

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ**

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри

_____ Литвиненко О.Є.

«___» _____ 2021 р.

ДИПЛОМНИЙ ПРОЕКТ
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

**ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ
"БАКАЛАВР"**

Тема: _____ Програмний інструмент визначення користувацьких переваг _____

Виконавець: _____ Воропай Д.В. _____

Керівник: _____ Апенько Н.В. _____

Нормоконтролер: _____ Тупота Є.В. _____

Київ 2021

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії

Кафедра комп'ютеризованих систем управління

Освітнього ступеня бакалавр

Напрямок (спеціальність) 123 "Комп'ютерна інженерія"
(шифр, найменування)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Литвиненко О. Є.

« » 2021 р.

ЗАВДАННЯ на виконання дипломної роботи (проекту)

Воропая Дениса Вікторівича

(прізвище, ім'я, по батькові)

1. **Тема роботи:** Програмний інструмент визначення користувацьких переваг

затверджена наказом ректора від "04" лютого 2021 року № 135 /ст.

2. **Термін виконання роботи:** з 17.05.2021 до 20.06.2021

3. **Вихідні дані до роботи:** нейронна система для реалізації моделі спільної фільтрації

4. **Зміст пояснювальної записки (перелік питань, що підлягають розробці):**

1) аналіз принципів побудови рекомендаційних систем;

2) методи побудови системи визначення користувацьких переваг;

3) проекування нейронної мережі визначення користувацьких переваг.

5. **Перелік обов'язкового графічного матеріалу:**

1) програмні системи з оцінкою переваг користувачів;

2) формули перетину множини рекомендованих, релевантних і протестованих книг;

3) вікна сайту з ілюстрацією роботи системи рекомендацій;

4) схема алгоритму фільтрації книг.

6. Календарний план

№ п/п	Етапи виконання дипломного проекту	Термін виконання етапів	Примітка
1	Ознайомитись з постановкою задачі дипломного проектування	17.05.21-19.05.21	
2	Вивчити спеціальну літературу і технічну документацію	19.05.21-23.05.21	
3	Проаналізувати типові моделі спільної фільтрації	24.05.21-26.05.21	
4	Проаналізувати методи і алгоритми роботи системи	27.05.21-29.05.21	
5	Провести опис роботи системи	30.05.21-02.06.21	
6	Написати пояснювальну записку	03.06.21-10.06.21	
7	Підготувати презентацію та графічні матеріали	11.06.21-14.06.21	

7. Дата видачі завдання « 17 » травня 2021 р.

Керівник дипломного проекту _____ Апенько Н.В.
(підпис)

Завдання прийняв до виконання _____ Воропай Д.В.
(підпис студента)

РЕФЕРАТ

Пояснювальна записка до дипломного проекту “Програмний інструмент визначення користувацьких переваг”: 64 с., 20 рис., 22 літературних джерела, 1 додаток.

КОРИСТУВАЦЬКА ПЕРЕВАГА, ГРУПОВА ФІЛЬТРАЦІЯ, НЕЙРОННА МЕРЕЖА, АНАЛІЗ ПОПИТУ, МАШИННЕ НАВЧАННЯ.

Мета дипломної роботи – аналіз методів визначення користувацьких переваг для реалізації програмного забезпечення групової фільтрації.

Об'єкт дослідження – визначення користувацьких переваг.

Предмет дослідження – нейронна система визначення користувацьких переваг.

Наукова значимість полягає у реалізації нейронної мережі визначення користувацьких переваг.

Практична значимість полягає у розробці програмного забезпечення, що планується використовувати у навчальному процесі, як практична складова у вивченні застосування нейронних мереж, та на веб-ресурсах для формування переліку пропозицій.

Прогнозні припущення щодо подальшого розвитку матеріалів роботи полягає у навчанні нейронної мережі для покращення результатів групової фільтрації.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ	6
ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ ПРИНЦИПІВ ПОБУДОВИ РЕКОМЕНДАЦІЙНИХ СИСТЕМ	11
1.1. Основні визначення	11
1.2. Огляд різних типів алгоритмів систем рекомендацій	14
1.3. Оцінка системи рекомендацій	20
1.4. Аналіз алгоритмів класифікації	22
1.5. Висновки до розділу	25
РОЗДІЛ 2 МЕТОДИ ПОБУДОВИ СИСТЕМИ ВИЗНАЧЕННЯ КОРИСТУВАЦЬКИХ ПЕРЕВАГ	28
2.1. Аналіз існуючих підходів до визначення користувачьких переваг	29
2.2. Реалізація методу фільтрації на основі вмісту	31
2.3. Реалізація методу спільної фільтрації	33
2.4. Реалізація методу гібридної фільтрації	38
2.5. Оціночні показники для рекомендаційних алгоритмів	41
2.5. Висновки до розділу	42
РОЗДІЛ 3 ПРОЕКТУВАННЯ НЕЙРОННОЇ МЕРЕЖІ ВИЗНАЧЕННЯ КОРИСТУВАЦЬКИХ ПЕРЕВАГ	43
3.1. Етапи машинного експерименту	43
3.2. Випробування моделі	48
3.3. Точність і повнота використання Програмних засобів	53
3.4. Висновки до розділу	58
ВИСНОВКИ	59
СПИСОК БІБЛЮГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ	63
ДОДАТОК А	64

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

<i>ANN</i>	–	<i>artificial neural networks</i>
<i>ARM</i>	–	<i>advanced risc machine</i>
<i>ISO</i>	–	<i>international organization for standardization</i>
ПММ	–	приховані Марковські моделі
ШНМ	–	штучна нейронна мережа

ВСТУП

Протягом останніх декількох десятиліть, із зростанням *Youtube*, *Amazon*, *Netflix* та багатьох інших таких веб-служб, системи рекомендацій зайняли дедалі більше місця в нашому житті. Від електронної комерції (пропонуйте покупцям статті, які можуть їх зацікавити) до реклами в Інтернеті (пропонуйте користувачам правильний вміст, що відповідає їхнім уподобанням), системи рекомендацій сьогодні неминучі в наших щоденних онлайн-подорожах.

У дуже загальному плані рекомендаційні системи - це алгоритми, спрямовані на те, щоб пропонувати користувачам відповідні предмети (це книги для перегляду, текст для читання, товари, що купуються, або щось інше, залежно від галузей).

Системи, що рекомендують, справді є критично важливими в деяких галузях, оскільки вони можуть приносити величезний прибуток, коли вони ефективні, або також можуть бути способом значно виділитися серед конкурентів. Як доказ важливості систем, що рекомендують, можемо згадати, що кілька років тому *Netflix* організував виклики («приз *Netflix*»), метою яких було створити систему рекомендацій, яка працює ефективніше, ніж її власний алгоритм з призом 1 мільйон доларів на виграш.

Система рекомендацій визначається як стратегія прийняття рішень для користувачів у складних інформаційних середовищах[1]. Крім того, система рекомендацій була визначена з точки зору електронної комерції як інструмент, який допомагає користувачам здійснювати пошук у записах знань, які пов'язані з інтересами та уподобаннями користувачів[1]. Система рекомендацій була визначена як засіб сприяння та посилення соціального процесу з використанням рекомендацій інших для прийняття рішень, коли немає достатніх особистих знань чи досвіду альтернатив.[1]. Системи, що рекомендують, вирішують проблему перевантаження інформації, з якою зазвичай стикаються користувачі, надаючи їм персоналізований ексклюзивний вміст та рекомендації щодо

обслуговування. Нещодавно були розроблені різні підходи до побудови систем рекомендацій, які можуть використовувати або спільну фільтрацію, і фільтрування на основі вмісту, або гібридну фільтрацію.[9],[10],[11]. Техніка спільної фільтрації є найбільш зрілою та найбільш часто застосовуваною. Спільна фільтрація рекомендує елементи шляхом виявлення інших користувачів зі схожим смаком; він використовує їх думку, щоб рекомендувати товари активному користувачеві. Спільні системи рекомендацій були впроваджені в різних сферах застосування. GroupLens - це архітектура, заснована на новинах, яка використовувала спільні методи, допомагаючи користувачам знаходити статті з масивної бази даних новин[12]. Ringo - це система фільтрації соціальної інформації в Інтернеті, яка використовує спільну фільтрацію для створення профілю користувачів на основі їх рейтингу в музичних альбомах[10]. Amazon використовує алгоритми диверсифікації тем для вдосконалення своїх рекомендацій[13]. Система використовує метод спільної фільтрації для подолання проблеми масштабованості шляхом створення таблиці подібних елементів в автономному режимі за допомогою матриці від елемента до елемента. Потім система рекомендує інші продукти, подібні в Інтернеті відповідно до історії покупок користувачів. З іншого боку, методи, що базуються на вмісті, відповідають ресурсам вмісту характеристикам користувачів. Методи фільтрації на основі вмісту зазвичай базують свої прогнози на інформації користувача, і вони ігнорують внески інших користувачів, як у випадку з техніками спільної роботи[14],[15]. Fab в значній мірі покладається на рейтинги різних користувачів, щоб створити навчальний набір, і це приклад системи рекомендацій на основі вмісту. Деякі інші системи, які використовують фільтрацію на основі вмісту, щоб допомогти користувачам знаходити інформацію в Інтернеті, включають Letizia[16]. Система використовує користувальницький інтерфейс, який допомагає користувачам в Інтернеті; він може відслідковувати режим перегляду користувача, щоб передбачити сторінки, які його можуть зацікавити. Pazzani та ін.[17]розроблений інтелектуальний агент, який намагається передбачити, які веб-сторінки зацікавлять користувача за

допомогою наївний байєсівський класифікатор. Агент дозволяє користувачеві проводити навчальні екземпляри, оцінюючи різні сторінки як гарячі, так і холодні.

Незважаючи на успіх цих двох методів фільтрації, було виявлено кілька обмежень. Деякі проблеми, пов'язані з методами фільтрації на основі вмісту, - це обмежений аналіз вмісту, надмірна спеціалізація та розрідженість даних [12]. Крім того, спільні підходи демонструють проблеми холодного старту, розрідженості та масштабованості. Ці проблеми зазвичай знижують якість рекомендацій. З метою пом'якшення деяких виявлених проблем було запропоновано гібридну фільтрацію, яка поєднує два або більше методів фільтрації різними способами для підвищення точності та продуктивності систем, що рекомендують [19],[10]. Ці методи поєднують два або більше підходів до фільтрації, щоб використати свої сильні сторони, вирівнюючи відповідні слабкі місця [11]. Їх можна класифікувати на основі їх діяльності на зважений гібрид, змішаний гібрид, комутаційний гібрид, комбінований комбінований гібрид, каскадний гібрид, доповнений функціями гібрид та метарівневий гібрид [12]. Сьогодні підходи до спільної фільтрації та фільтрації на основі вмісту широко застосовуються, застосовуючи методи, що базуються на вмісті та спільній роботі, а результати їх прогнозування згодом поєднують або додають характеристики спільної фільтрації на основі вмісту і навпаки. Нарешті, може бути розроблена загальна уніфікована модель, яка включає як властивості фільтрації на основі вмісту, так і спільні [12]. Проблема розрідженості даних та холодного запуску була вирішена шляхом поєднання рейтингів, характеристик та демографічної інформації про предмети в каскадній гібридній техніці рекомендацій у [13]. У [14] було запропоновано гібридний підхід до спільної фільтрації для використання масової таксономічної інформації, призначеної для суворої класифікації продукції для вирішення проблеми розрідженості даних рекомендацій CF, заснованої на формуванні профілів на основі висновку про супертематичний бал та диверсифікацію тем. Гібридна методика рекомендацій також пропонується в Ghazantar та Prigel-Benett [13], і для цього

використовується профіль окремого користувача на основі вмісту, щоб знайти схожих користувачів, які використовуються для прогнозування. У роботі Sarwar та співавт.[15], спільна фільтрація поєднувалася з агентом фільтрації інформації. Тут автори запропонували основу для інтеграції агентів фільтрації на основі вмісту та спільної фільтрації. Гібридний алгоритм рекомендацій застосовується багатьма програмами в результаті нової проблеми користувача з методами фільтрації на основі вмісту та середньої проблеми користувача спільної фільтрації[16]. Простий і зрозумілий метод поєднання фільтрації на основі вмісту та спільної роботи був запропонований Каннінгамом та ін.[17]. Система рекомендацій щодо музики, що поєднує інформацію про теги, кількість п'єс та соціальні стосунки, була запропонована в Констас та ін.[18]. Для того, щоб визначити кількість сусідів, яких можна автоматично підключити на соціальній платформі, Лі та Брусиловський[19] вбудована соціальна інформація в алгоритм спільної фільтрації. Байєсівський модель змішаних ефектів що інтегрує рейтинги користувачів, функції користувачів та елементи в єдину уніфіковану структуру, запропоноване Condiff et al.[10].

РОЗДІЛ 1

АНАЛІЗ ПРИНЦИПІВ ПОБУДОВИ РЕКОМЕНДАЦІЙНИХ СИСТЕМ

1.1. Основні визначення

Мета системи рекомендацій - пропонувати користувачам відповідні пункти. Для досягнення цього завдання існує дві основні категорії методів: методи спільної фільтрації та методи на основі вмісту. Перш ніж детальніше вивчати деталі конкретних алгоритмів, давайте коротко обговоримо ці дві основні парадигми.

1.1.1. Методи спільної фільтрації

Методи спільної роботи для рекомендованих систем - це методи, які базуються виключно на минулих взаємодіях, зафіксованих між користувачами та елементами, з метою вироблення нових рекомендацій. Ці взаємодії зберігаються у так званій "матриці взаємодій між елементами користувача".

Тоді основна ідея, яка керує методами спільної роботи, полягає в тому, що цих минулих взаємодій між елементами користувача достатньо для виявлення схожих користувачів та / або подібних елементів та прогнозування на основі цих оцінених близькості.

Клас спільних алгоритмів фільтрації поділяється на дві підкатегорії, які зазвичай називаються підходами на основі пам'яті та моделями. Підходи, засновані на пам'яті (рис. 1.1), безпосередньо працюють зі значеннями записаних взаємодій, припускаючи відсутність моделі, і по суті засновані на пошуку найближчих сусідів (наприклад, знайти найближчих користувачів та запропонувати найбільш популярні предмети серед цих сусідів).

Кафедра КСУ				НАУ 21 02 36 000 ПЗ			
Виконав	Воропай Д.В.			Аналіз принципів побудови рекомендаційних систем	Літера	Аркуш	Аркушів
Керівник	Апенько Н.В.				Д	11	71
Консульт.					123 СП 437		
Норм. контр.	Тупота Є.В.						
Зав. Каф.	Литвиненко О.Є.						

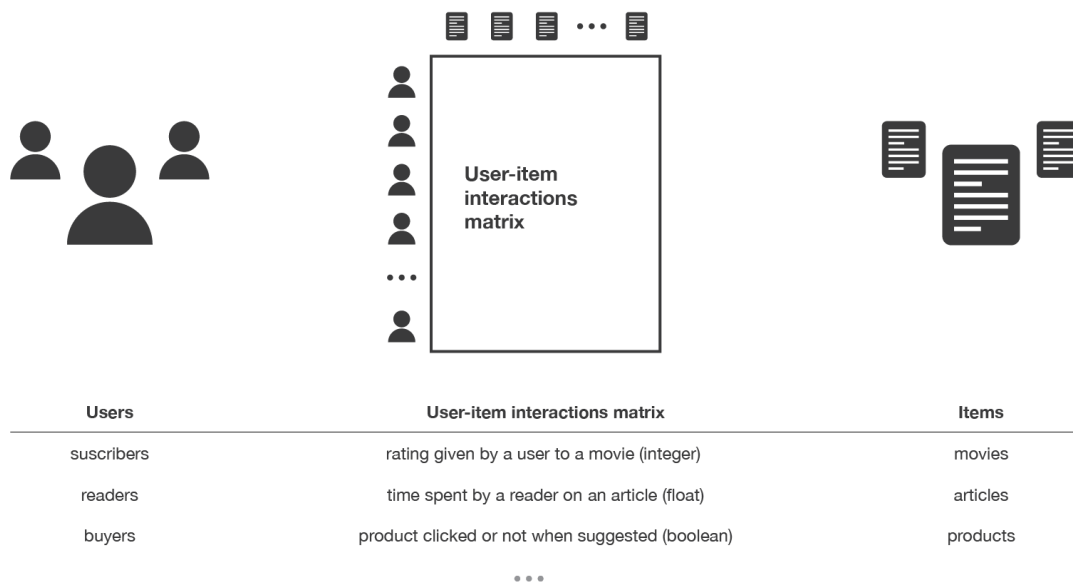


Рис. 1.1. Ілюстрація матриці взаємодій елемент-користувач.

