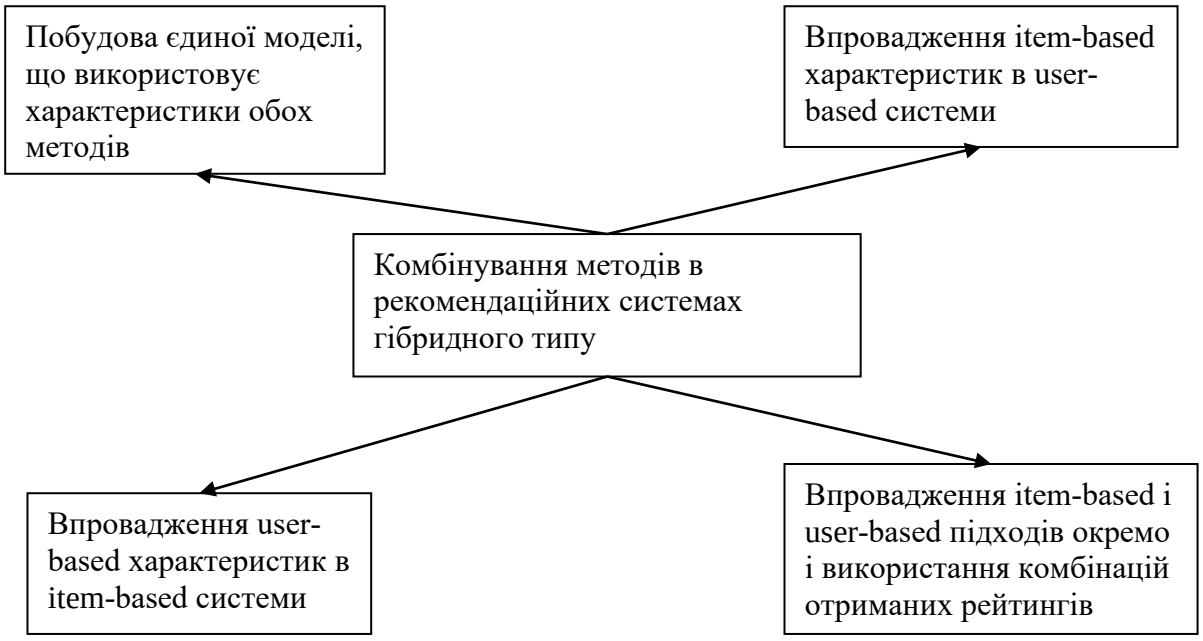

Рис. 1.6 – Комбінування методів в рекомендаційних системах гібридного типу

Останнім часом даний варіант був дуже поширений серед великої кількості дослідників. Основний принцип методу в тому, щоб використовувати характеристики фільтрації вмісту і колаборативної фільтрації (наприклад, жанр музичної композиції і професію користувача) в одній рекомендаційній системі.

До таких підходів належить уніфікований імовірнісний метод, який застосовує латентно-семантичний аналіз.

Такі гібридні рекомендаційні системи побудовані на колаборативній складовій, але також включають в призначений для користувача профіль деякі дані фільтрації вмісту. Після цього дані служать підставою для обчислення схожості користувачів замість загальних оцінених об'єктів. Такий підхід дозволяє уникнути проблеми з недостатньою кількістю даних, що виникає через малу кількість пар користувачів з досить великим числом загальних оцінених об'єктів. Також, метод дозволяє рекомендувати користувачеві не тільки об'єкти з позитивними відгуками від інших користувачів, а й ті об'єкти, які користувачеві можуть сподобатися виходячи з його особистих переваг.

Одним з найбільш відомих методів даної групи гібридних рекомендаційних систем є зменшення розмірності профілів, які використовують фільтрацію вмісту. В одному з таких методів використовується латентно-семантичний аналіз (LSA) для створення колаборативного змісту профілів користувачів, які представлені сукупністю векторів. Такий підхід значно покращує продуктивність у порівнянні з системами, що використовують тільки фільтрацію вмісту.

Одна комбінація гібридної рекомендаційної системи полягає в роздільному впровадженні item-based і user-based підходів. Далі, є два шляхи розвитку подій. У першому випадку можна комбінувати оцінки від двох систем, використовуючи лінійну комбінацію рейтингів. Другий шлях пропонує використовувати ту систему, яка повинна краще підійти в конкретному випадку. Існує система «DailyLearner», яка застосовує рекомендаційну систему, що дає рекомендації з мінімальним рівнем помилки. Гібридні рекомендаційні системи іноді доповнюють методами, які використовують базу знань для поліпшення якості рекомендацій і розв’язання основних проблем більшості рекомендаційних систем (холодний старт і інші). Існують роботи різних авторів, де порівняння продуктивності показує перевагу гібридних підходів над чистими item-based і user-based рекомендаційними системами.

1.3 Аналіз критеріїв та методів оцінювання рекомендаційних систем

Існує множина різних критеріїв за якими можна оцінювати рекомендаційну систему – такі як точність, новизна, можливість дивувати, стійкість до атак, залежність від холодного старту, переконливість і так далі, але одним з найбільш важливих все-таки є точність. Вона показує на скільки наші передбачення близькі до еталонного результату. Для вимірювання точності є множина методів, рекомендую підійти до вибору акуратно, так як від цього багато залежить. Один з найпопулярніших методів – розрахунок середньоквадратичної помилки (RMSE). Після того як на основі тестових даних алгоритм виробляє передбачення, помилку можна обчислити за такою формулою:

$$\mathrm{RMSE} = \sqrt{\frac{1}{|\mathcal{T}|} \sum_{(u,i)\in\mathcal{T}} (p_{ui} - r_{ui})^2} \tag{1.10}$$

U – користувач;

i – предмет;

r – оцінка;

p – прогнозована оцінка;

T – загальна кількість тестових оцінок, чим менше тим краще;

Дослідники вивчили властивості рекомендаційних систем, які можуть бути корисні розробникам за межами точності рекомендацій, а також метрики для їх захоплення. Кілька таких властивостей, часто з відповідними метриками, були ідентифіковані в літературі рекомендаційних систем, таких як охоплення, різноманітність, новизна і адаптивність. Наприклад, різноманітні рекомендації, в яких елементи відрізняються один від одного, можуть забезпечити більшу цінність для користувача. Якщо відомо, що користувачеві подобаються науково-фантастичні фільми, рекомендація всіх фільмів Star Wars може бути точною, але вона не різноманітна і може бути незадовільною для користувача.

Друге завдання, на якому ми зосереджені – це оцінка стабільності. Стабільність рекомендаційної системи вимірює узгодженість рекомендацій після внесення змін в набір даних. Розглянемо рекомендаційну систему, яка

пропонує фільм Interstellar в якості найбільш рейтингового елемента для 1% користувачів. Якщо якийсь користувач додає новий рейтинг для Interstellar і в результаті він більше не рекомендується нікому, то рекомендаційна система може вважатися нестабільною і негативно впливати на довіру користувачів.

Ми визначаємо загальну модель рекомендаційних систем, яку потім використовуємо для формального визначення та систематичного дослідження властивостей простору рекомендаційних систем.

Для поліпшення продуктивності веб-сайтів може бути використано кешування. Воно дозволяє не генерувати часто запитувані сторінки кожен раз заново, а використовувати збережені версії. Це дає можливість значно скоротити час завантаження сторінки і навантаження на сервер. Наприклад, система управління базою даних може зберігати результат часто виконуваних запитів, окремі фрагменти інформації можуть завантажуватися в оперативну пам'ять сервера, HTML-код сторінок або їх невеликих блоків може бути витягнутий з файлової системи веб-сервера, сторінки цілком можуть бути закешовану інтернет-провайдером, і т. д.

Для того щоб кешування було ефективним, інформація повинна використовуватися багаторазово і часто. Але персоналізовані сторінки можуть містити дані, що відносяться до конкретного користувача. Веб-сайт ідентифікує відвідувача по cookie-файлах і генерує сторінки спеціально для нього. Такі сторінки не можуть бути закешовані цілком. Навіть якщо сторінка буде закешована, ймовірність того, що вона буде повторно завантажена з кешу, як правило, досить низька, тому в результаті значно виросте обсяг використовуваної пам'яті на жорсткому диску сервера, в той час як навантаження на процесор знизиться незначно.

Розв’язанням даної проблеми може бути виділення всіх персоналізованих блоків в окремі URL-адреси і їх динамічне підвантаження за допомогою технології AJAX.

Крім проблеми, пов'язаної з неможливістю кешування сторінок, існують труднощі, викликані специфікою вихідних даних. Кожному користувачеві відповідають пари «документ-оцінка», де документом може бути сторінка сайту

або певна тема (тег), а оцінкою – деяка міра інтересу користувача до неї, наприклад, число переглядів.

Можна уявити собі ці дані як матрицю, кожен рядок якої відповідає користувачеві, а стовпець – документу. В такому випадку завдання зводиться до обчислення невідомих елементів матриці. Матриця рейтингів, як правило, дуже розріджена - і користувачів, і сторінок багато, а рейтингів на практиці набагато менше, ніж їх твір, адже середній користувач відвідує небагато сторінок. Інші елементи матриці невідомі, і їх значення треба передбачити, тобто кількість оцінок, які необхідно передбачити, набагато перевищує кількість відомих оцінок. Важливо, щоб система вміла ефективно передбачити оцінки, виходячи з невеликої кількості прикладів. Також необхідна наявність критичної кількості користувачів. Подолати проблему розрідженості оцінок можна, якщо при пошуку схожих користувачів використовувати інформацію про користувача, що міститься в його профілі.

Колаборативна фільтрація по користувачах є одним з найбільш ефективних рекомендаційних алгоритмів, але має низку недоліків. Серед них, в тому числі, неможливість генерувати рекомендації для нових користувачів і погана масштабованість.

Перша проблема може бути вирішена за допомогою використання колаборативної фільтрації по схожості сторінок. Для використання колаборативної фільтрації на високонагруженних сайтах з великою кількістю сторінок доцільно проводити розрахунки схожості користувачів паралельно в кілька потоків. Швидкість складання рекомендацій може бути підвищена шляхом зниження розмірності вихідних даних. Значно поліпшити продуктивність дозволить попереднє виділення груп користувачів зі схожою поведінкою. При цьому дистанцію між двома користувачами, які входять в різні групи, можна буде вважати нескінченною. Для цього може бути використано попереднє виділення груп схожих користувачів і виконання алгоритму колаборативної фільтрації всередині груп. Це дозволить вирішити проблему з розрідженістю вихідних даних і поліпшити масштабованість.