Підходи, що базуються на моделях, передбачають основоположну «генеративну» модель, яка пояснює взаємодію між елементами користувача та намагається виявити її, щоб робити нові прогнози.

Головною перевагою спільних підходів є те, що вони не вимагають інформації про користувачів чи предмети, і тому їх можна використовувати в багатьох ситуаціях. Більше того, чим більше користувачів взаємодіє з елементами, тим більше нових рекомендацій стає точнішим: для фіксованого набору користувачів та предметів нові взаємодії, зафіксовані з часом, приносять нову інформацію та роблять систему все більш ефективною.

Однак, оскільки він враховує лише попередні взаємодії для вироблення рекомендацій, спільна фільтрація страждає від "проблеми холодного старту": неможливо рекомендувати що-небудь новим користувачам або рекомендувати новий елемент будь-яким користувачам, і багато користувачів або елементи мають занадто мало взаємодій для ефективної обробки. Цей недолік можна вирішити по-різному: рекомендувати випадкові предмети новим користувачам або нові предмети випадковим користувачам (випадкова стратегія), рекомендувати популярні предмети новим користувачам або нові предмети найбільш активним користувачам (стратегія максимального очікування), рекомендувати набір різних елементи для нових користувачів або нові предмети

для набору різних користувачів (дослідницька стратегія) або, нарешті, за допомогою методу, що не співпрацює, для раннього життя користувача або елемента.

1.1.2. Методи на основі змісту

На відміну від методів спільної роботи, які покладаються лише на взаємодію між елементами користувача, підходи на основі змісту використовують додаткову інформацію про користувачів та / або елементи. Якщо розглянемо приклад системи рекомендації фільмів, цією додатковою інформацією може бути, наприклад, вік, стать, робота чи будь-яка інша особиста інформація для користувачів, а також категорія, основні актори, тривалість чи інші характеристики для фільмів (предметів).

Потім ідея методів, що базуються на вмісті, полягає в спробі побудувати модель, засновану на наявних «особливостях», що пояснюють спостережувані взаємодії між елементами користувача. Досі розглядаючи користувачів та книги, спробуємо, наприклад, змодельовати той факт, що молоді жінки, як правило, оцінюють деякі книги, що молоді чоловіки, як правило, оцінюють деякі інші книги тощо. Якщо нам вдається отримати таку модель, тоді робити нові прогнози для користувача досить просто: нам просто потрібно подивитися профіль (вік, стать, тощо) цього користувача і, виходячи з цієї інформації, визначити відповідні книги для припустити. Методи, що базуються на контенті, страждають набагато менше від проблеми холодного старту, ніж спільні підходи: нових користувачів або елементи можна описати за їх характеристиками (вмістом), тому для цих нових сутностей можна зробити відповідні пропозиції. Від цього недоліку логічно потерпатимуть лише нові користувачі або елементи з раніше не баченими функціями, але як тільки система стане досить старою, у неї майже немає шансів.

Далі в цьому дописі далі обговоримо підходи, засновані на вмісті, і побачимо, що залежно від нашої проблеми можуть бути використані різні класифікаційні та регресійні моделі, починаючи від дуже простих і набагато складніших моделей.

1.1.3. Порівняння методів спільної фільтрації і аналізу змісту

У методах спільної роботи на основі пам'яті не передбачається прихованої моделі. Алгоритми безпосередньо працюють із взаємодіями елементів користувача: наприклад, користувачі представлені їх взаємодією з елементами, а пошук найближчих сусідів за цими поданнями використовується для створення пропозицій. Оскільки не передбачається жодної прихованої моделі, ці методи мають теоретично низький зсув, але велику дисперсію.

У методах спільної роботи на основі моделі передбачається деяка модель прихованої взаємодії. Модель навчена реконструювати значення взаємодії між елементами користувача на основі власного представлення користувачів та елементів. Потім на основі цієї моделі можна робити нові пропозиції. Користувачі та елементи, приховані зображення, вилучені моделлю, мають математичне значення, яке для людини може бути важко інтерпретувати. Оскільки передбачається (досить безкоштовна) модель взаємодії між елементами користувача, теоретично ці методи мають вищий зсув, але меншу дисперсію, ніж методи, які не передбачають прихованої моделі. У методах, заснованих на вмісті, також передбачається деяка модель прихованої взаємодії. Однак тут модель забезпечена вмістом, який визначає представлення користувачів та / або елементів: наприклад, користувачі представлені заданими функціями, і намагаємось змодельювати для кожного елемента тип профілю користувача, якому цей елемент подобається чи ні. Тут, як і для методів спільної роботи на основі моделі, передбачається модель взаємодії між елементами користувача.

1.2. Огляд різних типів алгоритмів систем рекомендацій

1.2.1. Спільні підходи на основі пам'яті

Основні характеристики підходів користувач-користувач та елемент-елемент наближаються до того, що вони використовують лише інформацію з матриці взаємодії елемент-користувач, і вони не приймають жодної моделі для вироблення нових рекомендацій.

Для того, щоб зробити нову рекомендацію користувачеві, метод «користувач-користувач» приблизно намагається ідентифікувати користувачів з найбільш подібним «профілем взаємодії» (найближчі сусіди), щоб запропонувати елементи, які є найбільш популярними серед цих сусідів (а саме: «новий» для нашого користувача). Цей метод називають "орієнтованим на користувача", оскільки він представляє користувачів на основі їх взаємодії з предметами та оцінює відстань між користувачами.

Припустимо, що хочемо дати рекомендацію для певного користувача. По-перше, кожен користувач може бути представлений його вектором взаємодії з різними елементами ("його рядок" у матриці взаємодії). Тоді можемо обчислити якусь „подібність” між нашим зацікавленим користувачем та іншими користувачами. Цей показник подібності такий, що двох користувачів із однаковою взаємодією на однакових предметах слід вважати близькими. Коли обчислюється схожість з усіма користувачами, можемо зберегти k -найближчих сусідів для нашого користувача, а потім запропонувати найпопулярніші предмети серед них (дивлячись лише на елементи, з якими ще не взаємодіяв наш референтний користувач).

При обчисленні подібності між користувачами кількість «загальних взаємодій» (рис. 1.2).

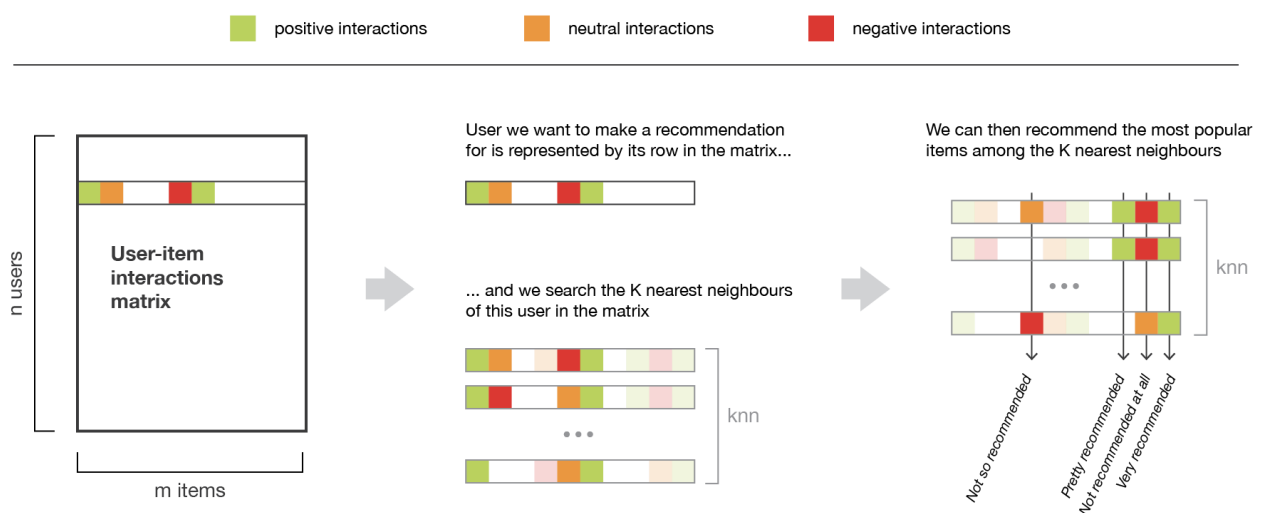


Рис. 1.2. Ілюстрація методу користувач-користувач

Більшу частину часу хочемо уникнути того, щоб хтось, хто має лише одну спільну взаємодію з нашим користувачем-референтом, міг мати 100% співпадіння і вважатись «більш близьким», ніж той, хто має 100 спільних взаємодій і погоджується «лише» на 98%. Вважаємо, що два користувачі схожі, якщо вони однаково взаємодіяли з великою кількістю загальних предметів (схожий рейтинг, схожий час).

Щоб зробити нову рекомендацію користувачеві, ідея методу елемент-елемент полягає у пошуку елементів, схожих на ті, з якими користувач вже «позитивно» взаємодіяв. Два елементи вважаються подібними, якщо більшість користувачів, які взаємодіяли з ними обома, робили це подібним чином. Зазначено, що цей метод «орієнтований на предмет», оскільки він представляє предмети на основі взаємодії користувачів з ними та оцінює відстань між цими предметами.

Припустимо, що хочемо дати рекомендацію для певного користувача. По-перше, розглядаємо елемент, який найбільше сподобався цьому користувачеві, і представляємо його (як і всі інші елементи) за його вектором взаємодії з усіма користувачами («його стовпець» у матриці взаємодії). Потім можемо обчислити подібність між «найкращим предметом» та всіма іншими предметами. Після обчислення подібності можемо перевести k -найближчих сусідів у вибраній «найкращий предмет», що є новим для нашого користувача, який цікавить, і рекомендувати ці елементи.

Зверніть увагу, що для того, щоб отримати більш відповідні рекомендації, можемо виконувати цю роботу не лише для улюбленого елемента користувача, а замість нього розглядати n бажаних елементів. У цьому випадку можемо рекомендувати предмети, близькі до кількох із цих бажаних предметів (рис. 1.3).

Метод користувач-користувач заснований на пошуку схожих користувачів з точки зору взаємодії з предметами. Оскільки, загалом, кожен користувач взаємодіяв лише з кількома елементами, це робить метод досить чутливим до будь-яких записаних взаємодій (велика дисперсія).

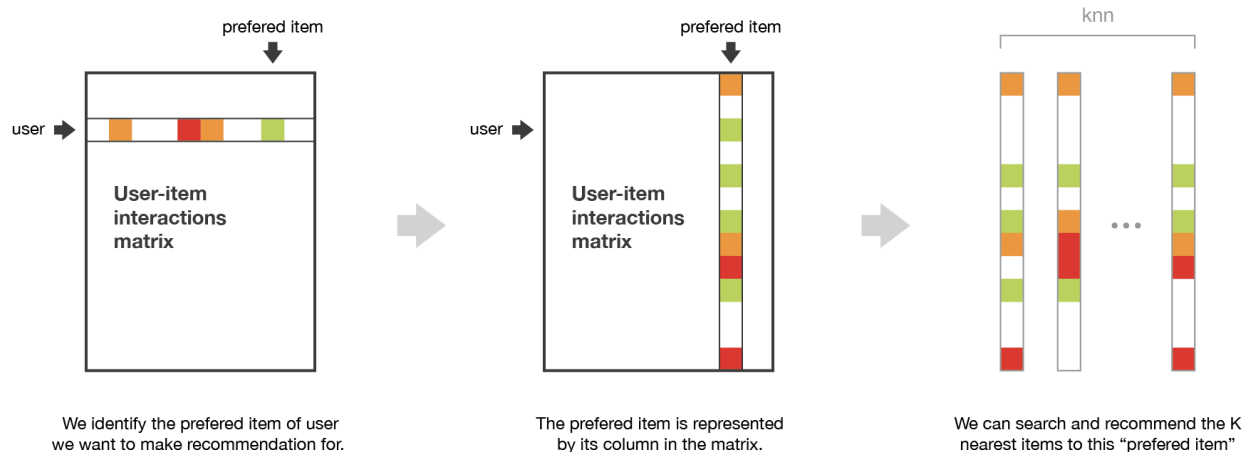


Рис. 1.3. Ілюстрація методу предмет-предмет

З іншого боку, оскільки остаточна рекомендація ґрунтується лише на взаємодії, зареєстрованій для користувачів, схожих на нашого користувача, який нас цікавить, отримуємо більш персоналізовані результати (низький зсув). І навпаки, метод предмет-предмет заснований на пошуку подібних елементів з точки зору взаємодії між елементами користувача. Оскільки, як правило, багато користувачів взаємодіють з елементом, пошук району набагато менш чутливий до одиничних взаємодій (менша дисперсія). Як аналог, взаємодія, що надходить від усіх типів користувачів (навіть користувачів, які дуже відрізняються від нашого користувача-референта), тоді розглядається в рекомендації, роблячи метод менш персоналізованим (більш упередженим). Таким чином, цей підхід менш персоніфікований, ніж підхід користувач-користувач, але більш надійний (рис. 1.4).

Однією з найбільших вад спільної фільтрації на основі пам'яті є те, що їх масштабування не відбувається легко: генерування нової рекомендації може зайняти надзвичайно багато часу для великих систем.

Дійсно, для систем з мільйонами користувачів та мільйонами елементів, крок пошуку найближчих сусідів може стати нерозв'язним, якщо не бути ретельно розробленим (алгоритм KNN має складність $O(ndk)$ з n кількістю користувачів, d кількістю елементів та k кількістю розглянутих сусідів).



Рис. 1.4. Ілюстрація різниці між методом елемент-елемент та користувачем-користувачем

Для того, щоб зробити обчислення більш зручними для величезних систем, обидва можемо скористатися розрідженістю матриці взаємодії при розробці нашого алгоритму або використовувати приблизні методи найближчих сусідів (ANN).

У більшості рекомендаційних алгоритмів необхідно бути гранично обережними, щоб уникнути ефекту «збагачення» для популярних предметів та уникати застрягання користувачів у тому, що можна назвати «зоною обмеження інформації». Іншими словами, не хочемо, щоб наша система, як правило, рекомендувала все більше і більше лише популярних предметів, а також не хочемо, щоб наші користувачі отримували рекомендації лише щодо предметів, надзвичайно близьких до того, який їм уже сподобався, без шансів дізнатися більше нові предмети, які також можуть їм сподобатися (оскільки ці предмети недостатньо «близькі», щоб їх можна було запропонувати). Якщо, як вже згадували, ці проблеми можуть виникнути в більшості алгоритмів рекомендацій, це особливо актуально для спільних, заснованих на пам'яті. Справді, за відсутності моделі «регуляризації», такий феномен можна підкреслювати та спостерігати частіше.

1.2.2. Спільні підходи на основі моделей

Модельні підходи на основі співпраці покладаються лише на інформацію про взаємодію між елементами користувача та передбачають приховану модель, яка повинна пояснювати ці взаємодії. Наприклад, алгоритми факторизації матриць полягають у розкладанні величезної та розрідженої матриці взаємодії між користувачем та елементом на добуток двох менших і щільних матриць: матриця з коефіцієнтом користувача (що містить уявлення користувачів), яка множить матрицю з фактором (містить представлення елементів).

1.2.3. Факторизація матриці

Основним припущенням факторизації матриць є те, що існує досить низький розмірний прихований простір функцій, в якому можемо представляти як користувачів, так і елементи, і такі, що взаємодія між користувачем і елементом може бути отримана шляхом обчислення точкового добутку відповідних щільних векторів в тому просторі.

Наприклад, у нас є матриця рейтингу фільмів користувачів. Для моделювання взаємодії між користувачами та фільмами можна припустити, що:

- є деякі особливості, які досить добре описують (і розрізняють) книги;
- ці функції також можна використовувати для опису користувацьких уподобань (високі значення для функцій, які сподобалися користувачеві, низькі значення в іншому випадку).

Однак не хочемо чітко надавати ці функції нашій моделі (як це можна зробити для підходів на основі вмісту, які опишемо пізніше). Натомість вважаємо за краще дозволити системі самостійно відкривати ці корисні функції та робити власні уявлення як користувачів, так і елементів. Оскільки вони вивчаються і не даються, витягнуті функції, взяті окремо, мають математичне значення, але не мають інтуїтивного тлумачення (і, отже, їх важко, а то й неможливо зрозуміти як людину). Однак незвичайно, що в результаті структури, що виходять із такого типу алгоритму, надзвичайно близькі до інтуїтивного розкладання, про яке людина могла б подумати. Дійсно, наслідком такої факторизації є те, що близькі користувачі з точки зору уподобань, а також близькі елементи з точки зору

характеристик в кінцевому підсумку мають тісні уявлення у прихованому просторі.

1.3. Оцінка системи рекомендацій

Що стосується будь-якого алгоритму машинного навчання, повинні мати можливість оцінити ефективність наших систем рекомендацій, щоб вирішити, який алгоритм найкраще відповідає нашій ситуації. Методи оцінки для систем, що рекомендують, в основному можна розділити на два набори: оцінка на основі чітко визначених метрик та оцінка в основному на основі людського судження та оцінки задоволеності.

1.3.1. Оцінка на основі метрик

Якщо наша система рекомендацій базується на моделі, яка виводить числові значення, такі як прогнозування рейтингів або ймовірності збігу, можемо оцінити якість цих результатів дуже класичним способом, використовуючи метрику вимірювання похибки, таку як, наприклад, середньоквадратична помилка (*MSE*). У цьому випадку модель тренується лише на частині доступних взаємодій і тестується на решті.

Однак, якщо наша система рекомендацій базується на моделі, яка передбачає числові значення, також можемо зробити бінаризацію цих значень класичним підходом порогових значень (значення, що перевищують поріг, є позитивними, а значення нижче - негативними), і оцінити модель більш «класифікаційно». Дійсно, оскільки набір даних про минулі взаємодії між елементами користувача також є двійковим (або його можна бінаризувати за допомогою порогового значення), тоді можемо оцінити точність (а також точність та відкликання) бінаризованих виходів моделі на тестовому наборі даних взаємодій, що не використовуються для навчання.

Якщо розглянемо систему рекомендацій, яка не базується на числових значеннях і яка повертає лише список рекомендацій (наприклад, користувач-користувач або елемент-елемент, що базуються на підході *knn*), все одно можемо

визначити точність, подібну метриці, оцінка частки рекомендованих предметів, які дійсно підходять нашому користувачеві. Для оцінки цієї точності не можемо взяти до уваги рекомендовані елементи, з якими наш користувач не взаємодіяв, і повинні розглядати лише елементи з тестового набору даних, про які маємо відгуки користувачів.

1.3.2. Оцінка на основі людини

Розробляючи систему рекомендацій, можемо бути зацікавлені не лише отримати модель, яка виробляє рекомендації, в яких дуже впевнені, але можемо також очікувати деяких інших хороших властивостей, таких як різноманітність та пояснюваність рекомендацій.

Як зазначалося в розділі про співпрацю, абсолютно хочемо уникнути застрягання користувача в тому, що раніше називали зоною обмеження інформації. Поняття "випадковість" часто використовується для вираження тенденції, яку модель має чи не створює таку зону обмеження (різноманітність рекомендацій). Випадковість, яку можна оцінити, обчислюючи відстань між рекомендованими предметами, не повинна бути занадто низькою, оскільки вона створює зони обмеження, але також не повинна бути занадто високою, оскільки це означатиме, що недостатньо враховуємо інтереси наших користувачів під час створення рекомендації (розвідка проти експлуатації). Таким чином, щоб внести різноманітність у запропонований вибір, хочемо рекомендувати предмети, які дуже добре підходять нашому користувачеві і не надто схожі один на одного. Наприклад, замість того, щоб рекомендувати користувачеві "Почати війни" 1, 2 та 3,

Пояснюваність - ще один ключовий момент успіху алгоритмів рекомендацій. Дійсно, було доведено, що якщо користувачі не розуміють, чому їх рекомендували як конкретний товар, вони, як правило, втрачають довіру до системи рекомендацій. Отже, якщо розробляємо модель, яку можна чітко пояснити, можемо додати, коли робимо рекомендації, невелике речення, в якому зазначається, чому рекомендований товар.

Різноманітність та пояснюваність може бути суттєво важко оцінити, можемо помітити, що також досить важко оцінити якість рекомендації, яка не належить до набору даних тестування: як дізнатися, чи нова рекомендація актуальне перед тим, як насправді рекомендувати його нашому користувачеві? З усіх цих причин іноді може бути спокусливо протестувати модель у „реальних умовах”. Оскільки метою системи, що рекомендує, є створення дії (перегляд фільму, придбання продукту, прочитання статті тощо), справді можемо оцінити її здатність генерувати очікувану дію. Наприклад, система може бути запущена у виробництво, застосовуючи підхід *A / B* тестування, або може бути протестована лише на вибірці користувачів. Однак такі процеси вимагають певного рівня довіри до моделі.

1.4. Аналіз алгоритмів класифікації

Системи, що рекомендують, мають на меті прогнозувати інтереси користувачів та рекомендувати товари, які цілком імовірно для них цікаві. Вони є одними з найпотужніших систем машинного навчання, які реалізують інтернет-магазини для стимулювання продажів.

Дані, необхідні для систем, що рекомендують, впливають із явних оцінок користувачів після перегляду фільму чи прослуховування пісні, з неявних запитів пошукової системи та історії покупок або з інших знань про самих користувачів / предмети.

Компанії, що використовують системи рекомендацій, зосереджуються на збільшенні продажів завдяки дуже персоналізованим пропозиціям та покращеній роботі з клієнтами.

Рекомендації зазвичай пришвидшують пошук і полегшують користувачам доступ до вмісту, який їх цікавить, і дивують їх пропозиціями, яких вони ніколи б не шукали.

Більше того, компанії можуть залучати та утримувати клієнтів, надсилаючи електронні листи із посиланнями на нові пропозиції, що відповідають інтересам

одержувачів, або пропозиції фільмів та телешоу, що відповідають їхнім профілям.

Користувач починає відчувати себе відомим і зрозумілим і, швидше за все, купує додаткові продукти або споживає більше вмісту. Знаючи, чого хоче користувач, компанія отримує конкурентні переваги, і загроза втрати клієнта для конкурента зменшується.

Забезпечення доданої вартості для користувачів шляхом включення рекомендацій до систем та продуктів є привабливим. Крім того, це дозволяє компаніям випередити своїх конкурентів і врешті-решт збільшити свої прибутки.

Системи, що рекомендують, функціонують із двома видами інформації:

- характерна інформація. Це інформація про елементи (ключові слова, категорії тощо) та користувачів (уподобання, профілі тощо).
- взаємодія між елементами користувача. Це така інформація, як рейтинги, кількість покупок, вподобань тощо.

Виходячи з цього, ми можемо розрізнити три алгоритми, що використовуються в системах, що рекомендують:

- системи на основі вмісту, які використовують характерну інформацію.
- Спільні системи фільтрації, які базуються на взаємодії між елементами користувача.
- гібридні системи, які поєднують обидва типи інформації з метою уникнення проблем, що виникають при роботі лише з одним видом.

Далі ми трохи глибше заглибимось у системи фільтрації на основі вмісту та спільної роботи та побачимо, чим вони відрізняються.

Системи спільної фільтрації базуються на припущенні, що якщо користувачеві подобається елемент А, а іншому користувачеві подобається той самий елемент А, а також інший елемент, елемент В, перший користувач також може бути зацікавлений у другому елементі. Отже, вони мають на меті передбачити нові взаємодії на основі історичних. Для досягнення цієї мети існує два типи методів: на основі пам'яті та на основі моделі.

Є два підходи: перший визначає скупчення користувачів і використовує взаємодію одного конкретного користувача для прогнозування взаємодії інших

подібних користувачів. Другий підхід визначає кластери елементів, які були оцінені користувачем А, і використовує їх для прогнозування взаємодії користувача А з іншим, але схожим елементом В. Ці методи зазвичай стикаються з великими проблемами при розрідженій матриці, оскільки кількість взаємодій між елементами користувача може бути занадто малою для створення кластерів високої якості.

Методи на основі моделі засновані на машинному навчанні та методах аналізу даних. Мета - навчити моделі вміння робити прогнози. Наприклад, ми могли б використати існуючі взаємодії з елементами користувача, щоб навчити модель прогнозувати 5 найпопулярніших елементів, які користувачеві можуть сподобатися найбільше. Однією з переваг цих методів є те, що вони можуть рекомендувати більшу кількість елементів більшій кількості користувачів, порівняно з іншими методами, такими як заснована на пам'яті. Ми говоримо, що вони мають велике покриття, навіть при роботі з великими розрідженими матрицями.

Є дві основні проблеми з системами спільної фільтрації:

1. Холодний старт: ми повинні мати достатньо інформації (взаємодії між елементами користувача) для роботи системи. Якщо ми налаштовуємо новий сайт електронної комерції, ми не можемо давати рекомендації, поки користувачі не взаємодіють із значною кількістю елементів.

2. Додавання нових користувачів / елементів до системи: чи це новий користувач, чи елемент, ми не маємо попередньої інформації про них, оскільки вони не мають взаємодій.

Ці проблеми можна усунути, попросивши користувачів під час реєстрації даних іншого типу (стать, вік, інтереси тощо), а також використовуючи метаінформацію з елементів, щоб мати можливість пов'язати їх з іншими існуючими елементами в базу даних.

Існує дві методики побудови спільної системи фільтрації: повністю підключені нейронні мережі та Item2vec.

Одним з класичних підходів є факторизація матриць. Мета полягає в тому, щоб заповнити невідоме в матриці взаємодій між елементами користувача

(назвемо це RR). Уявіть, що у нас якимось магічним чином є дві матриці UU та Π , такі, що $U \setminus \text{раз } IU \times I$ дорівнює RR у відомих записах. Використовуючи продукт $U \setminus \text{times } IU \times I$, ми також матимемо значення для невідомих записів RR , які потім можна використовувати для формування рекомендацій.

За допомогою онлайн-методів (також званих A / B -тестуванням) реакція користувачів вимірюється з урахуванням рекомендацій. Наприклад, ви можете виміряти, коли користувач натискає на рекомендовані елементи - а також коефіцієнт конверсії - і оцінювати прямий вплив системи. Такий підхід до оцінки є ідеальним, хоча його зазвичай важко реалізувати, оскільки єдиний спосіб проведення експериментів - це взаємодія із системою, яка вже є у виробництві. Будь-який невдалий експеримент, швидше за все, матиме прямий вплив на дохід та досвід користувачів. Більше того, використання ваших реальних клієнтів для експериментів буде повільнішим, ніж якщо б ви вже мали дані заздалегідь.

Автономні методи ідеально підходять для експериментальних етапів, оскільки користувач не бере безпосередньої участі, і на відміну від онлайн-методів, систему не потрібно розгортати. Дані поділяються на навчальні та перевірочні набори даних, що означає, що частина даних буде використана для побудови системи, а інша частина для її оцінки. Використовуючи ці методи, потрібно бути обережним, оскільки можуть бути фактори, що впливають на результати і не можуть бути адекватно представлені. Наприклад, фактор часу може бути дуже важливим у рекомендаціях (сезонність, погода тощо), як і настрої клієнта в певний момент часу.

1.5. Висновки до розділу

В першому розділі проведено аналіз можливості включення рекомендацій до онлайн-систем. З точки зору користувача, це збільшує досвід та створює зацікавленість. Для бізнесу це приносить більше доходу. Краще мати базову систему рекомендацій для невеликого набору користувачів і інвестувати в більш потужні техніки, коли база користувачів зростає.

Бізнес-цілі системи рекомендацій визначатимуть тип системи, на якій слід зосередитися спочатку: чи це стимулює більше зацікавленості для вже активних користувачів, чи штовхає тих нечастих клієнтів стати більш активними.

Окрім визначення бізнес-мети, важливо, щоб могли аналізувати та розуміти інформацію, що генерується на вашому сайті. Враховуючи це, ніщо не може завадити вам успішно впровадити вашу систему рекомендацій.

Основні положення, що було отримано з аналізу матеріалів даного розділу:

– алгоритми рекомендацій можна розділити на дві великі парадигми: спільні підходи (такі як користувач-користувач, позиція-елемент та факторизація матриці), які базуються лише на матриці взаємодії між елементом-користувачем та підходах на основі вмісту (наприклад, моделі регресії або класифікації), що попередня інформація про користувачів та / або предмети

– методи спільної роботи на основі пам'яті не передбачають жодної прихованої моделі і мають тоді низьку упередженість, але велику дисперсію; підходи, що базуються на моделях, передбачають приховану модель взаємодії, яка потребує вивчення уявлень як користувачів, так і предметів з нуля і, отже, має більший ухил, але меншу дисперсію; Методи, що базуються на вмісті, передбачають приховану модель побудови навколо користувачів та / або елементів елементів, явно наданих, і, таким чином, мають найбільший упередженість і найменшу дисперсію.

– рекомендаційні системи набувають все більшої важливості у багатьох великих галузях промисловості, і при розробці системи слід враховувати деякі міркування (краще використання розрідженості, ітеративні методи факторизації чи оптимізації, приблизні методи пошуку найближчих сусідів ...)

– системи, що рекомендують, важко оцінити: якщо можна використовувати деякі класичні показники, такі як MSE , точність, відкриття або точність, слід мати на увазі, що деякі бажані властивості, такі як різноманітність (випадковість) та пояснюваність, не можуть бути оцінені таким чином; оцінка реальних умов (наприклад, тестування A / B або тестування зразків), нарешті, єдиний реальний спосіб оцінки нової системи рекомендацій, але вимагає певної впевненості в моделі.

Слід зазначити, що не обговорювали гібридні підходи у цій вступній статті. Ці методи, що поєднують спільну фільтрацію та підходи на основі вмісту, у багатьох випадках досягають найсучасніших результатів і, отже, сьогодні використовуються у багатьох широкомасштабних системах рекомендацій. Поєднання, створене в гібридних підходах, може мати переважно дві форми: можемо або навчити дві моделі самостійно (одну модель спільної фільтрації та одну модель, що базується на вмісті) та об'єднати їх пропозиції, або безпосередньо побудувати єдину модель (часто нейронну мережу), яка об'єднує обидва підходи. використовуючи як вхідні дані попередню інформацію (про користувача та / або предмет), а також інформацію про взаємодію.

РОЗДІЛ 2
МЕТОДИ ПОБУДОВИ СИСТЕМИ ВИЗНАЧЕННЯ КОРИСТУВАЦЬКИХ
ПЕРЕВАГ

Вибухове зростання кількості доступної цифрової інформації та кількості відвідувачів Інтернету створило потенційну проблему перевантаження інформації, яка перешкоджає своєчасному доступу до предметів, що цікавлять Інтернет. Системи пошуку інформації, такі як Google, DevilFinder та Altavista, частково вирішили цю проблему, але визначення пріоритетів та персоналізації (де система відображає наявний вміст за інтересами та уподобаннями користувача) інформації відсутнє. Це збільшило попит на рекомендаційні системи як ніколи раніше. Системи, що рекомендують - це системи фільтрації інформації, які займаються проблемою перевантаження інформації [1] фільтруючи фрагмент життєво важливої інформації з великої кількості динамічно генерованої інформації відповідно до уподобань користувача, інтересу або спостережуваної поведінки щодо товару [1]. Система рекомендацій має можливість передбачити, чи віддасть певний користувач перевагу елементу чи ні, виходячи з профілю користувача.