Для скорочення числа операцій по складанню рекомендацій можна навчати алгоритм не на всіх даних про користувачів, а на деякій релевантній вибірці. Виділення підмножин з вихідних даних може бути також використано для створення навчальної та перевірочної вибірок. В цьому випадку параметри алгоритму налаштовуються на основі даних з першої вибірки, а точність рекомендацій оцінюється на основі решти даних. Головним завданням при формуванні вибірки є пошук релевантної підмножини у вихідних даних. Найбільш простим способом є використання випадкової вибірки, де у всіх елементів вихідних даних рівна ймовірність попадання у вибірку. У деяких випадках виправдане використання стратометричгого відбору, коли генеральна сукупність ділиться на групи, що володіють певними характеристиками (стать, вік, політичні уподобання, освіта, рівень доходів тощо.), і відбираються елементи з відповідними характеристиками. Як правило, використовуються вибірки без повторень - один і той же елемент не може бути обраний двічі. Широко поширений підхід, при якому застосовується випадкова вибірка без повторів з виділенням тренувального і тестового набору даних в пропорції 80/20. Процес навчання алгоритму може повторюватися кілька разів, щоб уникнути помилок, пов'язаних з використанням невеликої підмножини замість всіх вихідних даних. Після створення навчальної та тестової вибірок налаштовується модель рекомендаційної системи, а потім оцінюється її точність. Далі створюється нова пара вибірок, і процес навчання і тестування повторюються k разів. В результаті використовується усереднене значення k моделей. Такий підхід називається крос-валідація. Вона може здійснюватися кількома методами. У випадку з повторюваними випадковими вибірками підмножини для навчання вибираються k раз поспіль випадковим чином. Іншим способом може бути виділення n підмножин вихідних даних. Одне з них використовується як тестове, а решта – як навчальні. Процес навчання і тестування повторюються n разів так, що кожна підмножина виявляється тестом один раз.

Відвідувачів і сторінки сайту можна класифікувати так, що вихідними даними буде інформація веб-перегляду, а результатом – клас відвідувачів, до

якого належить конкретний користувач. Існує множина способів класифікації, серед яких можна виділити два типи – з навчанням і без нього. В алгоритмах класифікації з навчанням заздалегідь відомі класи і є дані про належність частини елементів до класів. Ґрунтуючись на цих даних, можна навчити алгоритм і класифікувати інші елементи. Якщо класи заздалегідь не відомі, то потрібно розділити всі елементи на групи так, щоб елементи, що входять в одну групу, були схожі між собою, але не схожі на елементи з інших груп.

Алгоритм k найближчих сусідів дозволяє розділити вихідні дані на непересічні класи, використовуючи інформацію про схожість окремих елементів. Користувач, клас якого ще ніхто не знає, буде віднесений до того класу, до якого відносяться k найближчих до нього інших користувачів. Одна з основних переваг цього методу полягає в тому, що при додаванні нових користувачів не потрібно перенавчати алгоритм на всіх даних – досить визначити найбільш відповідні класи тільки для нових відвідувачів. Але при класифікації доданих користувачів потрібно попарно порівняти їх вектори з даними про всіх вже наявних відвідувачів, що може зажадати великого обсягу пам'яті для обчислень і займе багато часу. Для високонавантажених веб-сайтів це може бути неприпустимо повільно.

Для виділення груп схожих користувачів може бути використаний Байєсівський класифікатор. Даний метод заснований на принципі максимізації апостеріорної ймовірності. Для користувача обчислюються функції правдоподібності кожного з класів і по ним обчислюються апостеріорні ймовірності класів. В результаті користувач відноситься до того класу, для якого апостеріорна ймовірність максимальна. Визначення класу може бути спрощено, якщо вважати, що всі параметри, що характеризують поведінку користувача і утворюють його вектор, незалежні один від одного.

Завдання функціонування рекомендаційної системи можна сформулювати як пошук N найцікавіших користувачеві посилань, а значить оцінити ефективність рекомендаційної системи, заснованої на класифікації, можна перевіривши, наскільки точно вона класифікує сторінки як цікаві для конкретного користувача. Таким чином, для подібних систем можна

застосовувати такі поняття, як точність і повнота. Для рекомендаційної системи точність – це частка цікавих користувачеві сторінок серед запропонованих в рекомендаціях. Повнота – частка відомих системою як цікаві користувачеві сторінок серед всіх існуючих.

1.4 Постановка задачі дослідження.

Задача дослідження – аналіз критеріїв та властивостей оцінювання рекомендаційних систем, опис принципу роботи рекомендаційних систем і їх порівняльний аналіз, аналіз принципу оцінювання рекомендаційних систем, дослідження методу колаборативної фільтрації на основі подібності елементів, створення діючої системи оцінювання рекомендаційних систем, сортування негативних результатів в матриці вихідних даних, удосконалення колаборативної фільтрації на основі пропущених даних з неявним зворотним зв’язком.

Актуальність – сервіси збирають інформацію про переваги користувачів і намагаються запропонувати їм корисні товари. На даний момент існує множина методів для формування рекомендацій, але всі вони мають свої переваги і недоліки. Саме тому дослідження в даній області актуальні. Проблема особливо актуальна для нових, щойно створених систем. Проблема релевантності оцінок часто виникає в разі холодного старту, оскільки нові об'єкти або користувачі ускладнюють створення релевантних рекомендацій.

Об’єкт дослідження – процес побудови рекомендацій.

Мета дослідження – підвищення ефективності побудови формування рекомендацій на основі ітеративної оцінки результатів роботи рекомендаційних систем.

Для того, щоб система мала високоякісні рекомендації, необхідно постійно оновлювати інформацію про інтереси користувача. Але існує й інша проблема – користувач завжди хоче мінімізувати час, що витрачається на взаємодію з системою, і неохоче ділиться інформацією для зворотного зв'язку з системою. Це типова проблема для всіх рекомендаційних систем, і оскільки

рекомендації засновані виключно на інформації профілю користувача, тобто чим менше інформації користувач надає, тим менше відповідний набір рекомендацій користувач отримає. Сучасні методи підготовки рекомендацій враховують властивості предметів а також користувачів, які вибирають ці предмети. Однак оцінювання ефективності цих методів виконується лише при їх реалізації. На практиці існує потреба у регулярній адаптації цих методів на основі оцінки їх ефективності. Це визначає актуальність теми дослідження.

2. ДОСЛІДЖЕННЯ МЕТОДІВ ОЦІНЮВАННЯ РЕКОМЕНДАЦІЙ З ВИКОРИСТАННЯМ ЯВНОГО ТА НЕЯВНОГО ЗВОРОТНЬОГО ЗВ'ЯЗКУ

2.1 Загальний підхід до оцінювання рекомендаційних систем і їх порівняльний аналіз

Рекомендаційної системи складається з алгоритму і набору даних D, як показано на рис. 2.1. Алгоритм рекомендації використовує набір даних для обчислення моделі користувацьких переваг. Потім ця модель може бути використана для обчислення рекомендацій для користувачів системи.

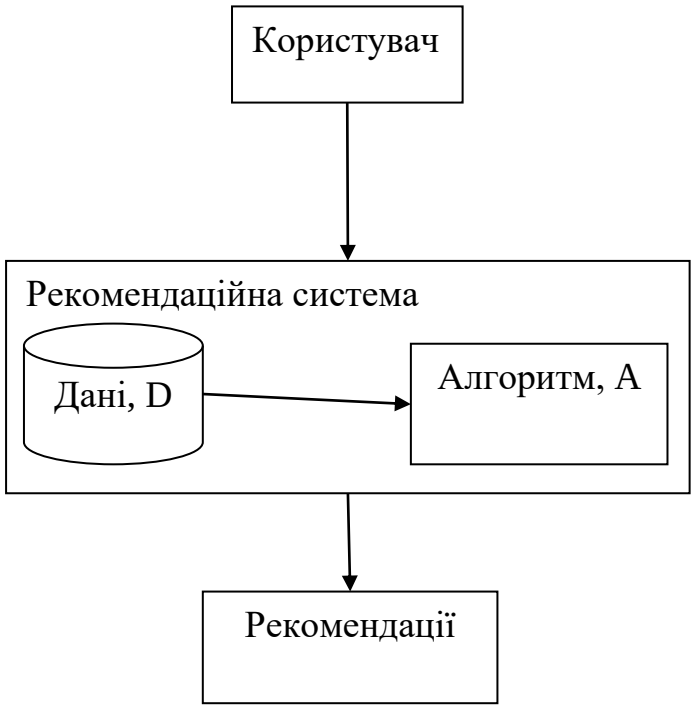

Рис. 2.1: Модель рекомендаційної системи

Діаграми класів моделі системи рекомендаційних показана на малюнку 2.2. Ця модель складається з користувачів, елементів і атрибутів, а також двох типів відносин між користувачами і елементами: рангів, які є відносинами між елементами і користувачами, створеними системою рекомендацій; і рейтингів, які є відносинами між користувачами і елементами, визначеними набором даних.

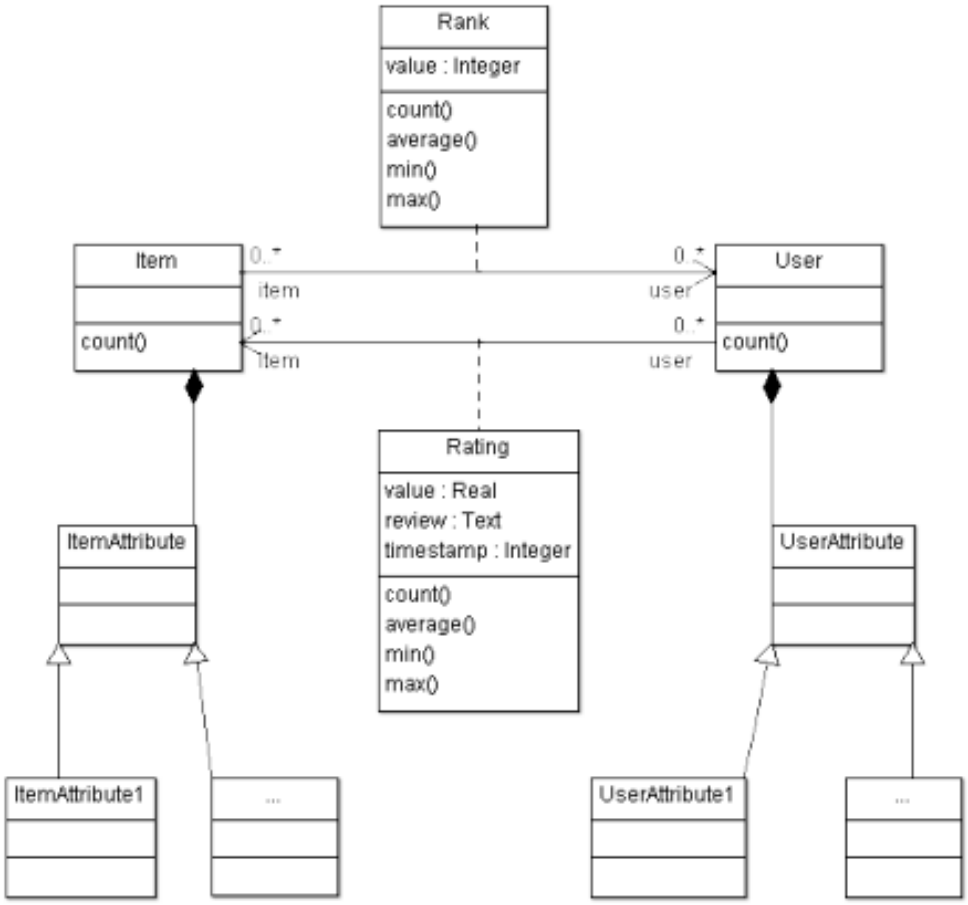

Рис. 2.2 - Схема класів рекомендаційної системи

Розглянувши основні типи рекомендаційних систем виділимо критерії і виконаємо порівняльний аналіз методів підбору рекомендацій.