Рекомендовані системи вигідні як постачальникам послуг, так і користувачам [1]. Вони зменшують трансакційні витрати на пошук та вибір предметів в Інтернет-магазинах [1]. Системи рекомендацій також довели, що покращують процес прийняття рішень та якість [1]. У налаштуваннях електронної комерції системи, що рекомендують, збільшують доходи, оскільки вони є ефективним засобом продажу більшої кількості товарів [1]. У наукових бібліотеках системи, що рекомендують, підтримують користувачів, дозволяючи їм вийти за межі пошуку за каталогами.

Кафедра КСУ				НАУ 21 02 36 000 ПЗ			
<i>Виконав</i>	<i>Воропай Д.В.</i>			Методи побудови системи визначення користувацьких переваг	<i>Літера</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Керівник</i>	<i>Апенько Н.В.</i>				<i>Д</i>	28	71
<i>Консульт.</i>					123 СП 437		
<i>Норм. контр.</i>	<i>Тупота Є.В.</i>						
<i>Зав. Каф.</i>	<i>Литвиненко О.Є.</i>						

Тому необхідність використання ефективних і точних методів рекомендацій у системі, яка надаватиме відповідні та надійні рекомендації для користувачів, не можна надмірно підкреслювати.

2.1. Аналіз існуючих підходів до визначення користувацьких переваг

Система збирає відповідну інформацію користувачів для створення профілю користувача або моделі для завдань прогнозування, включаючи атрибут користувача, поведінку або вміст ресурсів, до яких користувач отримує доступ. Агент рекомендацій не може функціонувати точно, поки профіль користувача / модель не будуть добре побудовані. Система повинна знати якомога більше від користувача, щоб з самого початку надати обґрунтовані рекомендації. Системи рекомендацій покладаються на різні типи введення, такі як найзручніший високоякісний явний зворотний зв'язок, який включає явне введення користувачами щодо їх інтересу до товару або неявний зворотний зв'язок шляхом опосередкованого виведення уподобань користувачів шляхом спостереження за поведінкою користувача[11]. Гібридний зворотний зв'язок також може бути отриманий комбінацією як явного, так і неявного зворотного зв'язку. На платформі електронного навчання профіль користувача - це сукупність особистої інформації, пов'язаної з конкретним користувачем. Ця інформація включає когнітивні навички, інтелектуальні здібності, стилі навчання, інтерес, уподобання та взаємодію з системою. Профіль користувача зазвичай використовується для отримання необхідної інформації для побудови моделі користувача. Таким чином, профіль користувача описує просту модель користувача. Успіх будь-якої системи рекомендацій багато в чому залежить від її здатності представляти поточні інтереси користувача. Точні моделі необхідні для отримання відповідних і точних рекомендацій з будь-яких методів прогнозування.

Зазвичай система пропонує користувачеві через системний інтерфейс надавати рейтинги для елементів для побудови та вдосконалення своєї моделі. Точність рекомендацій залежить від кількості оцінок, наданих користувачем.

Єдиним недоліком цього методу є те, що він вимагає зусиль від користувачів, а також користувачі не завжди готові надати достатньо інформації. Незважаючи на те, що явний зворотний зв'язок вимагає більше зусиль від користувача, він все одно розглядається як надання більш надійних даних, оскільки він не передбачає вилучення переваг із дій, а також забезпечує прозорість процесу рекомендацій, що призводить до дещо вищої якості сприйняття рекомендацій. і більше впевненості в рекомендаціях[12].

Система автоматично визначає уподобання користувача, відстежуючи різні дії користувачів, такі як історія покупок, історія навігації та час, проведений на деяких веб-сторінках, посилання, за якими переходить користувач, вміст електронної пошти та клацання кнопок серед інших. Неявний зворотний зв'язок зменшує навантаження на користувачів, роблячи висновки про переваги їх користувачів з їх поведінки в системі. Метод хоч і не вимагає зусиль від користувача, але він менш точний. Крім того, також стверджувалося, що неявні дані переваг насправді можуть бути більш об'єктивними, оскільки немає упередженості, що виникає в результаті реагування користувачів соціально бажаним способом[12] і не виникає проблем із самовідчуттям або необхідності підтримувати образ для інших[13].

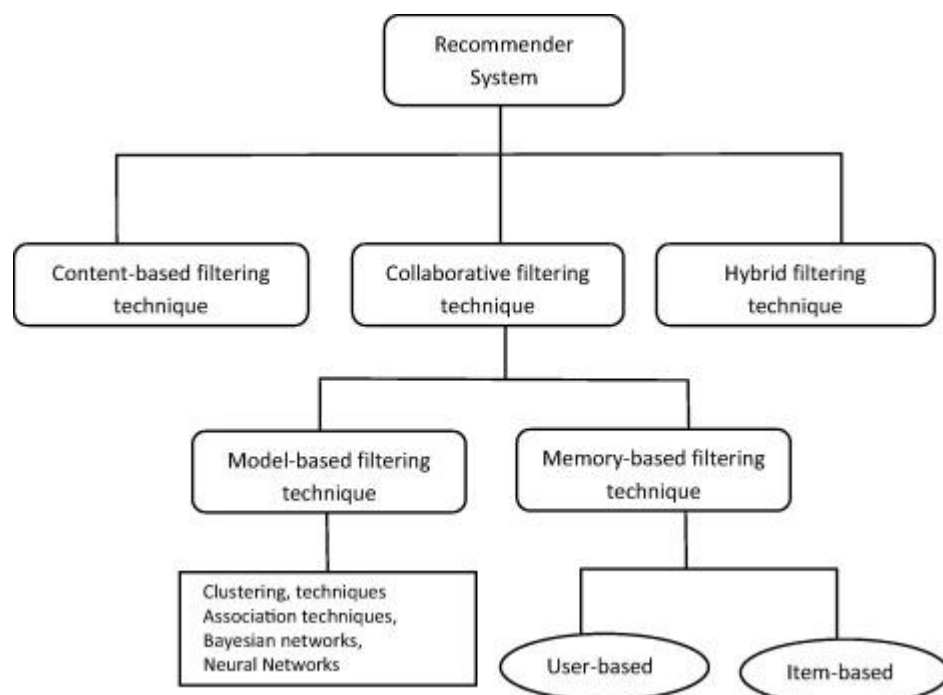


Рис. 2.1. Рекомендаційні етапи

Сильні сторони як неявного, так і явного зворотного зв'язку можуть поєднуватися в гібридній системі, щоб мінімізувати їх слабкі сторони та отримати найкращу систему. Цього можна досягти, використовуючи неявні дані як перевірку явного рейтингу або дозволяти користувачеві давати явний відгук лише тоді, коли він вирішить висловити явний інтерес.

Використання ефективних і точних методів рекомендацій дуже важливо для системи, яка надаватиме корисні та корисні рекомендації своїм окремим користувачам. Це пояснює важливість розуміння особливостей та потенціалів різних рекомендаційних методів. Рис.2 показано анатомію різних методик фільтрації рекомендацій.

2.2. Реалізація методу фільтрації на основі вмісту

Методика, що базується на вмісті, більше наголошує на аналізі атрибутів елементів, щоб генерувати прогнози. Коли такі документи, як веб-сторінки, публікації та новини, рекомендуються, метод фільтрації на основі вмісту є найбільш успішним. У техніці фільтрації на основі вмісту рекомендація надається на основі профілів користувачів з використанням функцій, вилучених із вмісту елементів, які користувач оцінював у минулому [14],[15]. Елементи, які здебільшого пов'язані з позитивно оціненими товарами, рекомендуються користувачеві. CBF використовує різні типи моделей, щоб знайти схожість між документами, щоб сформувані значущі рекомендації. Це може використовувати Вектор космічної моделі такі як Термінова частота, Зворотна частота документа (TF / IDF) або імовірнісні моделі, такі як Наївний Класифікатор Байєса [16], Дерева рішень [17] або Нейронні мережі [18] для моделювання взаємозв'язку між різними документами в корпусі. Ці методи дають рекомендації, вивчаючи базову модель за допомогою статистичного аналізу або техніки машинного навчання. Техніка фільтрації на основі вмісту не потребує профілю інших користувачів, оскільки вони не впливають на рекомендації. Крім того, якщо профіль користувача змінюється, техніка CBF все ще може коригувати свої рекомендації протягом дуже короткого періоду часу. Основним недоліком цієї

техніки є необхідність мати поглиблені знання та опис особливостей елементів у профілі.

Методи фільтрації СВ долають виклики CF. Вони можуть рекомендувати нові товари, навіть якщо немає оцінок, наданих користувачами. Отже, навіть якщо база даних не містить налаштувань користувача, точність рекомендацій не впливає. Крім того, якщо налаштування користувача змінюються, він має можливість коригувати свої рекомендації за короткий проміжок часу. Вони можуть керувати ситуаціями, коли різні користувачі не діляться однаковими елементами, а лише ідентичними предметами відповідно до своїх внутрішніх особливостей. Користувачі можуть отримувати рекомендації, не ділячись своїм профілем, і це забезпечує конфіденційність[19]. Методика CBF може також надати пояснення щодо того, як генеруються рекомендації для користувачів. Проте методики страждають від різних проблем, про які йдеться в літературі[12]. Методи фільтрації на основі вмісту залежать від метаданих елементів. Тобто вони вимагають детального опису предметів та дуже добре організованого профілю користувача, перш ніж можна буде робити рекомендації для користувачів. Це називається обмеженим аналізом вмісту. Отже, ефективність CBF залежить від наявності описових даних. Надмірна спеціалізація вмісту[10] є ще однією серйозною проблемою техніки CBF. Користувачі можуть отримувати рекомендації, подібні до елементів, уже визначених у їхніх профілях.

Терези [12]- це система рекомендацій щодо книг на основі вмісту, яка використовує інформацію про книги, зібрану з Інтернету. Він реалізує класифікатор Наїва Байєса на інформації, витягнутій з Інтернету, для вивчення профілю користувача для складання рейтингового списку назв на основі прикладів навчання, наданих окремим користувачем. Система може надавати пояснення щодо будь-яких рекомендацій, зроблених користувачам, перелічуючи функції, що сприяють найвищим рейтингам, а отже, дозволяючи користувачам мати повну впевненість у рекомендаціях, наданих користувачам системою.

2.3. Реалізація методу спільної фільтрації

Спільна фільтрація - це незалежна від домену техніка прогнозування вмісту, який неможливо легко та адекватно описати такими метаданими, як фільми та музика. Техніка спільної фільтрації працює шляхом побудови бази даних (матриця елементів користувача) переваг для елементів користувачами. Потім він відповідає користувачам з відповідними інтересами та уподобаннями, обчислюючи схожість між їхніми профілями, щоб робити рекомендації[13]. Такі користувачі будують групу під назвою сусідство. Користувач отримує рекомендації щодо тих елементів, які він раніше не оцінював, але які вже були позитивно оцінені користувачами в його сусідстві. Рекомендації, вироблені CF, можуть бути як передбаченнями, так і рекомендаціями. Прогнозування - це числове значення, R_{ij} , що виражає прогнозовану оцінку елемента j для користувача i , тоді як Рекомендація - це список найпопулярніших N елементів, які користувачеві найбільше сподобаються, як показано в Рис.3. Методика спільної фільтрації можна розділити на дві категорії: на основі пам'яті та на основі моделі[15],[14].

1. Завантажити: Завантажте зображення в повному розмірі

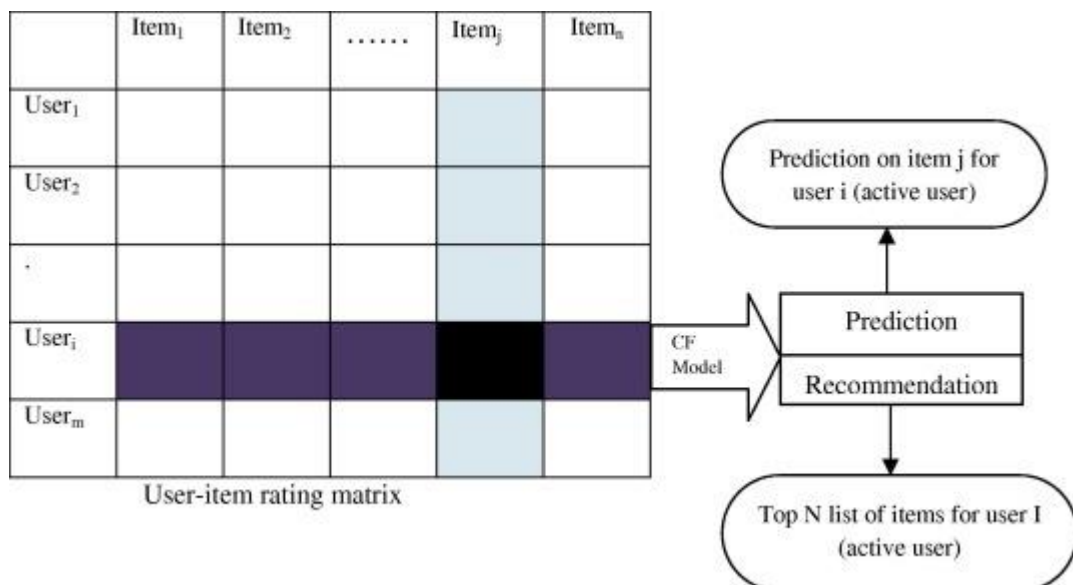


Рис. 2.3. Процес спільної фільтрації.

Елементи, які користувач уже оцінив раніше, відіграють важливу роль у пошуку сусіда, який поділяє з ним вдячність[15],[16]. Як тільки сусід користувача знайдений, різні алгоритми можуть бути використані для поєднання переваг сусідів для формування рекомендацій. Завдяки ефективності цих методів вони досягли широкого успіху в реальному житті. CF на основі пам'яті можна досягти двома способами за допомогою методів, що базуються на користувачах та на основі елементів. Техніка спільної фільтрації, заснована на користувачах, обчислює схожість між користувачами, порівнюючи їх оцінки за одним і тим же елементом, а потім обчислює прогнозовану оцінку для елемента активним користувачем як середньозважену оцінку елемента користувачами, подібну до активного користувача ваги - це схожість цих користувачів із цільовим елементом. Методи фільтрації на основі елементів обчислюють прогнози, використовуючи схожість між елементами, а не подібність між користувачами. Він створює модель схожості елементів, отримуючи всі елементи, оцінені активним користувачем, з матриці елементів користувача, визначає, наскільки схожі отримані елементи до цільового елемента, потім вибирає k найбільш подібних елементів та їх відповідні подібності. визначається. Прогнозування здійснюється шляхом взяття середньозваженого рейтингу активних користувачів за подібними елементами k . Для обчислення подібності між елементом / користувачем використовується кілька типів мір подібності. Дві найбільш популярні міри подібності - це кореляція та косинус. Прогнозування здійснюється шляхом взяття середньозваженого рейтингу активних користувачів за подібними елементами k . Для обчислення подібності між елементом / користувачем використовується кілька типів мір подібності. Дві найбільш популярні міри подібності - це кореляція та косинус. Прогнозування здійснюється шляхом взяття середньозваженого рейтингу активних користувачів за подібними елементами k . Для обчислення подібності між елементом / користувачем використовується кілька типів мір подібності. Дві найбільш популярні міри подібності - це кореляція та косинус.

Міра схожості також називається метрикою подібності, і це методи, що використовуються для обчислення балів, що виражають схожість користувачів

або предметів між собою. Потім ці оцінки можна використовувати як основу формування рекомендацій на основі користувачів чи елементів. Залежно від контексту використання, метрики подібності також можуть називатися метриками кореляції або метрики відстані[12].