На підставі проведеного порівняльного аналізу можна зробити висновок про недоцільність розробки гібридної рекомендаційної системи через високу складність реалізації і відсутності команди розробників, між якими можна було б розділити виконувані завдання. Колаборативна фільтрація реалізована в соціальних мережах і показує досить сумнівні результати. Також великими мінусами є проблема так званого холодно старту, коли систему необхідно навчити шляхом вираження користувацьких симпатій/антипатій до результатів, що видаються, і залежність алгоритму аналізу від інших користувачів. Таким чином, ми приходимо до двох можливих варіантів, які по суті аналізують дані про сам об'єкт – це контентна фільтрація і фільтрація заснована на знаннях. Спочатку система працює, використовуючи контентну фільтрацію – вона

підбирає новини, схожі з тими, які читає користувач, але після видачі користувач може зреагувати на підібрані рекомендації – поставити позначки «Мені подобається» / «Поділитися з друзями», або блокувати новину. Додаток запам'ятовує дії користувача і враховує їх надалі.

Таблиця 2.1 - Порівняльний аналіз методів підбору рекомендацій

	Контекстна фільтрація	**Колаборативна фільтрація**	Фільтрація на знаннях	Гібридна фільтрація
Проблема холодного старту	Відсутня	Присутня	Відсутня	Відсутня
Складність реалізації	Низька	Низька	Середня	Висока
Специфіка роботи	Інтернет-магазини, портали кіно, музики та ін.	Інтернет-магазини, портали кіно, музики та ін.	Інтернет-магазини	Будь-яка область
Точність рекомендацій	Середня	Середня	Висока	Висока
Потенційна швидкість роботи	Висока	Середня	Середня	Висока
Залежність від інших користувачів	Відсутня	Присутня	Відсутня	Залежить від реалізації

Колаборативна фільтрація використовується в рекомендаційних системах. Рекомендаційні системи традиційно використовуються для прогнозування вибору цікавих користувачеві об'єктів на основі наявної інформації, як про поточного користувача, так і про інших клієнтів. В якості об'єктів, що представляють інтерес для користувача, розглядаються: новини, веб-сайти, фільми, товари і т. п. Залежно від типу вхідних даних рекомендаційні системи застосовують два види зворотного зв'язку: явний і неявний. Явний заснований на оцінках користувачів, що відображають їх

інтерес до книги, фільму або іншого об'єкта. Такий зворотний зв'язок використовує, зокрема, компанія Netflix при формуванні рейтингу для фільмів і телевізійних програм. Користувачі мають можливість виставляти оцінки в даному сервісі. Однак в більшості випадків отримання явних оцінок користувачів пов'язано з труднощами, що і призводить до необхідності використання неявного зворотного зв'язку від користувача. Неявний зв’язок відображає переваги користувача на основі спостережень за його поведінкою. В якості вхідних даних при неявному зв'язку зазвичай використовують: історію покупок; переглянутих сайтів; шаблони пошуку; шаблони руху миші, і т. п. Ключовим підходом, який використовується при побудові рекомендаційних систем з неявним зворотним зв'язком, є колаборативна фільтрація. Даний підхід ґрунтується на принципах відбору за схожістю користувачів або за схожістю об'єктів, з якими взаємодіють користувачі. При колаборативної фільтрації з неявним зворотним зв'язком виявляють приховані фактори, які впливають на вибір користувача. При реалізації колаборативної фільтрації виникають проблеми, пов'язані з розрідженістю даних, а також наявністю релевантних даних. Розрідженість даних пов'язана з тим, що більшість комерційних рекомендаційних систем заснована на великій кількості даних про товари і незначній кількості оцінок користувачів.

В результаті матриця «об'єкт-користувач» виходить дуже великою з малою кількістю даних про покупки/перегляди. Це не дозволяє підвищити точність рекомендацій. Зазначена проблема особливо актуальна для нових, щойно створених систем. Проблема релевантності оцінок часто виникає в разі холодного старту, оскільки нові об'єкти або користувачі ускладнюють створення релевантних рекомендацій.

Для вирішення зазначених проблем доцільно використовувати не тільки розріджені «позитивні» дані-інформацію про кількість покупок, переглядів, але і даних про «негативні» переваги. Однак існуючі версії колаборативної фільтрації орієнтовані на позитивні дані. Це не дає можливість отримати генералізовану модель і може призводити до виникнення проблеми перенавчання.

Рекомендаційні системи класифікують за способом відбору необхідного користувачеві матеріалу. В основному застосовується два базових підходи: колаборативна фільтрація і контентна фільтрація. Також існує гібридна фільтрація, яка поєднує в собі як колаборативну, так і контентну фільтрацію.

В рекомендаційних системах, що використовують контентну фільтрацію (фільтрація по вмісту), користувачі не залежать від інших користувачів системи. Для формування рекомендацій системі необхідний профіль користувача з інформацією про його інтересах. У профілі в певній формі зберігається інформація про об'єкти, що цікавлять користувача. Система також містить інформацію про всі предмети, які вона може рекомендувати. Така система використовує опис об'єктів в профілі користувача, знаходить схожі об'єкти у своїй базі даних, а потім рекомендує їх йому. Застосування фільтрації такого роду дуже доречно, коли користувач має чітко визначені, конкретні інтереси і шукає схожі рекомендації. Перевага контентної фільтрації полягає в тому, що для початку надання рекомендацій не потрібна велика кількість зареєстрованих користувачів, тобто рекомендації не залежать від інших користувачів системи. Основним обмеженням методу є неможливість системи з таким видом фільтрації рекомендувати нові об'єкти, які не відповідають інтересам користувача.

Для того, щоб система мала високоякісні рекомендації, необхідно постійно оновлювати інформацію про інтереси користувача. Але існує й інша проблема – користувач завжди хоче мінімізувати час, що витрачається на взаємодію з системою, і неохоче ділиться інформацією для зворотного зв'язку з системою. Це типова проблема для всіх рекомендаційних систем, і оскільки рекомендації засновані виключно на інформації профілю користувача, тобто чим менше інформації користувач надає, тим менше відповідний набір рекомендацій користувач отримає. В такому випадку для вирішення цього завдання використовуються автоматичні системи збору інформації для підтримки профілю, але це ускладнює реалізацію системи. Цей метод фільтрації використовується в електронній комерції. У поєднанні з

колаборативної фільтрації, використовується в таких областях, як електронна освіта, програмування і т. д.

Рекомендаційні системи, що використовують колаборативну фільтрацію, шукають користувачів із загальними інтересами і рекомендують об'єкти, які були високо оцінені іншими користувачами. Інтереси користувача зберігаються у вигляді графіка переваг, який показує, наскільки користувачеві подобається той чи інший предмет. Такої системи достатньо для оцінки користувачів за обраною тематикою. Перевага цієї системи полягає в тому, що система повинна знати інтереси багатьох користувачів. Ця система не може створювати рекомендації, засновані на інтересах однієї людини.

Головним недоліком колаборативної фільтрації є «холодний старт». Система не здатна ефективно генерувати рекомендації, якщо більша частина користувачів не вкажуть свої інтереси в профілі.

Гібридні системи зазвичай поєднують колаборативну фільтрацію і контентну фільтрацію. Це дозволяє вирішити ряд проблем, які виникають при використанні цих методів окремо. В гібридній системі інформація про інтереси користувачів представлена в профілі в двох видах – як перелік властивостей певного об'єкта і його оцінка користувачем. Ця характеристика є як перевагою, так і недоліком системи. Перевагою є повнота інформації, що дозволяє використовувати ефективні алгоритми фільтрації і формувати необхідні рекомендації. Недоліком є те, що користувач повинен ввести більше інформації в свій профіль, яка завжди дуже неохоче, або не робиться взагалі.

Якість рекомендаційних систем можна визначити, оцінивши якість роботи алгоритмів, якість рекомендацій. Для оцінки якості рекомендацій з точки зору точності і повноти існує кілька стандартних підходів. Одним з таких підходів є метод ковзного контролю. Використання даного методу полягає в розбитті вихідного набору множини U на два підмножини: тестове Utest і навчальне Utraning. Повнота і точність пошуку оцінюється на тестовій вибірці. Кожен елемент тестової вибірки Utest розбивається на дві частини: на оцінені ознаки Ivisible і неоцінені Ihidden.

Алгоритм рекомендацій за оцінками користувачів використовує схожість об'єктів з тестової та навчальної вибірки для формування рекомендацій. Кожен користувач з Utest отримує рекомендації, як множину фіксованого розміру $r_n(u) = \{i1, i2, ..., in\}$. Далі точність і повнота розраховуються наступним чином:

$$recall = \frac{|r_n(u) \cap u^I \cap I_{hidden}|}{|u^I \cap I_{hidden}|} \quad (2.1)$$

$$precision = \frac{|r_n(u) \cap u^I \cap I_{hidden}|}{|r_n(u) \cap I_{hidden}|} \quad (2.2)$$

де u^I - сукупність всіх оцінених користувачем u об'єктів з I. Значення цих заходів обчислюються для кожного користувача і усереднюються. На додаток до стандартних способів обчислення точності і повноти пропонуються нетрадиційні варіанти розкриття невизначеностей, що виникають при їх оцінці.

Існує множина метрик для оцінки якості алгоритмів. В основному це метрики для оцінки точності очікуваного і реального значення. Розглянемо деякі з них: MAE, RMSE. MAE (Mean Absolute Error, пер. середня абсолютна помилка) – помилка оцінюється як різниця між розрахунковою і фактичною оцінкою по модулю:

$$MAE = \frac{\sum_{i \in n} |P_i - R_i|}{n} \quad (2.3)$$

RMS (Root Mean Squared Error, пер. середня квадратична помилка) – помилка обчислюється як корінь із суми квадратів різниць між можливим і дійсним значенням:

$$RMSE = \sqrt{\frac{\sum_{i \in n} (P_i - R_i)^2}{n}} \quad (2.4)$$

Інші характеристики рекомендаційних систем також можуть бути оцінені, але на основі всього переліку рекомендацій. Для цього необхідно мати інформацію про те, які елементи в списку рекомендацій відповідають запиту, а які ні. Виходячи з цього, характеристику точності можна представити у вигляді формули:

$$Accuracy = \frac{||T||}{L} \quad (2.5)$$

де L – довжина всього списку рекомендацій, T – множина відповідних рекомендацій, F – множина не відповідних елементів в списку F.