Метод на основі моделі використовує попередні рейтинги для вивчення моделі з метою покращення ефективності техніки спільної фільтрації. Процес побудови моделі можна виконати за допомогою машинного навчання або методи обробки даних. Ці методи можуть швидко рекомендувати набір елементів, оскільки вони використовують заздалегідь обчислену модель, і вони довели, що вони дають результати рекомендацій, подібні до методів рекомендацій, що базуються на сусідських районах. Прикладами цих методів є техніка зменшення розмірності, така як Одиничне значення Розкладання (SVD), техніка завершення матриці, приховані семантичні методи та регресія та кластеризація. Методи, що базуються на моделях, аналізують матрицю елементів користувача для виявлення відносин між елементами; вони використовують ці відносини для порівняння списку найкращих рекомендацій. Методи, засновані на моделі, вирішують розрідженість проблеми, пов'язані з системами рекомендацій.

Застосування алгоритмів навчання також змінило спосіб рекомендацій: від рекомендацій, що споживати користувачам, до рекомендацій, коли насправді споживати продукт. Тому дуже важливо вивчити інші алгоритми навчання, що використовуються на основі моделей рекомендаційних систем:

Правило асоціації: Алгоритми майнінгу правил асоціації[19]правила вилучення, які передбачають появу предмета на основі присутності інших предметів у транзакції. Наприклад, з огляду на набір транзакцій, де кожна транзакція є набором елементів, a_n правило асоціації застосовує формулу $A \rightarrow B$, де A і B - це два набори елементів[10]. Правила асоціації можуть сформулювати дуже компактне представлення даних про переваги, що може підвищити ефективність зберігання, а також продуктивність.

Кластеризація: Методи кластеризації застосовуються в різних сферах, таких як розпізнавання зразків, обробка зображень, статистичний аналіз даних та виявлення знань [11]. Алгоритм кластеризації намагається розділити набір даних

на набір підкластерів, щоб виявити значущі групи, що існують у них[12]. Після формування кластерів думки інших користувачів кластеру можуть бути усереднені та використані для вироблення рекомендацій для окремих користувачів. Добрий метод кластеризації дасть кластери високої якості, у яких схожість між кластерами висока, тоді як подібність між кластерами низька. У деяких підходах кластеризації користувач може частково брати участь у різних кластерах, і тоді рекомендації базуються на середньому показнику кластерів участі, який зважується за ступенем участі[13]. К-засоби та Самоорганізуюча карта (SOM) є найбільш часто використовуваними серед різних методів кластеризації. К-означає приймає вхідний параметр, а потім розділяє набір з n елементів на

Дерево рішень: Дерево рішень базується на методології дерева графіки який побудований шляхом аналізу набору навчальних прикладів, для яких відомі позначення класів. Потім вони застосовуються для класифікації небачених раніше прикладів. Якщо вони навчаються на дуже якісних даних, вони можуть робити дуже точні прогнози[16]. Дерева рішень є більш зрозумілими, ніж інші класифікатори, такі як Підтримка машини вектор(SVM) та нейронних мереж, оскільки вони зрозуміло поєднують прості питання щодо даних. Дерева рішень також гнучкі в обробці предметів із поєднанням справжніх та категоріальних ознак, а також предметів, які мають деякі специфічні відсутні функції.

Штучна нейронна мережа: ANN - це структура багатьох з'єднаних нейронів (вузлів), які систематично розташовані шарами. Зв'язки між нейронами мають ваги, пов'язані з ними, залежно від ступеня впливу одного нейрона на інший. Є деякі переваги використання нейронних мереж у деяких особливих проблемних ситуаціях. Наприклад, через те, що вона містить багато нейронів, а також присвоює вагу кожному з'єднанню, штучна нейронна мережа є досить надійною щодо шумних і помилкових наборів даних[17]. ANN має здатність оцінювати нелінійні функції та фіксувати складні взаємозв'язки в наборах даних, вони також можуть бути ефективними і навіть функціонувати, якщо частина мережі виходить з ладу. Основним недоліком є те, що важко знайти ідеальну

топологию мережі для даної проблеми, і як тільки топологія буде прийнята, це буде діяти як нижня межа для класифікаційна помилка.

Аналіз посилань: Аналіз посилань - це процес побудови мереж взаємопов'язаних об'єктів з метою вивчення закономірностей та тенденцій[18]. Це дало великий потенціал для вдосконалення веб-пошуку. Аналіз посилань складається з алгоритмів PageRank та HITS. Більшість алгоритмів аналізу посилань обробляють веб-сторінку як єдиний вузол у веб-графік[19].

Регресія: Регресійний аналіз застосовується, коли вважається, що дві або більше змінних систематично пов'язані лінійною залежністю. Це потужний і різноманітний процес для аналізу асоціативних зв'язків між залежною змінною та однією або кількома незалежними змінними. Використання регресії містить підгонку кривої, прогнозування та тестування систематичних гіпотез про взаємозв'язок між змінними. Крива може бути корисною для виявлення тенденції в наборі даних, будь вона лінійною, параболічною чи якоїсь іншої форми.

Баєсові класифікатори: Вони є імовірнісні рамки для вирішення класифікаційних задач, що базується на визначенні умовної ймовірності теорема Байєса. Байєсівські класифікатори[16] розглядають кожен атрибут та мітку класу як випадкові величини.

Основні переваги класифікаторів Naive Bayes полягають у тому, що вони стійкі до ізольованих точок шуму та нерелевантних атрибутів, і вони обробляють відсутні значення, ігноруючи екземпляр під час розрахунку оцінки ймовірності. Однак припущення про незалежність може не виконуватись для деяких атрибутів, оскільки вони можуть корелювати. У цьому випадку звичайним підходом є використання Байєсових мережі. Баєсові класифікатори можуть виявитись практичними для середовищ, в яких знання про переваги користувачів змінюються повільно щодо часу, необхідного для побудови моделі, але не підходять для середовищ, в яких моделі переваг користувачів повинні оновлюватися швидко або часто. Він також має успіх у системах рекомендацій на основі моделей, оскільки часто використовується для отримання моделі для систем рекомендацій на основі вмісту.

Суть техніки заповнення матриці полягає у передбаченні невідомих значень у матрицях елемента користувачького елемента. K-найближчий сусід на основі кореляції є одним з основних методів, що застосовуються в системах рекомендацій щодо спільної фільтрації[10]. Вони значною мірою залежать від історичних даних про рейтинг користувачів щодо предметів. Найчастіше рейтингова матриця завжди дуже велика і розріджена через те, що користувачі не оцінюють більшість елементів, представлених у матриці[11]. Ця проблема завжди призводить до нездатності системи давати надійні та точні рекомендації користувачам. На практиці використовуються різні варіації моделей низького рангу для заповнення матриці, особливо щодо застосування в спільній фільтрації [12].

2.4. Реалізація методу гібридної фільтрації

Техніка гібридної фільтрації поєднує різні методи рекомендацій, щоб отримати кращу оптимізацію системи, щоб уникнути деяких обмежень та проблем чистих систем рекомендацій[14],[15]. Ідея гібридних методів полягає в тому, що поєднання алгоритмів надаватиме більш точні та ефективні рекомендації, ніж один алгоритм, оскільки недоліки одного алгоритму можуть бути подолані іншим алгоритмом[15]. Використання декількох рекомендаційних методів може придушити слабкі сторони окремої техніки в комбінованій моделі. Поєднання підходів можна здійснити будь-яким із наступних способів: окрема реалізація алгоритмів та комбінування результату, використання певної фільтрації на основі вмісту в спільному підході, використання певної спільної фільтрації у підході на основі вмісту, створення єдиної системи рекомендацій, яка забезпечує разом обидва підходи.

Зважена гібридизація поєднує результати різних рекомедаторів, щоб сформувати список рекомендацій або прогнозування, інтегруючи бали з кожної із застосовуваних методик за лінійною формулою. Прикладом зваженої гібридизованої системи рекомендацій є P-танго[16]. Система складається з рекомендацій на основі вмісту та спільної роботи. Спочатку їм дають рівні ваги,

але ваги коригуються у міру підтвердження прогнозів чи іншим чином. Перевага зваженого гібрида полягає в тому, що всі переваги системи, що рекомендує, використовуються під час процесу рекомендацій прямолінійно.

Система переходить на один із методів рекомендацій відповідно до евристики, що відображає здатність рекомендатора скласти хороший рейтинг. Комутаційний гібрид має можливість уникнути проблем, характерних для одного методу, наприклад, нової проблеми користувача контент-рекомендатора, переключившись на спільну систему рекомендацій. Перевага цієї стратегії полягає в тому, що система чутлива до сильних і слабких сторін тих, хто рекомендує. Основним недоліком комутаційних гібридів є те, що він, як правило, вносить більше складності в процес рекомендацій, оскільки критерій перемикання, який зазвичай збільшує кількість параметрів в системі рекомендацій, повинен бути визначений[14]. Прикладом перемикання гібридних рекомендацій є DailyLearner[17] який використовує як гібрид на основі вмісту, так і спільний гібрид, коли рекомендація на основі вмісту застосовується спочатку перед рекомендацією щодо спільної роботи в ситуації, коли система, що базується на вмісті, не може давати рекомендації з достатньою кількістю доказів.

Техніка каскадної гібридизації застосовує ітеративний процес доопрацювання при побудові порядку переваги серед різних предметів. Рекомендації однієї техніки уточнені іншою методикою рекомендацій. Перший метод рекомендацій видає грубий список рекомендацій, який, у свою чергу, уточнюється наступним методом рекомендацій. Техніка гібридизації дуже ефективна і толерантна до шуму завдяки грубому до тоншого характеру ітерації. EntreeC[14] є прикладом методу каскадної гібридизації, який використовував каскадний заснований на знаннях та спільний рекомендатор.

Змішані гібриди поєднують результати рекомендацій різних технік рекомендацій одночасно, замість того, щоб мати лише одну рекомендацію на товар. Кожен елемент має кілька рекомендацій, пов'язаних із різними методами рекомендацій. При змішаній гібридизації окремі показники не завжди впливають на загальну ефективність місцевого регіону. Прикладом рекомендованої системи

в цій категорії, яка використовує змішану гібридизацію, є система PTV[18], які рекомендують графік перегляду телевізора для користувача, поєднуючи рекомендації з контенту та спільні системи формування графіків. Profinder[19] та PickAFlick[10] є також прикладами змішаних гібридних систем.

Характеристики, вироблені конкретною методикою рекомендацій, додаються до іншої методики рекомендації. Наприклад, рейтинг подібних користувачів, який є особливістю спільної фільтрації, використовується вміркування на основі справирекомендаційна техніка як одна з ознак для визначення подібності між предметами. Ripper - приклад техніки поєднання функцій, яка використовувала рейтинги спільного фільтра в системі, що базується на вмісті, як функцію для рекомендації фільмів[11]. Перевага цієї методики полягає в тому, що вона не завжди покладається виключно на спільні дані.

Методика використовує рейтинги та іншу інформацію, вироблену попереднім рекомендаторм, а також вимагає додаткових функціональних можливостей систем, що рекомендують. Наприклад, система Терезів[12]робить рекомендації на основі вмісту книг щодо даних, знайдених на Amazon.com, використовуючи найвний класифікатор тексту Байєса. Гібриди, що збільшують особливості, перевершують методи комбінування особливостей, оскільки вони додають невелику кількість функцій до основного рекоменлятора.

Внутрішня модель, сформована за допомогою однієї методики рекомендацій, використовується як вхідна інформація для іншої. Створена модель завжди багатша інформацією у порівнянні з одним рейтингом. Мета-рівень[17]гібриди здатні вирішити проблему розрідженості спільних методів фільтрації, використовуючи всю модель, засвоєну першим методом, як вхідні дані для другого методу. Прикладом техніки метарівню є праця[12]який використовує миттєве навчання для створення контенту на основі профілю користувача, який потім порівнюється спільно.

2.5. Оціночні показники для рекомендаційних алгоритмів

Якість алгоритмів рекомендацій можуть бути оцінені за допомогою різних типів вимірювань, які можуть бути точністю або охопленням. Тип використовуваних показників залежить від типу техніки фільтрування. Точність - це частка правильних рекомендацій від загальної кількості можливих рекомендацій, тоді як охоплення вимірює частку об'єктів у просторі пошуку, для яких система може надати рекомендації. Метрики для вимірювання точності систем фільтрації рекомендацій поділяються на статистичні та метрики точності підтримки прийняття рішень[13]. Придатність кожного показника залежить від особливостей набору даних та типу завдань, які система рекомендацій зробилю[16].

Статистичні показники точності оцінюють точність техніки фільтрування шляхом порівняння прогнозованих оцінок безпосередньо з фактичним рейтингом користувача. Середня абсолютна помилка (MAE) [14], Середньоквадратична помилка (RMSE) та кореляція зазвичай використовуються як метри статистичної точності. MAE - найпопулярніший і найпоширеніший; це міра відхилення рекомендації від конкретного значення користувача. Вона обчислюється наступним чином[16]

Коренева середньоквадратична помилка (RMSE) робить більший акцент на більшій абсолютній похибці, і чим нижча RMSE, тим краща точність рекомендацій.

Популярними показниками точності підтримки прийняття рішень є частота сторнування, зважені помилки, робочі характеристики приймача (ROC) та крива відкликання точності (PRC), точність, відкликання та F-міра. Ці показники допомагають користувачам вибрати елементи, які мають дуже високу якість із доступного набору елементів [16]. Метрики розглядають процедуру передбачення як двійкова операція, що відрізняє хороші предмети від тих, що не є хорошими. Криві ROC дуже успішні при виконанні комплексної оцінки продуктивності деяких конкретних алгоритмів. Точність - це частка рекомендованих елементів, яка насправді актуальна для користувача, тоді як

відкриття може бути визначена як частка відповідних елементів, яка також є частиною набору рекомендованих елементів[17].

Покриття пов'язане з відсотком товарів та користувачів, які система рекомендацій може надати прогнози. Прогнозування може бути практично неможливим, якщо жоден користувач або декілька користувачів не оцінили товар. Покриття можна зменшити, визначивши невеликі розміри району для користувача чи предметів [18].

2.5. Висновки до розділу

Системи рекомендацій відкрили нові можливості отримання персоніфікованої інформації в Інтернеті. Це також допомагає полегшити проблему перевантаження інформації, що є дуже поширеним явищем інформаційно-пошукові системи і надає користувачам доступ до продуктів та послуг, які не є доступними для користувачів у системі. У цій роботі було обговорено дві традиційні методики рекомендацій та висвітлено їх сильні сторони та проблеми за допомогою різноманітних видів стратегій гібридизації, що використовуються для поліпшення їх ефективності. Різні алгоритми навчання, що використовуються при формуванні рекомендаційних моделей та метрики оцінки використовуються при вимірюванні якості та продуктивності алгоритми рекомендацій були обговорені. Ці знання розширяють можливості дослідників та слугуватимуть дорожньою картою для вдосконалення сучасних методик рекомендацій.

РОЗДІЛ 3
ПРОЕКТУВАННЯ НЕЙРОННОЇ МЕРЕЖІ ВИЗНАЧЕННЯ
КОРИСТУВАЦЬКИХ ПЕРЕВАГ

3.1. Етапи машинного експерименту

Для практичних випробувань розробленого методу рекомендацій на основі візерункових структур (*RAPS*), було прийнято рішення скористатися вільно поширюваними даними оцінок книг *GoodBook4U* (*GoodBook4U.umn.edu*). Збір даних проходив в рамках проекту *The GroupLens Research Project* Університету Міннесоти.

В обраних даних представлено 100000 оцінок по 1682 книгам від 943 різних користувачів. Причому кожен користувач в цих даних поставив оцінки не менше, ніж 20 книгам.