Характеристику повноти, в свою чергу, можна представити у вигляді наступної формули:

$$Re\ call = \frac{||T||}{||T||+||G||} \tag{2.6}$$

де G – множина елементів, які повинні були бути порекомендовані, але не опинилися в списку.

2.2 Побудова рекомендаційних систем за допомогою колаборативної фільтрації

Основна ідея алгоритмів колаборативної фільтрації полягає в пропозиції нових елементів для конкретного користувача на основі попередніх перевагах користувача або думки інших однодумців користувача. На сьогоднішній день дослідники розробили цілий ряд алгоритмів КФ, які можна розділити на дві основні категорії:

1. Методи, засновані на аналізі наявних оцінок, – анамнестичні методи (Memory-based). Ці алгоритми ґрунтуються на статистичних методах, щоб знайти групу користувачів близьких до цільового користувача. Цей підхід ще називають метод найближчих сусідів: використання попередніх оцінок, зроблених клієнтом, і аналіз оцінок інших користувачів, які мають подібні переваги. Тоді рекомендації (прогноз) для цільового користувача формуються на підставі обчислення якоїсь міри схожості за всіма накопиченими даними.

2. Методи, засновані на аналізі моделі даних, – модельні методи (Model-based). У цьому випадку спочатку за сукупністю оцінок формується описова модель переваг користувачів, товарів і взаємозв'язку між ними, а потім формуються рекомендації на підставі отриманої моделі. Процес формування рекомендацій розбитий на два етапи: ресурсномістке навчання моделі у відкладеному режимі і досить просте обчислення рекомендацій на основі існуючої моделі в реальному часі. Ці алгоритми можуть бути засновані на імовірнісному підході, кластерному аналізі, аналізі прихованих чинників.

3. Методи, засновані на об'єднанні попередніх алгоритмів, – гібридні методи. Ці підходи в свою чергу можуть бути розбиті далі на групи методів.

Так, методи на основі сусідства (близькості) поділяються на аналіз: – подібності користувачів (User-based); – подібності елементів (Item-based).

Метою обох напрямків є виділення схожих об'єктів в групи на основі матриці оцінок. У першому випадку визначається схожість користувачів: знайти інших користувачів, чиї минулі оцінки поведінки схожі на ті, що й у поточного користувача, і використовувати їх оцінки інших елементів для прогнозування переваги поточного користувача. Другий підхід, на основі подібності елементів, вперше запропонований В, і ця версія використовується в Amazon.com в даний час. У цьому випадку замість того щоб використовувати подобу між поведінкою призначених для користувача оцінок для прогнозування переваги, використовується схожість між оцінками моделей елементів. Якщо два елементи, як правило, мають однакові оцінки користувачів, то вони схожі, і користувачі повинні мати аналогічні переваги для подібних елементів.

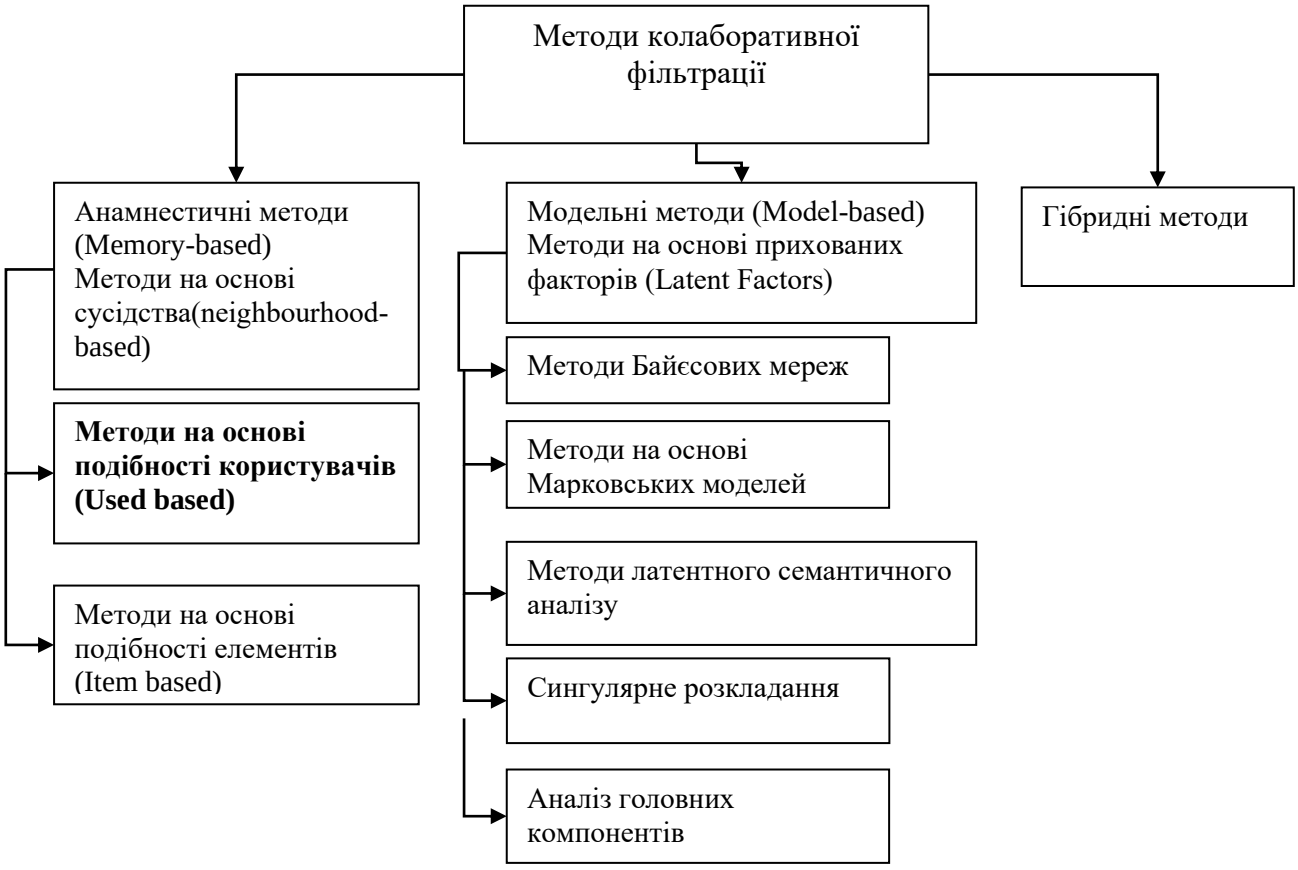

Рис. 2.3 – Класифікація методів колаборативної фільтрації

Для визначення подібності між користувачами або елементами можна використовувати такі підходи: – відстань Евкліда, Хеммінга; – кореляція Пірсона; – рангова кореляція Спірмена; – коефіцієнт Жаккара. Колаборативна фільтрація на основі подібності користувачів (User-based) має високу точність. Однак, недоліком є ресурсомісткість (вимога до пам'яті) і складність (кількість обчислень, необхідну для отримання рекомендацій). До того ж обчислення ступеня близькості може проводитися тільки в реальному часі, так як дані про поточні транзакції стають доступними тільки в момент вироблення рекомендацій. Тому даний метод може застосовуватися тільки до відносно невеликих баз даних. В алгоритмі на основі подібності елементів (Item-based) ступінь близкості аналізованого елемента до всіх інших може бути обчисленим у відкладеному режимі за розкладом, так як вектори рейтингів всіх елементів доступні до моменту формування рекомендації. Таким чином цей алгоритм виявляється більш ефективним з точки зору часу формування рекомендацій завдяки можливості проведення відкладенї попередньої обробки даних.

Інформаційна область для систем КФ складається з користувачів, які висловили переваги для різних предметів. Перевага (оцінка) часто представляється у вигляді триплета (користувач, предмет, оцінка). Ці оцінки можуть приймати різні форми, в залежності від даної системи. Деякі системи використовують речову або цілочисельну оціночну шкалу, таку як 0-5 зірок, інші використовують бінарні або потрійні заходи. Множина всіх триплетів оцінок формує розріджену матрицю, звану матрицею оцінок. Пари (користувач, предмет), в яких користувачі не віддали перевагу предмету, є невідомими значеннями цієї матриці (Табл. 2.2).

Таблиця 2.2 – Приклад матриці оцінок

	Елемент 1	Елемент 1	Елемент 1
Користувач 1	3	?	2
Користувач 2	?	4	3
Користувач 3	5	4	?

При використанні системи КФ необхідно вирішити два завдання:

1) спрогнозувати оцінку або перевагу, яку користувач віддасть предмету. Метою прогнозу є заповнення в матриці оцінок відсутніх значень;

2) Видача рекомендації, тобто формування ранжованого списку N елементів для даного користувача. Визначимо математичні позначення для прив'язки різних елементів моделі рекомендаційних систем. Генеральна сукупність складається з набору користувачів U і набору елементів I.

I – множина елементів, які оцінили користувачі u.

U_i – множина користувачів, які оцінили елемент i.

$r_{u,i}$ – оцінка користувача u для елемента i.

r_u – вектор всіх оцінок користувача u.

r_i – вектор всіх оцінок елемента i.

r_u та r_i – середні значення оцінок користувача u і елемента i відповідно.

Рекомендаційний прогноз позначимо як $r_{u,i}$.

2.3 Удосконалення колаборативної фільтрації на основі пропущених даних з неявним зворотним зв’язком

В якості вихідних даних при побудові рекомендацій використовуються:

– список користувачів $U = \{u_i\}$;

– список об'єктів, що цікавлять користувачів $E = \{e_j\}$;

– матриця рейтингів/покупок $R \subseteq U \times E$;

Користувачі взаємодіють з об'єктами (товарами), формуючи матрицю рейтингів в рекомендаційній системі: $R = \{r_{ij}\}$, де R_{ij} – рейтинг j – предмета e_j у користувача u_i.

Очевидно, що кожен з користувачів ставить рейтинг (у разі явного зворотного зв'язку), купує (у разі неявного зв'язку) тільки незначну підмножину товарів із загального списку. Тому матриця рейтингів зазвичай сильно розріджена, більшість її елементів рівні "0". Іншими словами, тільки невелика частина з можливих взаємодій «користувач-товар» буде в наборі даних.

Результатами побудови рекомендацій є упорядкований список об'єктів E_U для кожного користувача або впорядкований список користувачів U_E , відповідний кожному об'єкту. Для обох списків має виконуватися умова: об'єкти і користувачі присутні в матриці R:

$$E_U = \left\{ e_i \middle| \exists r_{ij} \in R \right\} \tag{4.1}$$

$$U_E = \left\{ u_j \middle| \exists r_{ij} \in R \right\} \tag{4.2}$$

Формується рекомендаційною системою порядок користувачів або об'єктів володіє властивостями загальності (4.3), транзитивности (4.4), антисиметричности (4.5):

$$\begin{gathered} \forall (e_i \neq e_k) \in E_U \underset{u_j}{\exists}(>) : e_i \underset{u_j}{>} e_k \vee e_i \underset{u_j}{<} e_k, \\ \forall (u_j \neq u_l) \in U_E \underset{e_i}{\exists}(>) : u_j \underset{u_j}{>} u_l \vee u_j \underset{u_j}{<} u_l, \end{gathered} \tag{4.3}$$

$$\begin{gathered} \forall (e_i \neq e_k \neq e_n) \in E_U : \\ e_i \underset{u_j}{>} e_k \wedge e_k \underset{u_j}{>} e_n \Rightarrow e_i \underset{u_j}{>} e_n, \\ \forall (u_j \neq u_l \neq u_m) \in U_E : \\ u_j \underset{u_j}{>} u_l \wedge u_j \underset{u_j}{>} u_l \wedge u_j \underset{u_j}{>} u_m, \end{gathered} \tag{4.4}$$

$$\begin{gathered} \forall (e_i = e_k) \in E_U : e_i \underset{u_j}{>} e_k \wedge e_i \underset{u_j}{<} e_k, \\ \forall (u_j = u_l) \in U_E : u_j \underset{u_j}{>} u_l \wedge u_j \underset{u_j}{<} u_l, \end{gathered} \tag{4.5}$$

Важлива відмінність систем з неявнним зворотним зв'язком полягає в тому, що замість позитивних і негативних рейтингів вони використовують тільки позитивні дані (наприклад, кількість покупок). Інші дані відображають об'єкти, що не цікавлять користувача, так і пропущену інформацію про покупки, перегляди і т. п. В процесі обробки матриці R в рекомендаційних системах зазвичай виконується бінаризація даних. При цьому позитивні дані представляються одиницею, а інша інформація – нулем. У подальшому в процесі представлені одиницею дані розглядаються як належні до класу, а всі інші, включаючи пропущені – які лежать за межами класу. Побудована в

результаті навчання модель повинна передбачати одиниці в матриці вихідних даних R . Отже, пропущена інформація виключається з подальшого розгляду. Однак отримана модель має впорядковувати всі об'єкти для кожного користувача згідно описаних вище властивостей, що вимагає сортування негативних результатів в матриці вихідних даних. Для виконання сортування доцільно виконати попарне порівняння об'єктів j_e для кожного користувача i_u , які відображені в матриці R .

Пропонований підхід до вирішення цього завдання включає в себе такі кроки:

1. Визначення відносного порядку між об'єктами для кожного користувача.

На даному кроці виконується попарне порівняння результатів (покупок, переглядів) для кожного користувача. Для користувача формується квадратна матриця, елементи якої вказують, який з об'єктів є більш цікавим для нього. Знак + в цій матриці означає, що елемент в рядку більш цікавий, ніж елемент в стовпці. Приклад реалізації даного кроку для користувача 1 u наведено на рис. 2.4.

	e_1	e_2	e_3	e_4
u_1	0	9	6	0

⇓

	e_1	e_2	e_3	e_4
e_1	=	9	6	0
e_2	+	=	+	+
e_3	+	-	=	+
e_4	?	-	-	=

Рис. 2.4 – Визначення відносного порядку між парами об'єктів для кожного користувача

Так, знак + в комірці (2.1) означає, що для користувача u_1 елемент e_2 краще елемента e_1. Дійсно, користувач вибрав елемент e_2 9 раз, а елемент e_1 – 0 раз. Знак питання в комірці квадратної матриці означає, що переваги встановити не вдалося. Наприклад, користувач не вибирав обидва елементи e_1 та e_4 тому в комірці (4.1) стоїть знак питання.

На даному кроці на основі відносин між об'єктами визначаються ваги, що відображають можливий інтерес до цих об'єктів. Основна ідея кроку полягає в наступному. Нам необхідно побудувати досить узагальнену модель, яка відображала б загальні тенденції та уподобання користувачів та дозволяла б уникнути окремих флуктуацій, що відображають специфіку конкретного набору даних. Традиційно така задача вирішується шляхом регуляризації. Коефіцієнт регуляризації підбирається таким чином, щоб згладити флуктуації окремих змінних щодо необхідної закономірності. У нашому випадку причинами пропуску об'єктів користувачем (наприклад, e_1 та e_4) можуть бути:

– відсутність інтересу;

– відсутність доступу;

– помилка в записах.

При традиційному підході всі ці причини об'єднуються і розглядаються як відсутність інтересу до об'єкту. Для виключення залежності від конкретних даних пропонується додати невеликі значення пропущеним елементам відповідно до встановленої на кроці 1 впорядкованістю. Для спрощеного прикладу отримуємо таку впорядкованість елементів: e_2 (9) , e_3 (6) , e_1 та e_4 (?).

Лінійна екстраполяція отриманої послідовності елементів i_e очевидно призводить до таких їх значень: 9; 6; 3; 3. Для виключення флуктуацій використовуємо коефіцієнт регуляризації, наприклад $\lambda = 0{,}1$. Тоді впорядкована послідовність значень елементів приймає такий вид: 9; 6; 0,3; 0,3. Останні 2 значення в послідовності елементів відображають потенційний інтерес користувача до елементів e_1 и e_4 як результат відсутності доступу або помилки в вихідних даних.

Очевидно, що представлена в прикладі лінійна інтерполяція є окремим випадком загальної закономірності, що зв'язує значення елементів e_1 – e_4.

Значення коефіцієнта λ підбирається експериментально. Для цього в рамках колаборативної фільтрації необхідно провести декілька циклів з різними значеннями λ.

Для описаних вище методів є необхідність у зберіганні всієї матриці даних, тобто переваг користувачів про елементи. У зв'язку з цим виникають труднощі при прогнозі переваг для нових користувачів або при появі нових елементів, тому що для них ще немає оцінок. Також обмежується можливість методів при обробці великих обсягів даних. У багатьох випадках зберігання всієї матриці переваг надмірно: як правило, користувачі і елементи діляться на групи з аналогічними профілями переваг. Наприклад, науково-фантастичні фільми будуть подобатися в аналогічній мірі тим же наборам користувачів. Тому виникає завдання в зниженні розмірності матриці оцінок. Такі завдання вирішують методи другої групи. У цьому випадку можливий варіант об'єднання користувачів (елементів) в кластери (профілі) за допомогою деякого індексу подібності. Елементи та оцінки, дані користувачами з одного кластера, використовуються для обчислення рекомендацій. Кластерні моделі краще масштабуються, тому що звіряють профіль користувача з відносно невеликою кількістю сегментів, а не з цілою користувацькою базою.

Крок 1: для кожного елемента j обчислюється міра близькості до елемента i. Для цього можна використовувати один із зазначених вище підходів, наприклад, коефіцієнт Пірсона:

$$s_{i,j} = \frac{\sum_{u \in U}(r_{u,i} - \bar{r}_i)(r_{u,j} - \bar{r}_j))}{\sqrt{\sum_{u \in U}(r_{u,i} - \bar{r}_i)^2}\sqrt{\sum_{u \in U}(r_{u,j} - \bar{r}_j)^2}} \quad (3.1)$$

де $U \in U_j \cup U_i$ – множина користувачів, які оцінили елементи i та j.

Крок 2: вибираємо множина елементів S, найбільш близьких до об'єкта i. Відомо, що достатні результати виходять при k=30 елементів множини S. Але ці дані залежні від розглянутої задачі і розрідженості матриці.

Крок 3: прогноз рейтингу (оцінки) об'єкта на основі рейтингів близьких до нього об'єктів:

$$\hat{r}_{u,i} = \frac{\sum_{j \in S} s_{i,j} \cdot r_{u,j}}{\sum_{j \in S} | s_{i,j} |} \quad (3.2)$$

Даний алгоритм відображає теоретичну базу методу, але на практиці ряд факторів вимагає переосмислення розрахунків. Як правило, переважна більшість оцінок невідома, і розрідженість матриці оцінок досить висока. З іншого боку, дані, які вже є в матриці досить суб'єктивні. Деякі користувачі-оптимісти, і їх оцінки завжди високі (середнє 4 з 5), у інших користувачів оцінки завжди занижені (середнє 2,5 з 5). Крім цього, завжди є елементи, які подобаються всім. В цілях боротьби з підгонкою розріджених даних з оцінками, проводиться регуляризація моделей таким чином, щоб скоротити ймовірність появи випадкових зв'язків між оцінками, які не відображають дійсність. Регуляризація контролюється константами, які позначаються як λ_1, λ_2, ... Точні значення цих констант визначаються перехресною перевіркою. У міру їх зростання, регуляризація стає все важче. Для того щоб оптимізувати продуктивність видачі рекомендацій, важливо нормалізувати оцінки до обчислення матриці подібності. Це може бути досягнуто шляхом обчислення базового прогнозу, в якому інкапсулюють відхилення користувача та елемента. Пари користувач-елемент (u, i), для яких оцінки $r_{u,\,i}$ відомі, складають множина K. Базовий прогноз для невідомої оцінки $r_{u,\,i}$ позначається $b_{u,\,i}$ та визначається формулою:

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

Де μ – загальна середня оцінка; b_u и b_i - параметри, які показують спостережуване відхилення користувача u і елемента i відповідно від середнього значення. Так як всі параметри взаємопов'язані, то розраховувати їх необхідно разом вирішивши завдання найменших квадратів.

$$min \sum_{(u,i) \in K} (r_{u,i} - \mu - b_u - b_i)^2 + \lambda_1 (\sum_u b_u^2 + \sum_i b_i^2) \quad (3.4)$$

Тут перша частина прагне знайти b_u и b_i , які відповідають даним оцінок. Частина регуляризації дозволяє уникнути підгонки даних, штрафом за величину параметрів. Для методу на основі подібності елементів цей підхід

відіб'ється так. Розрахунок заходи близкості заснований тільки на оцінках користувачів, які оцінили обидва елементи:

$$s_{i,j} = \frac{n}{n+\lambda_2} \cdot p_{i,j} \tag{3.5}$$

Де n – кількість користувачів, які оцінили обидва елементи i та j; λ_2 - константа регуляризації; $p_{i,j}$ – коефіцієнт кореляції Пірсона за формулою (3.1). Прогнозоване значення $r_{u,i}$ отримаємо як середньозважену оцінку сусідніх елементів, у той час як коригування для користувачів і елементів проводяться через базові прогнози:

$$\hat{r}_{u,i} = b_{u,i} + \frac{\sum_{j \in S} s_{i,j} \cdot (r_{u,j} - b_{u,j})}{\sum_{j \in S} | s_{i,j} |} \tag{3.6}$$

$$min \sum_{(u,i) \in K} \left(r_{u,i} - \mu - b_u - b_i - \frac{\sum_{j \in S} s_{i,j} \cdot (r_{u,j} - b_{u,j})}{\sum_{j \in S} | s_{i,j} |} \right)^2 + \lambda_3 \left(\sum_u b_u^2 + \sum_i b_i^2 + \sum_{j \in K} s_{i,j}^2 \right) \tag{3.7}$$

$$b_u \leftarrow b_u + \gamma_1 \cdot (e_{i,j} - \lambda_1 \cdot b_u) \tag{3.8}$$

$$b_i \leftarrow b_i + \gamma_1 \cdot (e_{i,j} - \lambda_1 \cdot b_i) \tag{3.9}$$

$$e_{i,j} = r_{i,j} - \hat{r}_{i,j} \tag{3.10}$$

Де γ_1 – константа регуляризації.