При проектуванні моделі деяких даних ми повинні зробити припущення щодо процесу, який породив дані. Насправді ми можемо сказати, що модель є набір припущень і набір припущень є модель. Взаємозв'язок між моделлю і припущенням, які він представляє, настільки важливі, що варто підкреслити: Модель = Набір припущень щодо даних

Вибираючи, які припущення включити до моделі найважливішою частиною моделі дизайн. Неправильні припущення призведуть до моделі, яка дають неточні прогнози через ці помилкові припущення. При проектуванні афакторного графіку, починаємо з вибору змінних, які ми хочемо мати на графіку. Принаймні, графік повинен містити змінні, що представляють дані, які ми насправді маємо (чи правильно кандидат обрав кожне запитання), та будь-які змінні, про які ми хочемо дізнатися (навички).

Кафедра КСУ				НАУ 21 02 36 000 ПЗ			
<i>Виконав</i>	<i>Воропай Д.В.</i>			Проектування нейронної мережі визначення користувачьких переваг	<i>Літера</i>	<i>Аркуш</i>	<i>Аркуші</i>
<i>Керівник</i>	<i>Апенько Н.В.</i>				<i>Д</i>	43	71
<i>Консульт.</i>					123 СП 437		
<i>Норм. контр.</i>	<i>Тупота С.В.</i>						
<i>Зав. Каф.</i>	<i>Литвиненко О.Є.</i>						

Часто корисно вводити інші, проміжні змінні. Вибравши змінні, ми можемо почати додавати фактори на наш графік, щоб кодувати, як ці змінні впливають одна на одну в процесі відповіді на запитання. Зазвичай корисно починати зі змінних, про які ми хочемо дізнатися (навички), і проробляти процес, щоб закінчити зі змінними, які ми можемо насправді виміряти (чи правильно кандидат визначив питання).

Отже, починаючи зі змінних навичок, ось наше перше припущення:

1. Кандидат або освоїв кожну навичку, або ні.

Значення 1 означає, що можемо представити вміння кандидата як двійкову (істинну / хибну) змінну, що є правдою якщо кандидат оволодів навиком та помилкою якщо вони цього не зробили. Викликаються змінні, які можуть приймати одне з фіксованого набору значень (як і всі змінні, які ми бачили до цього часу) дискретні змінні. Далі в цьому розділі ми зустрінемося з безперервними змінними, які можуть приймати будь-яке значення у безперервному діапазоні значень, наприклад будь-яке дійсне число від 0 до 1. Як ми побачимо, безперервні змінні корисні для вивчення ймовірності подій, серед багатьох інших застосувань.

Далі нам потрібно зробити припущення про попередню ймовірність кандидата, що має кожну з цих навичок.

2. Перед тим, як побачити будь-які результати тестування, однаково ймовірно, що кандидат має або не має якихось особливих навичок.

Значення 2 означає, що попередня ймовірність для кожної змінної навички слід встановити нейтрально 50%, (рис. 3.1), що становить коефіцієнт Бернуллі (0,5).

Продовжуючи процес відповіді на запитання, ми зараз повинні зробити деякі припущення про те, як відповіді тестових відповідей кандидата стосуються їхніх навичок.

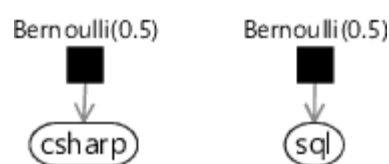


Рис. 3.1. Факторний графік для двійкових змінних навичок

Припустимо, що у них є всі навички для того чи іншого питання, ми все одно повинні допустити, що вони іноді можуть помилятися. Якщо ми задали деякі запитання щодо SQL експерту з SQL, скільки слід очікувати, що вони отримають право? Можливо, не всі з них, але, можливо, вони отримали б 90% або близько того правильно. Ми могли б перевірити це припущення, попросивши деяких справжніх експертів зробити таку вікторину і побачивши, які бали вони отримують, але наразі ми будемо вважати, що помилятися з кожним десятим неправильно:

3. Якщо кандидат має усі навички, необхідні для питання, тоді вони, як правило, правильно поставлять запитання, за винятком того, що раз на десять вони помиляться.

Щодо питань, де у кандидата не вистачає необхідних навичок, ми можемо припустити, що вони навмання здогадуються:

4. Якщо кандидат не має усіх навичок, необхідних для питання, він вибере відповідь навмання. Оскільки це іспит із кількома варіантами з п'ятьма відповідями, існує кожен п'ятий шанс, що вони правильно поставлять запитання.

Таблиця умовних ймовірностей (P) показує ймовірність кожного значення $isCorrect1$ зумовлене кожним із двох значень $sharp$.

Ми зателефонуємо до факторів, представляючи цю умовну таблицю ймовірностей $AddNoise$ оскільки вихід є "галасливою" версією вводу. Оскільки наші припущення стосуються однаково всіх навичок, ми можемо використовувати однакові фактори $isCorrect2$. Це дає наступний факторний графік для перших двох питань:

A	B	C=false	C=true
false	false	1.000	0.000
false	true	1.000	0.000
true	false	1.000	0.000
true	true	0.000	1.000

Рис. 3.2. Таблиця для перших двох питань тесту

При будівництві модель часто хочемо включити змінну, яка є фіксованою функцією деяких інших змінних у моделі. Наприклад, ми можемо хотіти, щоб двійкова змінна була істинною, якщо всі інші двійкові змінні є істинними (І) або якщо будь-яка з них є істинною (АБО). Для безперервної змінної, ми можемо хотіти, щоб це була сума чи добуток якогось іншого безперервної змінної.

Можемо досягти цього, поставивши детермінований фактор в нашому факторний графік. умовна ймовірність розподіл для детермінований фактор завжди має значення або 1, або 0. Це 1, якщо дочірня змінна дорівнює шуканій функції батьківські змінні 0 - інакше. Наприклад, якщо ми хочемо додати змінну C, який має дорівнювати AІВ, можемо додати детермінований фактор чия умовний розподіл ймовірностей є:

Ось частковий факторний графік показуючи, як фактор може бути використаний для створення має Навички потрібна змінна:

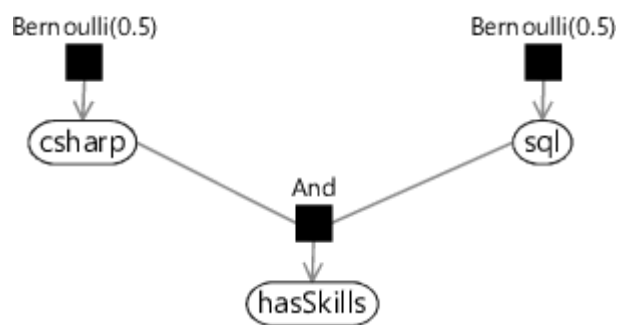


Рис. 3.3. І фактор є детермінований фактор, що стримує навички

Тепер ми можемо зібрати все, щоб побудувати факторний графік з усіх трьох питань. Нам просто потрібно підключитися має Навички до нашого `isCorrect3` змінної, ще раз використовуючи `AddNoise` фактор:

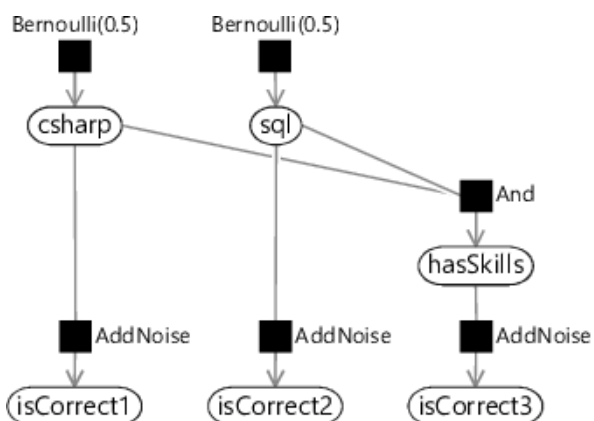


Рис. 3.4. Факторний графік для трьох питань із множинним вибором

Спільний розподіл ймовірностей для цього факторний графік досить довгий, тому що зараз їх загалом шість факторні вузли, що означає, що він містить шість термінів

Тому що спільні розподіли ймовірностей як цей, великі і незручні для роботи, як правило, простіший у використанні факторні графіки як більш читабельний та керований спосіб висловити а модель.

Важливо, щоб будь-який модель містить змінні, що відповідають спостережуваним даним, і що ці змінні одного типу. Це дозволяє приєднувати дані до моделі фіксуючи ці змінні до відповідних спостережуваних значень даних. А нумовивідТоді обчислення можна використовувати для пошуку граничних розподілів для будь-якої іншої (неспостережуваної) змінної в модель. Для наших модель, нам потрібно переконатись, що ми можемо приєднати дані про результати тесту до моделі, що складається з результату так / ні для кожного питання залежно від того, чи правильно кандидат визначив це питання. Ми дійсно можемо долучити ці дані до наших модель, оскільки ми маємо двійкові змінні (`isCorrect1`, `isCorrect2`, `isCorrect3`), який ми можемо встановити правд якщо кандидат правильно зрозумів питання і помилковий інакше.

У цьому є ще одне припущення модель про що ще не згадувалося. Насправді це, як правило, одне з найбільших припущень, зроблених будь-ким модель! Це передбачається сфера застосування модель: тобто припущення, що лише змінні, включені в модель актуальні. Наприклад, наш модель не згадує про психічний стан кандидата (втомленого, напруженого), а також про умови, в яких вони виконували тест, або про те, чи могло відбуватися шахрайство, чи кандидат навіть розуміє мову, на якій виникають запитання написано в. Виключивши ці змінні з нашого модель, ми твердо припустили, що вони є незалежний з (не впливають) відповідей кандидата.

Погані припущення щодо обсягу часто призводять до незадовільних результатів умовивід такий процес, як зниження точності прогнозування. Сфера застосування а модель- це припущення, яке слід критично оцінити під час модель процес проектування, хоча б лише для виявлення аспектів проблеми,

які ігноруються. Тож, щоб бути ясним, останнє припущення щодо наших навичок навчання модельє:

5. Чи правильно кандидат отримує питання, залежить лише від того, якими навичками цей кандидат володіє, а не від нічого ще.

Не будемо чітко називати це припущення в майбутньому моделі, але гарною практикою є ретельно розглянути, які змінні ігноруються, щоразу, коли ви проектуєте або використовуєте модель.

A	B	C=false	C=true
false	false	1.000	0.000
false	true	1.000	0.000
true	false	1.000	0.000
true	true	0.000	1.000

Рис. 3.5. П'ять припущень, закодованих у нашому моделі.

Дуже важливо ретельно переглянути всі припущення щодо моделювання, щоб переконатися, що вони обґрунтовані. Спрощення припущень можна робити, якщо пам'ятаєте, що це може знизити точність результатів. Значення 2 здається очевидно безпечним, оскільки це просто припущення про незнання. Однак також передбачається, що кожна зі змінних навичок є такою незалежний, тобто знання того, що хтось має якусь певну навичку, нічого не говорить вам про те, чи є у них якісь інші навички. Якщо деякі навички якимось чином пов'язані, це цілком може бути не так.

3.2. Випробування моделі

Побудувавши а модель, перше, що потрібно зробити, це перевірити його на простих прикладах даних, щоб переконатися, що його поведінка є розумною. Припустимо, кандидат знає C #, але не SQL - ми очікували б, що вони правильно зрозуміли перше запитання, а інші два - неправильно. Тож давайте перевіримо модель подивіться на цей випадок і подивіться, які навички він здобуває для такої моделі відповідей. Для зручності ми будемо

використовувати правильно посилається на масив [isCorrect1, isCorrect2, isCorrect3] і тому ми розглянемо випадок де правильноє [правда, помилковий, помилковий].

Робимо умови від обчислення вручну займають багато часу, і легко помилитися. Натомість ми можемо зробити той самий розрахунок механічно, використовуючи алгоритм передачі повідомлень. Це працює, передаючи повідомлення по крайх факторний графік, де повідомлення - це розподіл ймовірностей над змінною, до якої підключено ребро. Ми побачимо, що використання алгоритм передачі повідомлень дозволяє нам робити умови від розрахунки автоматично - величезна перевага модельна основі підходу!

У цьому додатковому розділі ми покажемо, яку умови від можна виконати за допомогою алгоритм передачі повідомлень називається поширенням переконань. Якщо хочете зосередитись на моделюванні, сміливо пропускайте цей розділ.

Переробимо умови для розрахунку csharp-вміння користуватися передачею повідомлень - спочатку ми опишемо процес передачі повідомлення для цього прикладу, а потім розглянемо загальну форму пізніше. Першим кроком у ручному розрахунку було закріплення значень спостережуваних змінних. Використовуючи передачу повідомлення, це відповідає кожній спостережуваній змінній, що надсилає повідомлення, яке є точковим розподілом маси за спостережуваним значенням. У нашому випадку кожен правильно змінна посилає точкова маса Бернуллі (0), якщо це спостерігається помилковий або точкова маса Бернуллі (1), якщо це спостерігається правда.

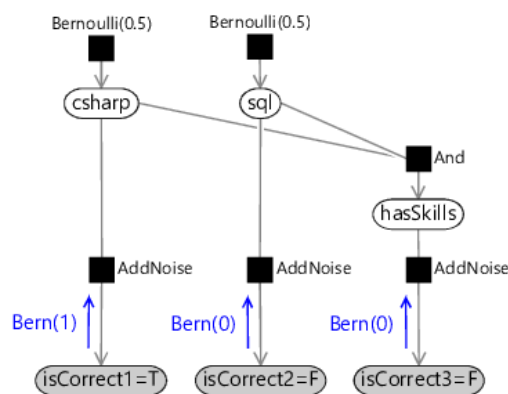


Рис. 3.6. Повідомлення, надіслані спостережуваним вузлам змінних

Ціточкова маса повідомлення надходить на AddNoise-факторні вузли. Вихідні повідомлення на кожному факторній вузол можна обчислити окремо наступним чином:

- Повідомлення з першого AddNoise фактору csharp можна обчислити як $P(\text{isCorrect1} = T | \text{csharp})$. Розподіл Бернуллі більше csharp. параметром Бернуллі є $p = 0,90,9 + 0,2 = 0,818$, отже, вихідне повідомлення є Бернуллі (0,818) Бернуллі (0,818).

- Повідомлення з другого AddNoise фактору можна обчислити як $P(\text{isCorrect2} = F | \text{sql})$. Розподіл Бернуллі більше sql. Параметр Бернуллі становить $p = 0,10,1 + 0,8 = 0,111$, отже, вихідне повідомлення є Бернуллі (0,111) Бернуллі (0,111).

- Повідомлення з третього AddNoise фактору має Навички- те саме, що друге повідомлення, оскільки воно обчислюється для того самого фактора з тим самим вхідним повідомленням. Отже, є і третє зверху повідомлення Бернуллі (0,111) Бернуллі (0,111).