На основі представлених математичних викладок були проведені дослідження з використання описаного методу. для оцінки ефективності використовувалася середньоквадратичне відхилення.

$$RMSE = \sqrt{\frac{1}{n} \sum_{u,i} \left(\hat{r}_{u,i} - r_{u,i} \right)^2} \tag{3.11}$$

Де $r_{u,i}$ – відома оцінка користувача u для елемента i, $r_{u,i}$ – прогнозована оцінка.

3 СТРУКТУРНО-ФУНКЦІОНАЛЬНЕ МОДЕЛЮВАННЯ ПРОЦЕСУ ОЦІНЮВАННЯ РЕКОМЕНДАЦІЙНИХ СИСТЕМ

3.1. Система оцінювання рекомендаційних систем

Методологія SADT є сукупністю методів, правил і процедур, призначених для побудови функціональної моделі об'єкту, яка відображає його функціональну структуру, тобто вироблювані їм дії та зв'язки між ними. Основні елементи цієї методології ґрунтуються на таких положеннях:

1. Графічне представлення блочного моделювання. Графіка блоків і дуг SADT – діаграми відображає функцію (процес) у вигляді блоку, а інтерфейси входу/виходу подаються дугами, що входять до блоку і відповідно виходять із нього. Ці дуги моделюють взаємодію блоків та виражають "обмеження", які у свою чергу визначають, коли і яким чином функції виконуються й керуються;

2. Строгість і точність, і виконання правил SADT, вимагає достатньої строгості і точності, не накладаючи у той же час надмірних обмежень на дії аналітика.

Для того, щоб мати таку інформацію, створюється модель бізнес-процесу.

Бізнес-процес являє собою систему послідовних, цілеспрямованих і регламентованих видів діяльності, в якій за допомогою керуючого впливу і за допомогою ресурсів входи процесу перетворюються в виходи, результати процесу, що представляють цінність для споживачів.

Моделювання бізнес-процесу – процес відображення суб’єктивного бачення потоку робіт у вигляді формальної моделі, що складається з взаємопов'язаних операцій. Метою моделювання є систематизація знань про інформаційну систему в наочній графічній формі з тим, щоб в подальшому дані процеси можна було аналізувати і вдосконалювати.

Використовується кілька різних методів, основою яких є як структурний, так і об’єктно-орієнтовний підходи до моделювання. Однак розподіл самих методів на структурні і об'єктні є досить умовним, оскільки найбільш розвинені

методи використовують елементи обох підходів. До числа найпоширеніших методів відносяться:

– метод функціонального моделювання SADT (IDEF0);

– метод моделювання процесів IDEF3;

– моделювання потоків даних DFD;

– метод ARIS;

– метод Ericsson-Penker.

На сьогодні найвідомішими мовами (нотаціями) графічного моделювання є UML, ARIS, IDEF (IDEF0, IDEF3 у програмній інтерпретації BPwin), BPMN. Але найбільш широко використовується методологія опису IDEF.

За допомогою методології сімейства IDEF можна ефективно відображати і аналізувати моделі діяльності широкого спектру складних систем в різних розрізах.

До сімейства IDEF відносяться такі стандарти:

IDEF0 - методологія функціонального моделювання, яка за допомогою наочної графічної мови представляється у вигляді набору взаємозалежних функцій;

IDEF1 - методологія моделювання інформаційних потоків усередині системи, що дозволяє відображати і аналізувати їх структуру та взаємозв’язки;

IDEF1X - методологія побудови реляційних структур;

IDEF2 - методологія динамічного моделювання розвитку систем;

IDEF3 - методологія документування процесів, що відбуваються в системі, яка використовується, наприклад, при дослідженні технологічних процесів на підприємствах (за допомогою IDEF3 описуються сценарій та послідовність операцій для кожного процесу);

IDEF4 - методологія побудови об’єктно-орієнтованих систем;

IDEF5 - методологія дослідження складних систем.

Метою побудови функціональних моделей, звичайно, є виявлення найбільш слабких і вразливих місць ІС. Аналіз недоліків і «вузьких місць» починають з побудови моделі AS-IS (як є), тобто моделі існуючої ІС. Одержана модель AS-IS служить для виявлення дубльованих робіт, робіт не забезпечених

ресурсами, неефективних робіт і інших недоліків в ІС. Виправлення недоліків, перенаправлення інформаційних і матеріальних потоків призводить до створення моделі близької до ідеальної організації процесів.

Етап формування первинних рекомендацій використовує метод колаборативної фільтрації з неявним зворотним зв'язком. На виході етапу формується перелік рекомендацій. На етапі формування рекомендацій до отриманого переліку рекомендацій додаються додаткові характеристики, відгуки, картинки з реляційній бази даних, що надає змогу видавати ці рекомендації в різному представленні.

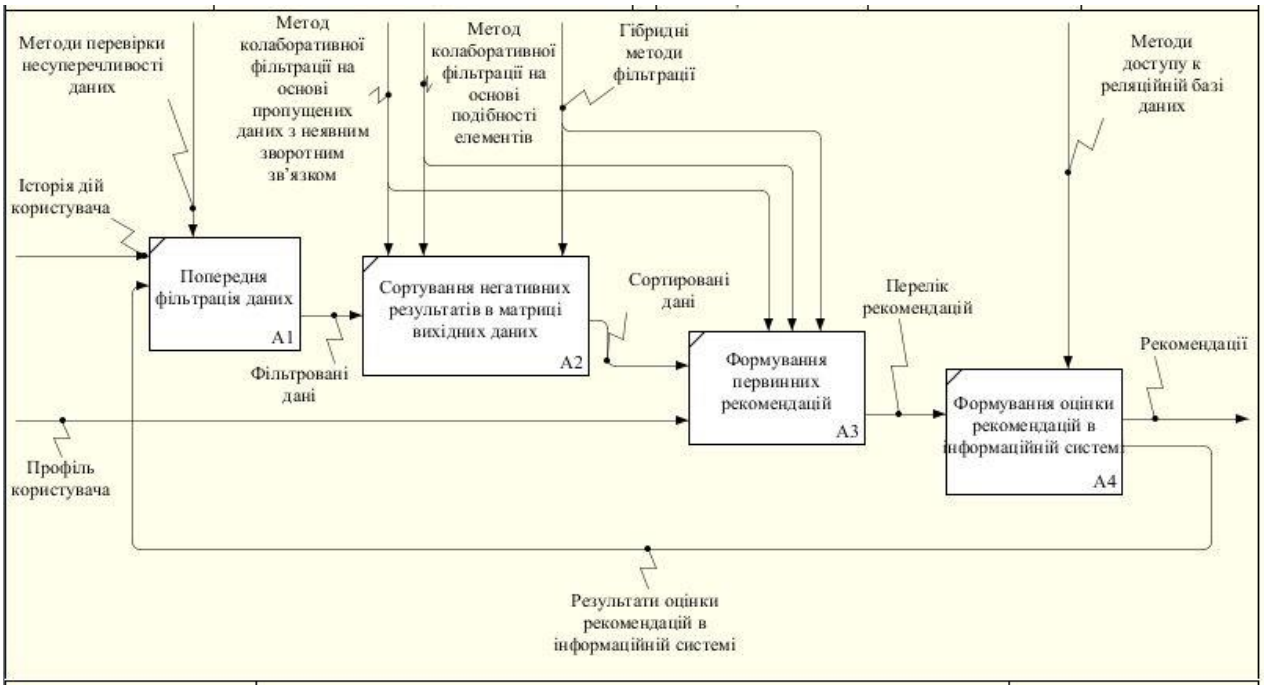

Рис 3.1 – Технологія використання удосконаленого методу

Частіше за все використовують CASE - технології, що базуються на методі структурного аналізу та проектування SADT. Особливостями цього стандарту є можливість наочно відобразити послідовність виконання будь-якого процесу, у тому числі, й процесу дослідження методів оцінки рекомендаційних систем.

4 УДОСКОНАЛЕННЯ ОЦІНЮВАННЯ РЕКОМЕНДАЦІЙНИХ СИСТЕМ НА ОСНОВІ КОЛАБОРАТИВНОЇ ФІЛЬТРАЦІЇ

4.1 Реалізація удосконаленого методу

Розроблюваний рекомендаційний механізм буде клієнт-серверним додатком. Серверна частина буде складатися з наступних модулів: модуль обробки вхідних даних, модуль побудови і тренування моделі і модуль генерації рекомендацій. Клієнтська частина програми буде запитувати рекомендації у серверній за номером користувача і, отримавши відповідь, виводити їх на екран.

На даний момент найбільшою популярністю в сфері розподіленого аналізу даних користується фреймворк Hadoop, але існують і альтернативи, що пропонують деякі важливі переваги порівняно з типовою Hadoop-платформою. Spark - це масштабована платформа аналізу даних, яка включає в себе примітиви для обчислень в оперативній пам'яті і, отже, має деякі переваги в частині продуктивності по відношенню до підходу Hadoop, заснованого на кластерній схемі зберігання даних. Spark реалізований на Scala і підтримує цю мову, яка забезпечує унікальне середовище для обробки даних.

Spark являє собою кластерну обчислювальну платформу з відкритим вихідним кодом, аналогічну Hadoop, але з деякими корисними особливостями, які роблять її чудовим інструментом для вирішення завдань машинного навчання. А саме, крім інтерактивних запитів Spark підтримує розподілені набори даних в оперативній пам'яті, оптимізуючи рішення ітеративных завдань і зменшуючи час доступу до даних.

Spark реалізований на мові Scala і використовує його в якості середовища розробки додатків. На відміну від Hadoop, Spark і Scala утворюють тісну інтеграцію, при якій Scala може легко маніпулювати 34 розподіленими наборами даних як локальними колективними об'єктами. Грунтуючись на вище перерахованих перевагах, виберемо для побудови рекомендаційної системи

фреймворк Apache Spark і мову Scala. Основним поняттям в Spark є RDD — Resilient Distributed Dataset (стійкий розподілений набір даних) — це відмовостійка незмінна розподілена колекція об'єктів, яку можна обробляти паралельно. RDD може містити об'єкти будь-якого типу.

Трансформації — це операції, що здійснюються над RDD. У підсумку вийде новий RDD, в якому міститься результат. Наведемо приклади найбільш популярних трансформацій:

.map(function) — виконує функцію-аргумент function щодо кожного елемента набору даних;

.filter(function) — поверне ті елементи набору даних, де функція-аргумент function повернула справжнє значення;

distinct(numTasks) - поверне набір даних, в якому містяться унікальні елементи вихідного набору.

Також відзначимо кілька операцій над множинами:

.intersection(otherDataset)

.union(otherDataset)

.cartesian(otherDataset) — поверне набір даних, у якому містяться комбінації значень пари (A, B), де A — елемент вихідного набору, B — елемент набору-аргументу.

Дії — це операції, які повертають значення, отримане в результаті обчислень в RDD. Їх виконують, коли треба матеріалізувати результат (наприклад, зберегти на диск дані або вивести їх в консоль). Найбільш часто використовувані дії:

.saveAsTextFile(path) — зберігає набір даних в текстовий файл;

.collect() — повертає елементи набору даних у вигляді масиву. В основному, використовується в тому випадку, коли даних в наборі вже мало (застосовані раніше Різні фільтрації і перетворення) і потрібна візуалізація;

.take (n) - поверне перші n елементів набору даних у вигляді масиву;

.count () - поверне кількість елементів в наборі даних.

Вхідні дані передаються в програму шляхом завдання аргументів для головного методу main серверної частини. Першим кроком до побудови моделі

є розбір наявних даних, їх парсинг і перетворення в форму, зручну для аналізу в Spark. В його алгоритмі ALS, реалізованому в бібліотеці ML, є одне обмеження — для роботи з ним потрібні позитивні 32-розрядні цілочисельні ID користувачів і виконавців. Це означає, що ID більше значення Integer.MAX_VALUE (це 2147483647) не можуть бути використані.

Отже, через те, що функція flatMap нижче може повернути або порожній List, або List з одного елемента, резонніше використовувати Some і None:

```
def buildingArtistByID(): DataFrame = {
 rawArtistsData.flatMap { line =>
 val (id, name) = line.span(_ != '\t')
 if (name.isEmpty) {
 None
 } else {
 try {
 Some((id.toInt, name.trim))
 } catch {
 case _: NumberFormatException => None
 }
 }
 }.toDF("id", "name")
}
```

Спочатку потрібно подивитися, чи мають сенс рекомендації артистів користувачеві, досліджуючи те, що система йому порекомендувала. Візьмемо для прикладу слухача 1000002. Знайдемо ID виконавців, яких він слухав і виведемо на друк їх імена:

```
println("User #" + exampleUserID + " listened to following artists:")
val existingArtistIDs: Array[Int] = allInfo.
 filter($" user " === exampleUserID).
 select(" artist ").as[Int].collect()
allArtistsByID.filter($" id " isin (existingArtistIDs: _*)).show()
```

Таблиця 4.1 – Переваги користувача

ID виконавця	Назва виконавця
1195	Elvis Costello
1004346	Joshua Redman
658	George Duke
1003472	Brian Hughes
3437	Enigma

Гіперпараметр – параметр "налаштування" роботи алгоритму. Різні комбінації гіперпараметрів підходять для різних завдань. При правильно підібраних значеннях можна домогтися найбільш високих результатів для конкретного алгоритму. Перерахуємо такі аргументи для ALS():

1) iterations: кількість ітерацій, за яке виконується факторизація матриці. Більша кількість займе більший час, але буде досягнута краща факторизація;

2) lambda: стандартний параметр перенавчання. (Перенавчання — явище, коли побудована модель добре пояснює приклади з навчальної вибірки, але відносно погано працює на прикладах, які не брали участі в навчанні (на прикладах з тестової вибірки)). Високі значення цього параметра перешкоджають перенавчанню, але якщо взяти занадто великі, то це зашкодить точності факторизації;

3) alpha: контролює в процесі факторизації відносну вагу спостережуваних взаємодії «користувач-виконавець» по відношенню до прихованих. Занадто високі значення цього параметра (наприклад, 100) налаштовують модель таким чином, що пропущені дані автоматично прирівнюються до негативних. Це не підходить для нашого випадку, тому що ставиться мета порекомендувати користувачеві якраз таки нових для нього виконавців. Значеннями за замовчуванням є:

1. lambda = 1.0

2. alpha = 1.0

3. iterations = 10 Lambda і alpha є гиперпараметрами моделі, iterations більше відноситься до ресурсних обмежень для факторизації.

Значення, перераховані вище, необов'язково повинні бути оптимальними. Вибір задовільних гіперпараметрів є проблемою в машинному навчанні. Найпростішим способом вибрати найкращі значення є перебір їх комбінацій. Спробуємо 81 таких наборів: lambda = 5, 0.0001 або 1, alpha = 40, 15 або 1, iterations = 5, 10 або 20. Ці значення вибрані довільно, але таким чином, щоб задіяти весь діапазон можливих величин. Результати виведені на друк у спадному за оцінкою AUC порядку:val calculations: Seq[(Double, (Int, Double, Double, Int))] =

```
 Seq(1, 10, 40).flatMap { lambda => Seq(1.0, 15.0, 40.0)
 .flatMap { alpha => Seq(5, 10, 20)
 .map { iterations =>
 val model: ALSModel = modelBuilder(trainingData, lambda, alpha, iterations)
 val auc: Double = calculatingOfAUC(testingData, broadcastedIdsOfAllArtists,
model.transform)
 (auc, (lambda, alpha, iterations))
 }
 }
 }
}
println("Evaluated AUCs with different parameters are: ")
calculations
 .sorted
 .reverse
 .foreach(println)
```

Нижче наведемо вибірку з отриманих результатів.

Таблиця 4.2 – AUC при різних гіпермараметрах

Величина AUC	Значення гіперпараметрів		
	Lambda	Alpha	Iterations
0.889	5.0	40.0	20
0.883	1.0	1.0	20
0.840	1.0	1.0	10
0.800	1.0	1.0	20
0.793	5.0	1.0	20
0.733	1.0E-4	40.0	10
0.694	1.0E-4	15.0	10
0.624	1.0E-4	40.0	10

Як видно з результатів, параметр alpha дає кращі результати при величині 40. Це можна вільно інтерпретувати як факт того, що краще модель фокусується на тому, що користувач слухав, ніж на те, що не слухав. При високій лямда значення виходять теж трохи краще. Це означає, Lamb лямбда, щоб перешкодити спробі заповнити розріджені вхідні дані від кожного користувача дуже точно. При більш високих значеннях ітерацій досягається краща факторизація матриці, обчислення хоч і займають більше часу. Отже, найкращими гіперпараметрами є:

1. lambda = 5.0
2. alpha = 40,0
3. iterations = 20

4.2 Експериментальна перевірка удосконаленого методу

Метод на основі подібності елементів має параметри, такі як розмір сусідства k, коефіцієнт базового відхилення оцінок γ_1, коефіцієнт регуляризації базового відхилення оцінок λ_1. Останні два параметри досліджені і прийняті: коефіцієнт швидкості процесу базового відхилення оцінок $\gamma 1$=0,001. коефіцієнт

регуляризації базового відхилення оцінок λ1 =0,005. За основу взято ці дані, визначимо оптимальну кількість факторів, при точності ε=0,00001.

Таблиця 4.2 – Результати похибок експерименту

№	Розмір сусідства	RMSE при навчанні	RMSE при тестуванні
1	25	0.87515	0.94378
2	50	0.87225	0.93456
3	75	0.86981	0.93103
4	100	0.86766	0.92748
5	125	0.86574	0.92657
6	150	0.86269	0.92486
7	175	0.86027	0.92361
8	200	0.85833	0.92276

Значення RMSE, отримані при тестуванні методу адекватні для заданого завдання, тому що всі дослідники прагнуть, відповідно до запитів Netfix, зменшити RMSE з 0.9514 до 0.8563. На рисунку 4.2 показано графічне відображення даних таблиці 4.1. Згідно з отриманими результатами можна зробити висновок про те, що зі збільшенням розміру сусідства, похибка прогнозів зменшується. Для вихідних даних оптимальним розміром сусідства є k =200.

Важливим є використання методів пошуку прихованих факторів зі скороченням розмірності вимірювань на основі сингулярного розкладання матриць.

Удосконалення оцінювання рекомендаційної системи включає в себе наступні етапи:

– підготовка відфільтрованої матриці вихідних даних R*;

– формування навчальної та тестової вибірок з матриці R*;

– виявлення неявних зв'язків між клієнтами і товарами на основі матричної факторизації; – оцінка отриманих рекомендацій.

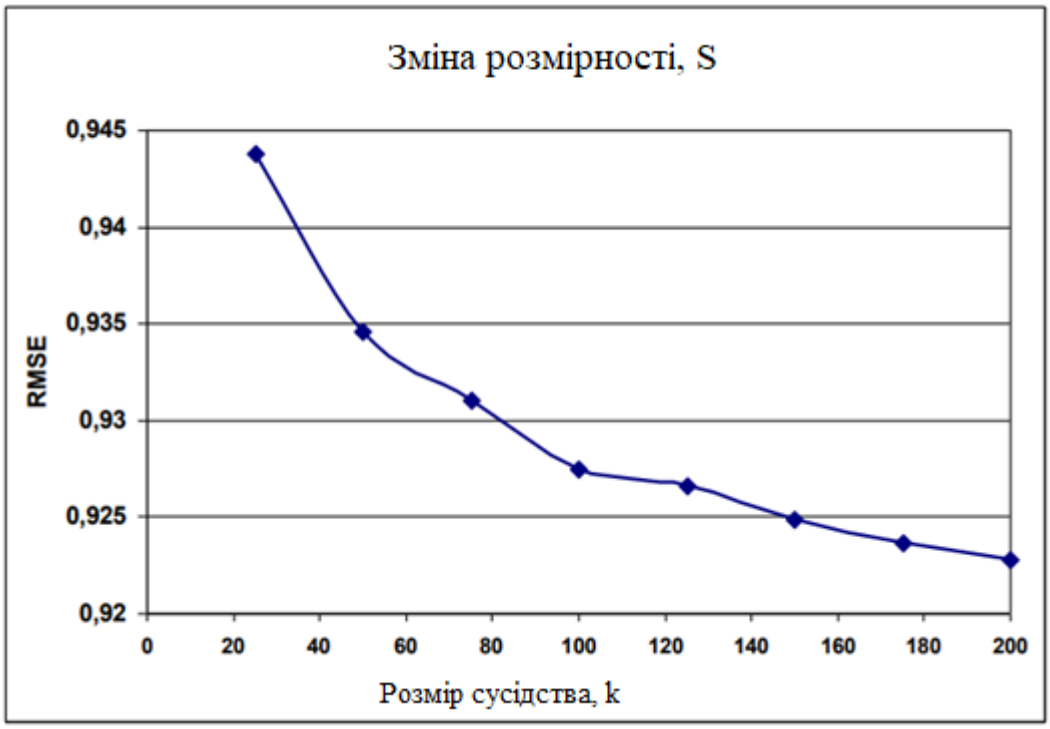

Рисунок 4.3 – Результати експерименту

Таблиця 4.3 – Зміна параметру AUC

Початкова величина AUC	0,862
Величина AUC за результатами корегування данних після першої оцінки ефективності	0,877
Величина AUC за результатами корегування данних після другої оцінки ефективності	0, 881

На етапі підготовки матриці R* виконуються такі кроки: фільтрація даних; перевірка придатності вихідних даних; побудова матриці R «користувачі-об'єкти» з вихідної бази даних; побудова матриці пріоритетів.