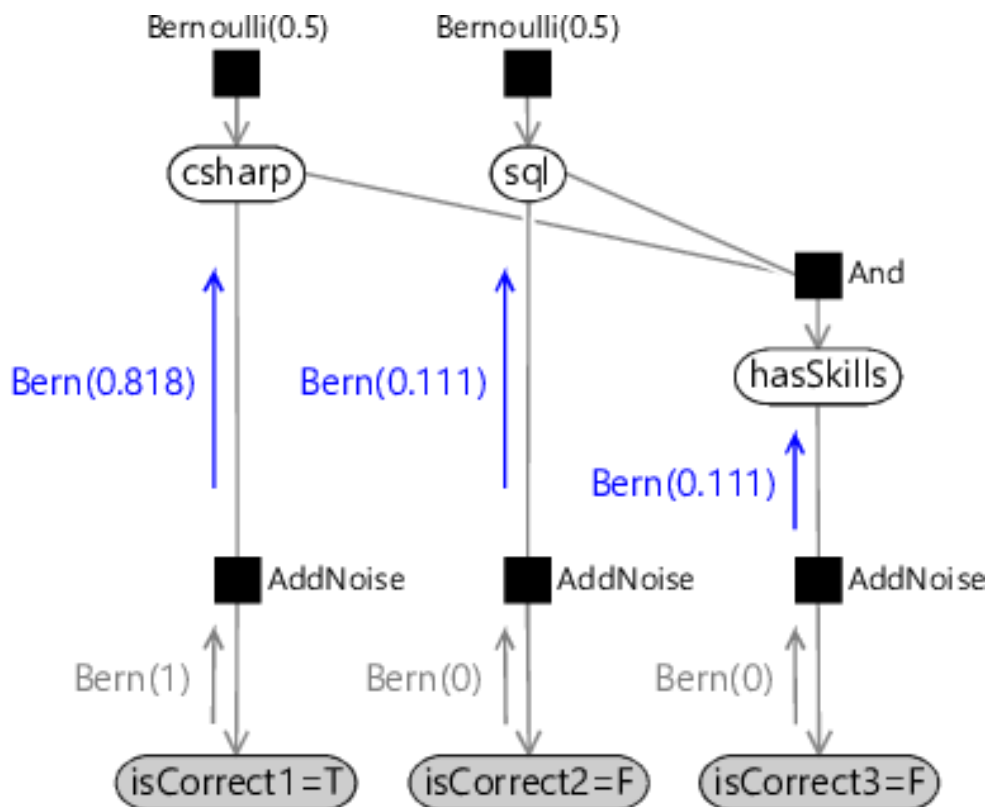


Рис. 3.7. Вихідні повідомлення з AddNoise факторні вузли

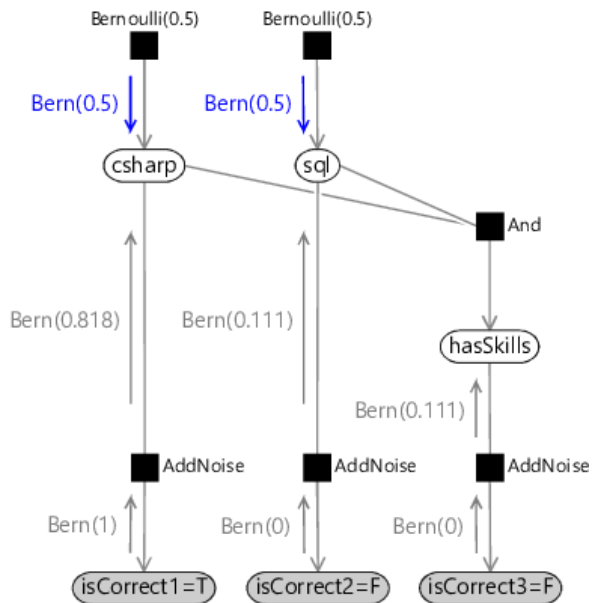


Рис. 3.7 Повідомлення відБернулліпріорфакторні вузли

Вихідне повідомлення для будь-якого змінного вузол є добутком вхідних повідомлень на інших краях, підключених до цього вузла. Дляквзмінний вузолтепер у нас є вхідні повідомлення з двох країв, що означає, що ми можемо обчислити вихідне повідомлення в напрямкуІфактор. Це Бернуллі (0,111), оскільки висхідне повідомлення залишається незмінним шляхом множення на рівномірне низхідне повідомлення Бернуллі (0,5). маєНавичкизмінний вузолце простіше: оскільки вхідне повідомлення лише одне, вихідне повідомлення є лише його копією.

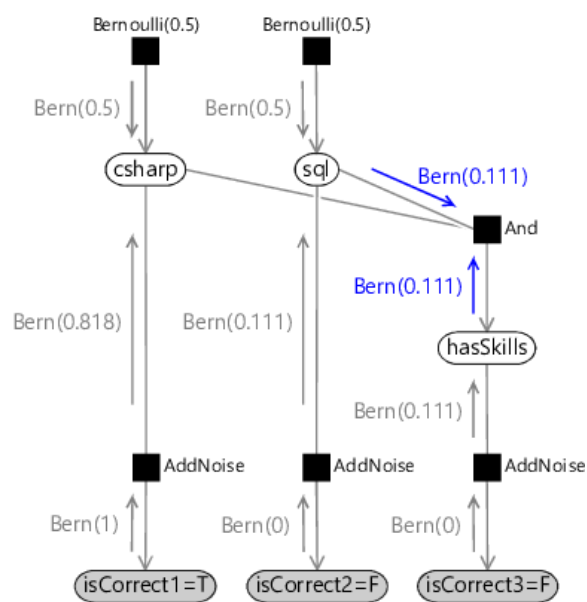


Рис. 3.8. Повідомлення Навички, які мають вузли змінних

Нарешті, ми можемо обчислити вихідне повідомлення з Іфактор до csharp змінна. Це обчислюється множенням вхідних повідомлень на факторфункція та підсумовування за всіма змінними, крім тієї, до якої надсилається.

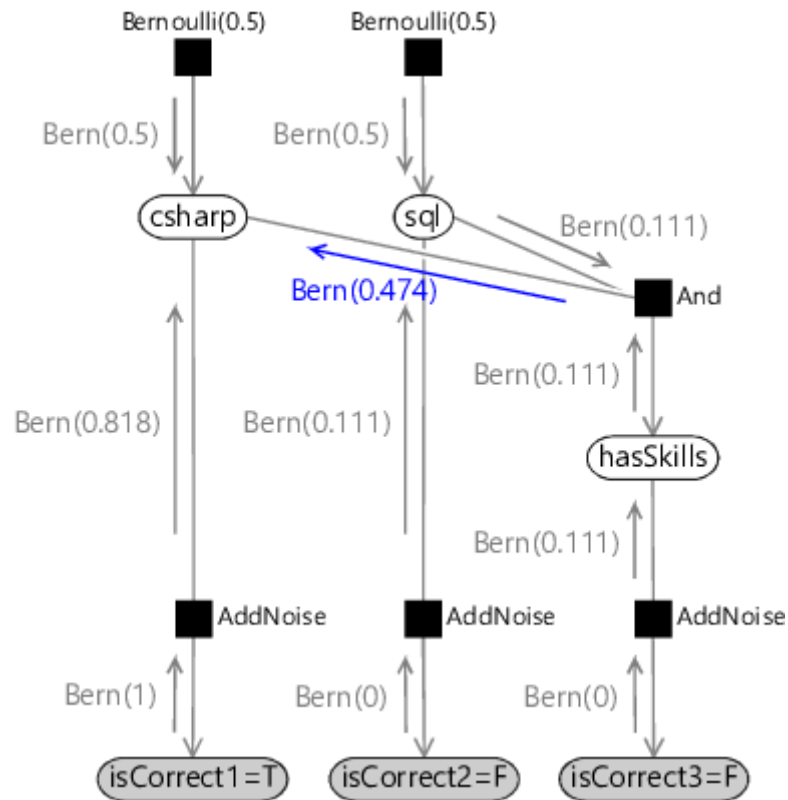


Рис. 3.9. Заключне повідомлення до csharp змінний вузол

Тепер є всі три вхідні повідомлення на csharp змінний вузол, що означає, що ми готові обчислити його задній граничний. Це досягається множенням трьох повідомлень - це розрахунок, який провели у і, отже, дає той самий результат Бернуллі (0,802) або 80,2%.

Для обчислення граничного для kv, ми можемо повторно використовувати більшість повідомлень, які ми щойно розрахували, і тому нам потрібно обчислити лише два додаткові повідомлення. Перше повідомлення від csharp до Іфактор, є добутком Бернуллі (0,818) та рівномірний розподіл Бернуллі (0,5), отже результат також Бернуллі (0,818).

Процедура передачі повідомлень, яку ми щойно побачили, виникає внаслідок застосування алгоритму телефонувати поширення переконань [Перлина, 1988; Lauritzen and Spiegelhalter, 1988]. В поширення переконань, повідомлення обчислюються одним із трьох способів, залежно від того, надходить

повідомлення зфакторний вузол, спостерігається змінний вузол або неспостережений змінний вузол.

3.3. Точність і повнота використання Програмних засобів

Для порівнювання методів рекомендацій *Slope One* і *RAPS* спочатку було прийнято рішення скористатися стандартними заходами точності (*precision*) і повноти (*recall*):

$$recall = \frac{\left| \left\{ \begin{array}{l} \text{relevant} \\ \text{books} \end{array} \right\} \cap \left\{ \begin{array}{l} \text{retrieved} \\ \text{books} \end{array} \right\} \right|}{\left| \left\{ \begin{array}{l} \text{retrieved} \\ \text{books} \end{array} \right\} \right|} \quad (3.1)$$

$$recall = \frac{\left| \left\{ \begin{array}{l} \text{relevant} \\ \text{books} \end{array} \right\} \cap \left\{ \begin{array}{l} \text{retrieved} \\ \text{books} \end{array} \right\} \right|}{\left| \left\{ \begin{array}{l} \text{relevant} \\ \text{books} \end{array} \right\} \right|} \quad (3.2)$$

Але з часом було прийнято рішення скориговану точність і повноту розраховувати за формулами зазначеним нижче:

$$recall = \frac{\left| \left\{ \begin{array}{l} \text{relevant} \\ \text{books} \end{array} \right\} \cap \left\{ \begin{array}{l} \text{retrieved} \\ \text{books} \end{array} \right\} \cap \left\{ \begin{array}{l} \text{test} \\ \text{books} \end{array} \right\} \right|}{\left| \left\{ \begin{array}{l} \text{retrieved} \\ \text{books} \end{array} \right\} \cap \left\{ \begin{array}{l} \text{test} \\ \text{books} \end{array} \right\} \right|} \quad (3.3)$$

$$recall = \frac{\left| \left\{ \begin{array}{l} \text{relevant} \\ \text{books} \end{array} \right\} \cap \left\{ \begin{array}{l} \text{retrieved} \\ \text{books} \end{array} \right\} \cap \left\{ \begin{array}{l} \text{test} \\ \text{books} \end{array} \right\} \right|}{\left| \left\{ \begin{array}{l} \text{relevant} \\ \text{books} \end{array} \right\} \cap \left\{ \begin{array}{l} \text{test} \\ \text{books} \end{array} \right\} \right|} \quad (3.4)$$

Такі оцінки точності і повноти дозволили уникнути описану вище проблему невизначеності, так як не знаємо як саме користувач оцінить

рекомендована книга. У реальному житті б могли запитати користувача, чи вірна рекомендація, але в нашому випадку цього дозволити не можемо. Іншими словами, при оцінці точності і повноти даних методом вважаємо, що в даний конкретний момент, для заданого користувача, музичних композицій крім *train GoodBook* і *test GoodBook* просто не існує.

Було проведено три групи тестів:

- гарна оцінка з інтервалу [1; 5];
- хороша оцінка в інтервалі [1; 5];
- гарна оцінка книги для рекомендації [1; 5].

Середні значення результатів вийшли такими (табл. 3.1).

Таблиця 3.1

Результати експериментів

	Середній час,	Середня точність,%	Середня повнота
<i>RAPS</i> [1,5]	3,62	19,42	50,52
<i>Slope One</i> [1,5]	18,37	1,57	23,41
<i>RAPS</i> [1,5]	18,23	55,62	63,33
<i>Slope One</i> [1,5]	18,90	53,99	30,39
<i>RAPS</i> [1,5]	32,98	80,11	83,65
<i>Slope One</i> [1,5]	18,51	83,81	81,88

Головна сторінка містить шапку сайту, слайдер зображень, панель пошуку по фільтрам і пошуку рекомендацій і каталог (рис. 3.10).

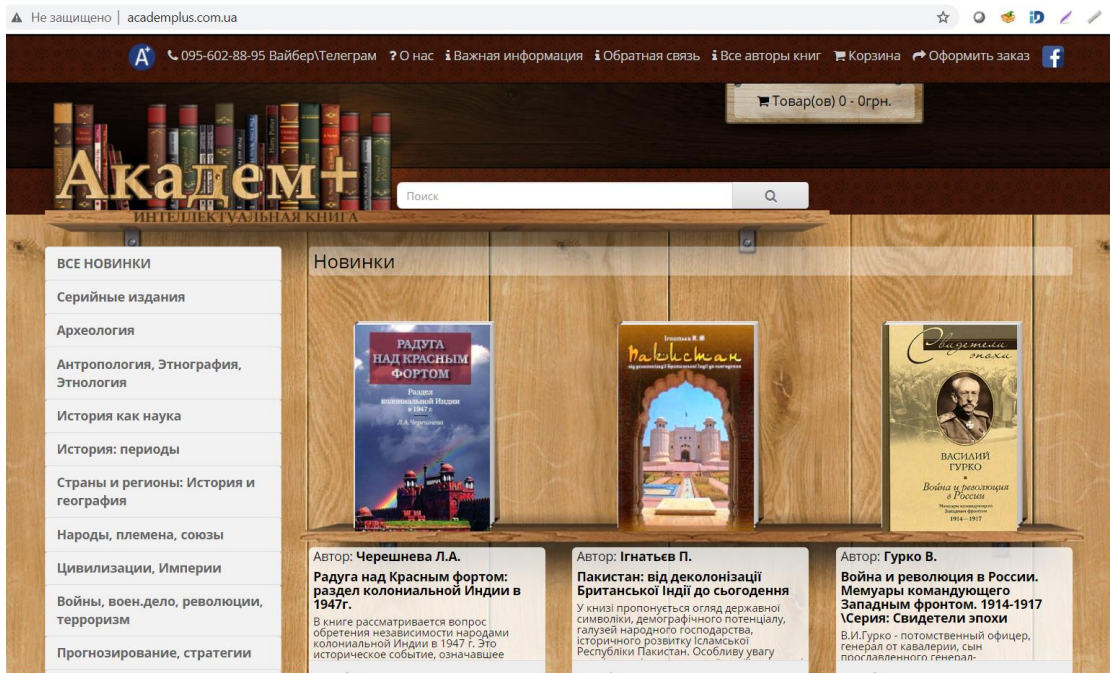


Рис. 3.10. Головна сторінка сайту

Пошук за фільтрами знаходить книги певних тематик і років випуску, а також новинки. Каталог тематик складається з усіх книг, що містяться на сайті, які представлені у тематиці (рис. 3.11).

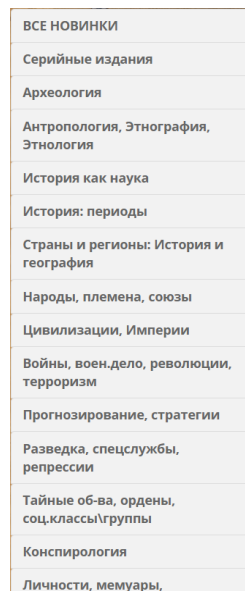


Рис. 3.11. Меню з тематикам книг

Тематики розташовані у вигляді стовпців і рядків (рис. 3.12). Кількість рядків може бути необмежено і залежить від кількості тематик, наявних в базі даних. При натисканні на будь-яку тематику користувач переходить на присвячену йому сторінку.