На кроці фільтрації даних з вихідних записів видаляються помилкові рядки, тобто такі, в яких відсутня інформація про користувача або про об'єкт

його інтересу (товар). На кроці перевірки придатності вихідних даних проводиться перевірка розрідженості матриці. На практиці в якості порога розрідженості зазвичай розглядають значення 99,5%. Якщо розрідженість не перевищує цей поріг, то вихідні дані вважаються придатними для формування рекомендацій. На кроці побудови матриці «користувач-об'єкт» реалізуються інженерні рішення, що дозволяють перетворити наявні записи про покупки (перегляди) в стандартний формат вихідних даних. На кроці побудови матриці пріоритети c_{ij} для «позитивних» даних формуються шляхом нормування кількості покупок для елементів r_{ij}.

Послідовність формування пріоритетів для «негативних» даних була наведена вище. На етапі формування навчальної та тестової вибірок виконується обробка вихідної матриці R*. Обидві вибірки створюються з цієї матриці. На даному етапі виконуються такі кроки: формується навчальна вибірка шляхом маскування частини даних; тестова вибірка формується шляхом бінаризації вихідної матриці. В якості навчальної вибірки використовується підмножина елементів матриці R* . Основне завдання даного кроку полягає в тому, щоб відобразити характер взаємодії максимальної кількості користувачів з більшістю об'єктів. Тому близько 30% елементів даної матриці видаляється випадковим чином. Тестова вибірка формується шляхом заміни кількості покупок/переглядів. Завдання факторизації полягає в мінімізації зваженого квадрата відхилень для всіх елементів вихідної матриці R :

$$\min \sum_{i,j} c_{ij} (r_{ij} - x_i^T y_j)^2 \tag{3.6}$$

Де x_i^T – транспонована і – рядок матриці «користувач - латентна змінна»;

y_j – рядок j матриці «латентна змінна – об'єкт»

На етапі факторизації матриць пропонується використовувати метод ALS. Його особливість полягає в почерговому знаходженні мінімуму. На етапі оцінки отриманих рекомендацій використовується крива помилок ROC і відповідна метрика AUC Площа під кривою помилок AUC використовується для оцінки результатів прогнозування Експеримент удосконалення методу

показав, що для вибірки обсягом 100 тис. записів, показники після сортування негативних результатів збільшилися з 0,862 до 0,881.

Можна зазначити, що сортування засноване на попарному порівнянні переваг кожного користувача по відношенню до об'єктів у матриці вихідних даних. У результаті сформувався порядок елементів кожного користувача, що дозволило повисити точність методів оцінки рекомендаційних систем.

Розробляється рекомендаційна система дозволяє обробляти величезну кількість вхідної інформації - близько декількох десятків мільйонів записів. Досліджуємо її ефективність при різних розмірах вихідних даних. Скористаємося функцією обчислення метрики AUC calculatingOfAUC, розглянутої раніше. Параметри моделі вибрані найкращі (lambda = 5.0, alpha = 40.0, iterations = 20). Також заміряємо час побудови моделі.

Таблиця 4.4 – Оцінка ефективності системи

Кількість записів	AUC	Час побудови моделі, с
1000	0.687	1.26
5000	0.689	1.81
10000	0.713	1.98
100000	0.745	2.15
1000000	0.777	3.98
5000000	0.779	4.49
10000000	0.850	10.49
24296858	0.889	17.77

Як видно з росту метрики AUC, із збільшенням кількості записів у вихідному файлі з даними система дає більш точні рекомендації. Але і при порівняно невеликих обсягах початкової інформації будуть отримані не найгірші результати. Також, зі зростанням кількості оброблюваних даних збільшується і час побудови моделі.

З проведеного досвіду можна зробити висновок, що для оптимального співвідношення часових витрат і точності розрахунків потрібно близько 5 000 000 вихідних записів.

Були отримані схожі результати. Для порівняння було проведено налаштування параметрів алгоритму колаборативної фильтрації. Оцінки вийшли аналогічними. Але у першому варіанті можна автоматизувати процесс.

ВИСНОВКИ

Сервіси збирають інформацію про переваги користувачів і намагаються запропонувати їм корисні товари. На даний момент існує множина методів для формування рекомендацій, але всі вони мають свої переваги і недоліки. Саме тому дослідження в даній області були актуальні.

У роботі виконано дослідження методів оцінки рекомендаційних систем. Проаналізовано існуючі методи оцінки, на підставі проведеного аналізу запропоновано удосконалення оцінювання рекомендаційних систем на основі колаборативної фільтрації на основі пропущених даних з неявним зворотним зв’язком шляхом сортування негативних результатів в матриці вихідних даних.

В ході дослідження отримані такі результати: визначені існуючі рекомендаційні системи, виконано аналіз процесу роботи рекомендаційних систем, аналіз критеріїв та методів оцінювання рекомендаційних систем, побудовано удосконалену рекомендаційну систему на основі колаборативної фільтрації, побудовано контекстну діаграму моделі «Дослідження методів оцінки рекомендаційних систем», проведено експериментальну перевірку удосконаленого методу.

СПИСОК ЛІТЕРАТУРИ

1. Методичні вказівки щодо розробки та оформлення магістерської атестаційної роботи за спеціальністю 8.05010101 – Інформаційні управляючі системи та технології. Освітньо-кваліфікаційний рівень – магістр / Упоряд.: Левикін В.М., Міхнов Д.К., Саєнко В.І., Євланов М.В., Міхнова А.В., Керносов М.А. – Харків: ХНУРЕ, 2012. – 28 С.

2. Chalyi S. Доповнення вхідних даних рекомендаційної системи в ситуації циклічного холодного старту з використанням темпоральних обмежень типу «next» / S. Chalyi, V. Leshchynskyi, I. Leshchynska // Системи управління, навігації та зв'язку. Збірник наукових праць. – Полтава: ПНТУ, 2019. – Т. 4 (56). – С. 105-109. – doi:https://doi.org/10.26906/SUNZ.2019.4.105.

3. Чалий С.Ф., Лещинський В.О., Лещинська І.О. Моделювання контексту в рекомендаційних системах. Науковий журнал «Проблеми інформаційних технологій», 2018, №. 1(023). С. 21-26.

4. Chalyi S. Доповнення вхідних даних рекомендаційної системи в ситуації циклічного холодного старту з використанням темпоральних обмежень типу «next» / S. Chalyi, V. Leshchynskyi, I. Leshchynska // Системи управління, навігації та зв'язку. Збірник наукових праць. – Полтава: ПНТУ, 2019. – Т. 4 (56). – С. 105-109. – doi:https://doi.org/10.26906/SUNZ.2019.4.105.

5. Савчук Т.О., Застосування кластерного аналізу для колаборативної фільтрації / Т.О. Савчук, А.В.Сакалюк // Вісник Хмельницького національного університету. –2011 – №1– С. 186-192

6. Sarwar B. M. Item-based collaborative filtering recommendation algorithms / B. M. Sarwar, G. Karypis, J. A. Konstan // Proceedings of ACM WWW '01, pp. 285–295, ACM, 2001.

7. Hu Y., Koren Y. and Volinsky C. (2008), "Collaborative filtering for implicit feedback datasets", Data Mining, ICDM'08. Eighth IEEE International Conference on. IEEE, pp. 263–272.

8. Yehuda Koren, Robert Bell, and Chris Volinsky. (2009), “Matrix factorization techniques for recommender systems”, Computer No.8, pp. 30–37.

9. Linden G. Amazon.com recommendations: Item-to-item collaborative filltering / G. Linden, B. Smith, J. York // IEEE Internet Computing, vol. 7, no. 1, pp. 76–80, 2003.

10. Xiaoyuan Su and Taghi M. Khoshgoftaar "A Survey of Collaborative Filtering Techniques A Survey of Collaborative Filtering Techniques" // Hindawi Publishing Corporation, Advances in Artificial Intelligence archive, USA : 2009. — p. 1-19.

11. Jannach D., Zanker M., Felfernig A. Friedrich G. Recommender Systems. An Introduction. New York: Cambridge University Press 32 Avenue of the Americas, 2011. 352 P.

12. Melville P., Sindhwani V. Recommender systems. Encyclopedia of Machine Learning. 2010. p. 30

13. Guo G., Zhang J. and Yorke-Smith N. (2015), “TrustSVD: Collaborative Filtering with Both the Explicit and Implicit Influence of User Trust and of Item Ratings” AAAI, pp. 123–129

14. Christian Desrosiers and George Karypis. A comprehensive survey of neighborhoodbased recommendation methods. In Recommender systems handbook, pages 107–144. Springer, 2011.

15. Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In Data Mining, 2008. ICDM’08. Eighth IEEE International Conference on, pages 263–272. IEEE, 2008.

16. Daniel Lemire and Anna Maclachlan. Slope one predictors for online rating-based collaborative filtering. In SDM, volume 5, pages 1–5. SIAM, 2005.

17. Istv´an Pil´aszy and Domonkos Tikk. Recommending new movies: even a few ratings are more valuable than metadata. In Proceedings of the third ACM conference on Recommender systems, pages 93–100. ACM, 2009.

18. Steffen Rendle. Factorization machines with libfm. ACM Transactions on Intelligent Systems and Technology (TIST), 3(3):57, 2012.

19. Nathan Srebro, Tommi Jaakkola, et al. Weighted low-rank approximations. In ICML, volume 3, pages 720–727, 2003.

20. Cantador I., Konstas I., Jose J. Categorising social tags to improve folksonomy-based recommendations // Web Semantics: Science, Services and Agents on the World Wide Web. — 2011. — Vol. 9, no. 1. — P. 1–15.

21. Groh G., Ehmig C. Recommendations in taste related domains: collaborative filtering vs. social filtering // Proceedings of the 2007 international ACM conference on Supporting group work / Citeseer. — 2007. — P. 127–136.

22. Matrix factorization and neighbor based algorithms for the netflix prize problem / G. Tak´acs, I. Pil´aszy, B. N´emeth, D. Tikk // Proceedings of the 2008 ACM conference on Recommender systems / ACM. — 2008. — P. 267–274.