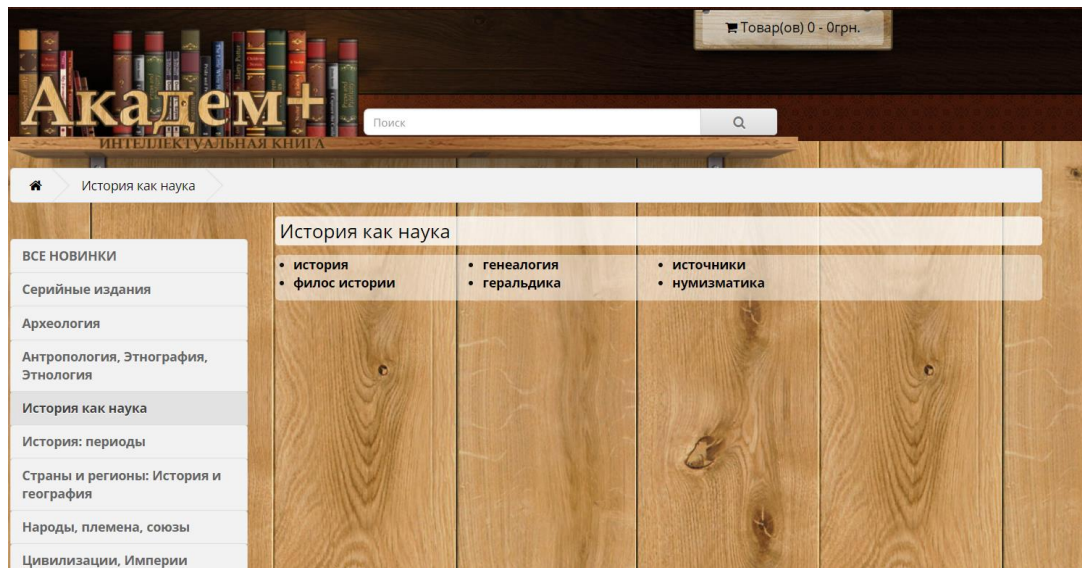


Рис. 3.12. Горизонтальне розташування вкладених тематик

На сторінці тематики користувач може отримати інформацію про авторів, книгу і, якщо користувач авторизований, поставити йому оцінку від одного до п'яти (рис. 3.4).

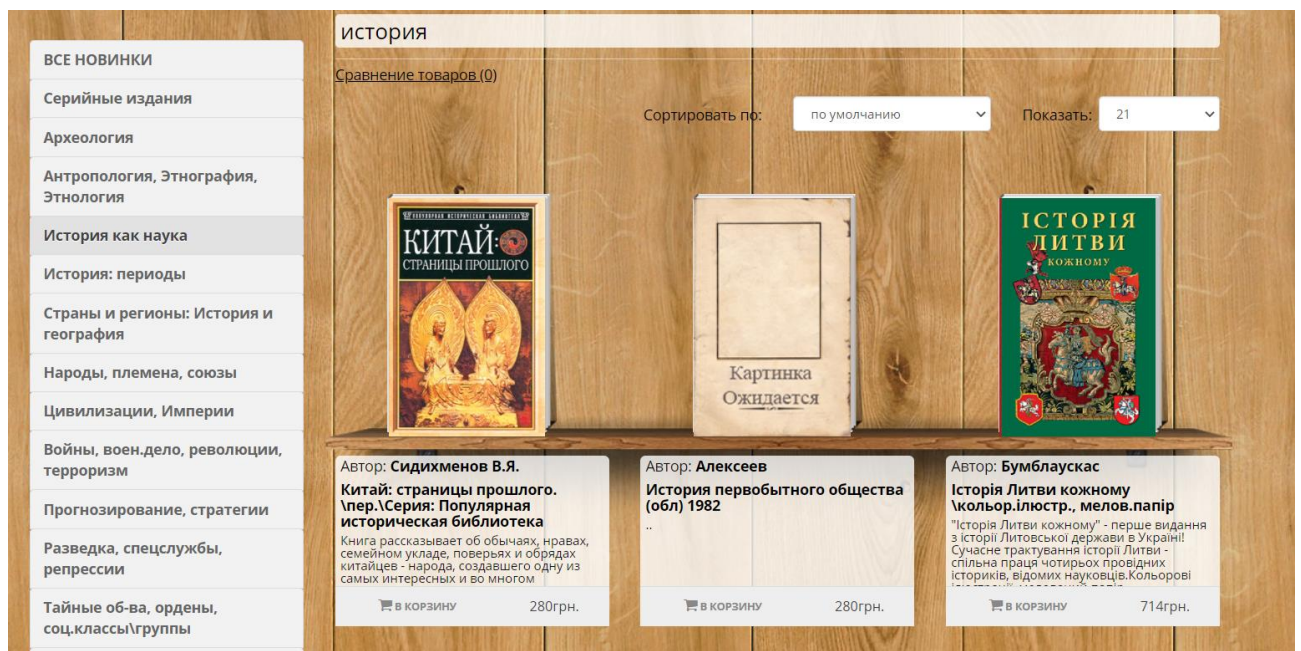


Рис. 3.13. Сторінка тематики

На рисунку 3.14 представлений приклад фільтрації тематики «цезар».

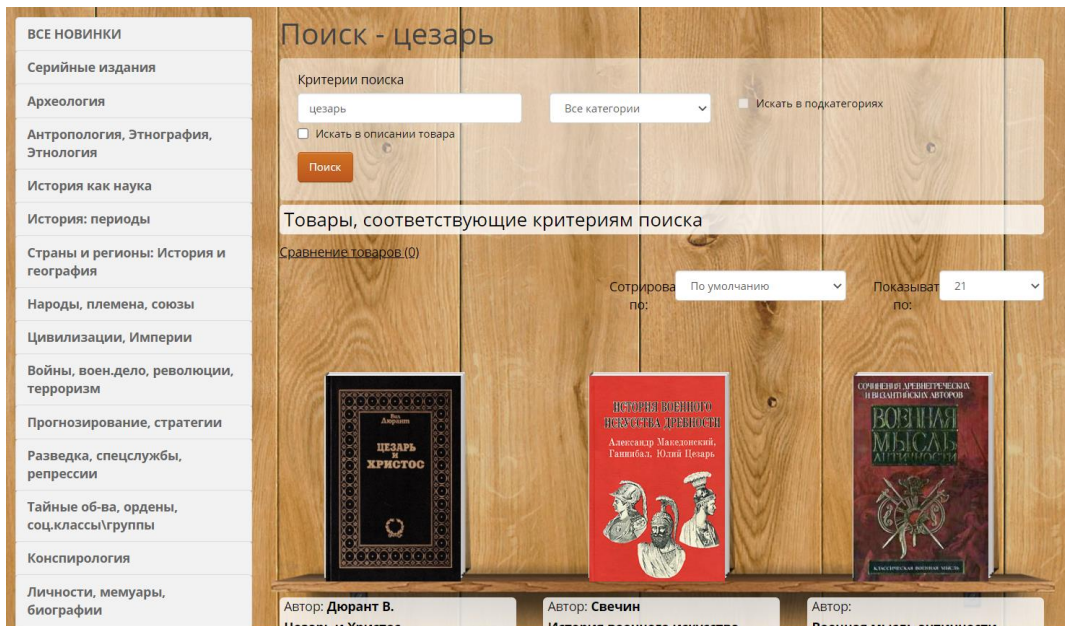


Рис. 3.14. Фільтрація в межах тематики

Якщо користувач авторизований, він може скористатися рекомендаційною системою, для чого необхідно натиснути кнопку «Рекомендації», після чого в області відображення всіх книг, з'являться ті книги, які ще не оцінив відвідувач, крім того, вони будуть розташовані в порядку убавання: від більш цікавих до менш (рис. 3.9).

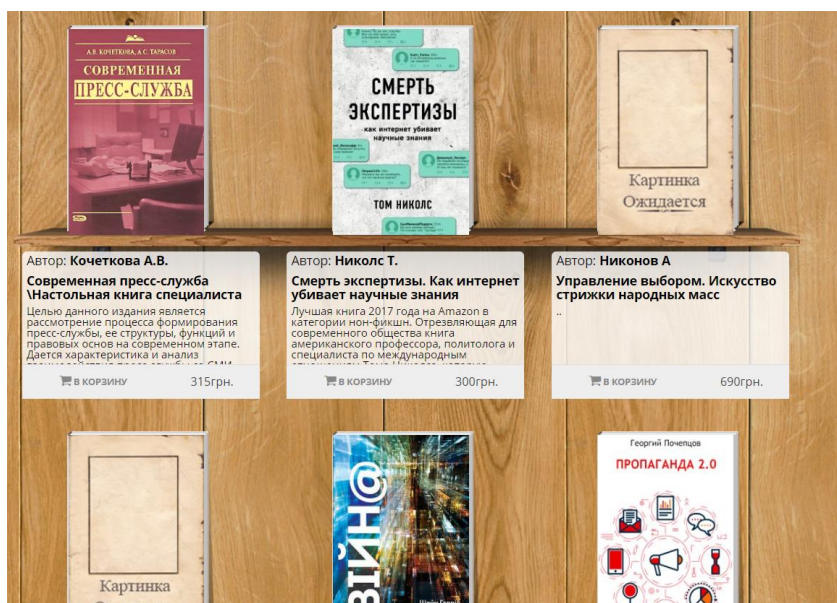


Рис. 3.15. Список рекомендованих книг

3.4. Висновки до розділу

У цьому розділі були описані дані, на яких проводились експерименти, а також наведено заходи точності і повноти для оцінки роботи алгоритмів.

Коли повністю протестували нашу модель на прикладах даних було опитано 22 добровольців пройти оціночний тест, що складається з 48 питань, призначених для оцінки семи різних тематик книг. Окрім проходження тесту, також попросили кожного волонтера сказати, якими тематиками вони цікавляться.

Реалізовані алгоритми було впроваджено на сайті книжок, де авторизовані користувачі можуть скористатися рекомендаційною системою, для чого необхідно натиснути кнопку «Рекомендації», після чого в області відображення всіх книг, з'являться ті книги, які ще не оцінив відвідувач, крім того, вони будуть розташовані в порядку убутання: від більш цікавих до менш цікавих.

ВИСНОВКИ

Системи рекомендацій відкрили нові можливості отримання персоніфікованої інформації в Інтернеті. Це також допомагає полегшити проблему перевантаження інформації, що є дуже поширеним явищем інформаційно-пошукові системи і надає користувачам доступ до продуктів та послуг, які не є доступними для користувачів у системі. У цій роботі було обговорено дві традиційні методики рекомендацій та висвітлено їх сильні сторони та проблеми за допомогою різноманітних видів стратегій гібридизації, що використовуються для поліпшення їх ефективності. Різні алгоритми навчання, що використовуються при формуванні рекомендаційних моделей та метрики оцінки використовується при вимірюванні якості та продуктивності алгоритми рекомендацій були обговорені. Ці знання розширяють можливості дослідників та слугуватимуть дорожньою картою для вдосконалення сучасних методик рекомендацій.

Рекомендаційні системи служать одним з найбільш популярних і монетизованих прикладів використання рішень на базі штучного інтелекту і машинного навчання. За рахунок актуальних рекомендацій клієнтам в потрібному місці, потрібний час і через релевантні канали вони допомагають бізнесу збільшувати конверсію.

В першому розділі проведено аналіз можливості включення рекомендацій до онлайн-систем. З точки зору користувача, це збільшує досвід та створює зацікавленість. Для бізнесу це приносить більше доходу. Краще мати базову систему рекомендацій для невеликого набору користувачів і інвестувати в більш потужні техніки, коли база користувачів зростає.

Бізнес-цілі системи рекомендацій визначатимуть тип системи, на якій слід зосередитися спочатку: чи це стимулює більше зацікавленості для вже активних користувачів, чи штовхає тих нечастих клієнтів стати більш активними.

Окрім визначення бізнес-мети, важливо, щоб могли аналізувати та розуміти інформацію, що генерується на вашому сайті. Враховуючи це, ніщо не може завадити вам успішно впровадити вашу систему рекомендацій.

Основні положення, що було отримано з аналізу матеріалів даного розділу:

– алгоритми рекомендацій можна розділити на дві великі парадигми: спільні підходи (такі як користувач-користувач, позиція-елемент та факторизація матриці), які базуються лише на матриці взаємодії між елементом-користувачем та підходах на основі вмісту (наприклад, моделі регресії або класифікації), що попередня інформація про користувачів та / або предмети

– методи спільної роботи на основі пам'яті не передбачають жодної прихованої моделі і мають тоді низьку упередженість, але велику дисперсію; підходи, що базуються на моделях, передбачають приховану модель взаємодії, яка потребує вивчення уявлень як користувачів, так і предметів з нуля і, отже, має більший ухил, але меншу дисперсію; Методи, що базуються на вмісті, передбачають приховану модель побудови навколо користувачів та / або елементів елементів, явно наданих, і, таким чином, мають найбільший упередженість і найменшу дисперсію

– рекомендаційні системи набувають все більшої важливості у багатьох великих галузях промисловості, і при розробці системи слід враховувати деякі міркування (краще використання розрідженості, ітеративні методи факторизації чи оптимізації, приблизні методи пошуку найближчих сусідів ...)

– системи, що рекомендують, важко оцінити: якщо можна використовувати деякі класичні показники, такі як *MSE*, точність, відкриття або точність, слід мати на увазі, що деякі бажані властивості, такі як різноманітність (випадковість) та пояснюваність, не можуть бути оцінені таким чином; оцінка реальних умов (наприклад, тестування *A / B* або тестування зразків), нарешті, єдиний реальний спосіб оцінки нової системи рекомендацій, але вимагає певної впевненості в моделі

Слід зазначити, що не обговорювали гібридні підходи у цій вступній статті. Ці методи, що поєднують спільну фільтрацію та підходи на основі вмісту, у багатьох випадках досягають найсучасніших результатів і, отже, сьогодні

використовуються у багатьох широкомасштабних системах рекомендацій. Поєднання, створене в гібридних підходах, може мати переважно дві форми: можемо або навчити дві моделі самостійно (одну модель спільної фільтрації та одну модель, що базується на вмісті) та об'єднати їх пропозиції, або безпосередньо побудувати єдину модель (часто нейронну мережу), яка об'єднує обидва підходи. використовуючи як вхідні дані попередню інформацію (про користувача та / або предмет), а також інформацію про взаємодію.

У цьому розділі були описані дані, на яких проводились експерименти, а також наведено заходи точності і повноти для оцінки роботи алгоритмів.

Коли повністю протестували нашу модель на прикладах даних було опитано 22 добровольців пройти оціночний тест, що складається з 48 питань, призначених для оцінки семи різних тематик книг. Окрім проходження тесту, також попросили кожного волонтера сказати, якими тематиками вони цікавляться.

Реалізовані алгоритми було впроваджено на сайті книжок, де авторизовані користувачі можуть скористатися рекомендаційною системою, для чого необхідно натиснути кнопку «Рекомендації», після чого в області відображення всіх книг, з'являться ті книги, які ще не оцінив відвідувач, крім того, вони будуть розташовані в порядку убавання: від більш цікавих до менш цікавих.

На розробленому сайті, де було розгорнуто системи користувач регулярно поповнює свою книжкову полицю, при цьому економлячи час і отримуючи незвичайний досвід. Вибір книги стає сюрпризом: система підбирає твори, аналізуючи рецензії провідних критиків, - в посилку потрапляють новинки видавництва і бестселери.

Щоб вирішити завдання збільшення продажів при скороченні витрат і підвищити лояльність користувачів було автоматизовано підбір книг і персоналізовано пропозицію, при цьому зберігши критерій "сюрпризу".

Таким чином реалізовано рекомендаційна система на базі ML, яка дозволяє на підставі інтересів читача виділяти з обширної бази книг топ-25 творів.

Користувач заповнює на сайті анкету, в якій вказує, що цікавлять його жанри, улюблені твори і авторів. Можна вказати один або декілька жанрів: художній, класика, фентезі, саморозвиток, нон-фікшн.

На основі цих даних система формує список з 25 рекомендованих книг, сортуючи їх за ймовірністю збігу з інтересами читача. У той же час наявність таких рекомендацій багаторазово підвищує ймовірність догодити читачеві.

Алгоритм машинного навчання аналізує більше 8000 україномовних і російськомовних книг з відкритих каталогів. Відсутні дані і читацькі рейтинги "підтягуються" з сайту Good Reads за унікальним ідентифікатором книги ISBN.

Розроблена система має великий потенціал для розвитку. Її можна "навчити":

- обробляти серії книг;
- враховувати книги, що не сподобалися;
- спростити формат введення даних;
- зводити користувачів зі схожими літературними уподобаннями;
- розподіляти рекомендовані книги за рівнем закупівельних цін, що дозволило б сайту зменшити собівартість посилки.

СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТОК А

Лістинг коду основного модулю програми побудови нейронної мережі