НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Навчально-науковий комплекс "Інститут прикладного системного аналізу"
(повна назва інституту/факультету)

Кафедра Системного проектування
(повна назва кафедри)

«На правах рукопису»
УДК _004.891.2_____

«До захисту допущено»

Завідувач кафедри
_____ А.І. Петренко
(підпис) (ініціали, прізвище)

“___”_____20__ р.

Магістерська дисертація

зі спеціальності _____8.05010103 Системне проектування_____
(код і назва спеціальності)

на тему:___Data mining як засіб структурування великих обсягів даних (Big Data) та їх обробки

Виконав: студент _6_ курсу, групи ДА-32м
(шифр групи)

_____Кислий Роман Володимирович___ _____
(прізвище, ім’я, по батькові) (підпис)

Науковий керівник __д.т.н., професор, Петренко А. І._____ _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант Охорона праці _к.б.н., доц. Гусєв А.М._____ _____
(назва розділу) (науковий ступінь, вчене звання, , прізвище, ініціали) (підпис)

Рецензент _____
_____ _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2015 року

Національний технічний університет України
«Київський політехнічний інститут»

Факультет (інститут) ___ННК “Інститут прикладного системного аналізу”_____
(повна назва)

Кафедра _____Системного проектування_____
(повна назва)

Освітньо-кваліфікаційний рівень _____«Магістр»_____
(назва ОКР)

Напрям підготовки _____6.050101 Комп’ютерні науки_____
(код і назва)

Спеціальність _____8.05010103 Системне проектування_____
(код і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ А.І.Петренко
(підпис) (ініціали, прізвище)

«___» _____2015 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

_____Кислому Роману Володимировичу_____
(прізвище, ім’я, по батькові)

1. Тема дисертації: Data mining як засіб структурування великих обсягів даних (Big Data) та їх обробки

науковий керівник дисертації Петренко Анатолій Іванович, д.т.н., проф. ,
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «__» _____ 2015 р. №_____

2. Строк подання студентом дисертації 08.06.2015

3. Об’єкт дослідження: структурування даних з соціальних мереж і їх обробка для підвищення точності рекомендаційних систем

4. Предмет дослідження: створення рекомендаційної системи з додатковим джерелом даних на вході — профілем користувача з соціальних мереж

5. Перелік завдань, які потрібно розробити:

1. Провести огляд існуючих методів надання рекомендацій
2. Розглянути соціальні мережі, як джерела додаткових даних для рекомендаційних систем

3. Розробити методи використання соціальних мереж в якості джерел даних для рекомендацій
4. Побудувати рекомендаційну систему, з використанням додаткових джерел даних (соціальних мереж) і проаналізувати її ефективність

6. Орієнтовний перелік ілюстративного матеріалу: презентація по темі «Data mining як засіб структурування великих обсягів даних (Big Data) та їх обробки»

7. Орієнтовний перелік публікацій

Кислий Р.В. Моделювання профілю користувача для уникнення проблеми холодного старту в рекомендаційних системах / Кислий Р. В. // Системний аналіз та інформаційні технології: материалы 17-й Міжнародної науково-технічної конференції SAIT 2015, Київ, 22-25 червня 2015 р. – 2015. – с.244

Кислий Р. В. Колаборативна фільтрація в рекомендаційних системах на основі даних з сщціальних мереж / Кислий Р. В. // Матеріали III-ої міжнародної науково-практичної конференції Обчислювальний інтелект (результати, проблеми, перспективи), Київ-Черкаси, 12-15 травня 2015р. - 2015. - с.208

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	доц., к.б.н. Гусєв А. М		
Основна частина	проф., д.т.н. Петренко А. І.		

9. Дата видачі завдання 30.09.2014

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	Отримання завдання на дипломну роботу	30.09.2014	
2	Огляд літератури за темою роботи	3.11.2014	
3	Проведення огляду існуючих методів надання рекомендацій	22.01.2015	
4	Розгляд соціальних мереж, як джерела даних для рекомендаційних систем	5.02.2015	
5	Розробка методів використання соціальних мереж для рекомендацій	18.02.2015	
6	Побудова рекомендаційної системи, з використанням додаткових джерел даних (соціальних мереж)	1.03.2015	
7	Аналіз та порівняння результатів рекомендації	19.04.2015	
8	Розробка розділу з охорони праці	10.05.2015	

9	Оформлення дипломної роботи	31.05.2015	
10	Отримання допуску до захисту та подача роботи в ДЕК	10.06.2015	

Студент _____ Р.В. Кислий
(підпис) (ініціали, прізвище)

Керівник роботи _____ А. І. Петренко
(підпис) (ініціали, прізвище)

РЕФЕРАТ

Магістерська дисертація: 93 стор. загального тексту, 21 рисунка, 9 таблиць, 43 використаних джерела.

З зростанням кількості інформації, зростає і можливість її аналізу. Структурування та обробка великих обсягів даних (big data) відкриває нові можливості в їх аналізі та знаходженні прихованих знань.

Наприклад, структурування інформації з соціальних мереж дозволяє більш точно зробити рекомендації користувачу, враховуючи не тільки традиційні джерела інформації про нього (як його історію дій, оцінювання), а й такі, як місце розташування користувача, його поведінку в соціальних мережах, тощо.

Об'єкт дослідження — покращення релевантгості рекомендацій в онлайн рекомендаційних системах.

Предмет дослідження – створення рекомендаційної системи з додатковим джерелом даних на вході — профілем користувача з соціальних мереж.

Теоретико-методологічною основою роботи є техніки інтелектуального аналізу даних для рекомендацій, статистичного аналізу, колаборативна фільтрація.

Метою роботи є дослідження використання соціальних мереж як додаткового джерела даних, що передбачає підвищення ефективності надання рекомендацій, а також, часткового подолання проблеми “холодного старту”.

За результатами виконання магістерської дисертації було подано тези на 17-ту міжнародну конференцію SAIT 2015 та 3-тю міжнародну науково-практичну конференцію COMINT-2015. Тези було надруковано в збірниках тез відповідних конференцій.

ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ, РЕКОМЕНДАЦІЙНА СИСТЕМА, ВЕЛИКІ ДАНІ, СОЦІАЛЬНІ МЕРЕЖІ, СТРУКТУРУВАННЯ ДАНИХ

ABSTRACT

Master’s thesis consists of 93 pages of general text, 21 graphics, 9 tables, 43 sources.

With the rising amount of information in the world, growing an opportunity to analyze it. Structuring and processing big data opens new possibilities in its analysis and finding hidden knowledge.

For example, structuring information from social networks can help to make recommendations more accurately, based not only on traditional sources of information about users (like its history of actions, evaluation), but such as location, users behavior in social networks.

The object of study - improvement recommendations accuracy of on-line recommendation engines.

Subject of research – creation of the recommendation system with an additional source of data on the input - user profile on social networks.

Theoretical and methodological basis of the work is data mining technologies for recommendations, statistical analysis, collaborative filtering.

The aim is to study the use of social networks as an additional source of data, providing efficiency recommendations, as well as partial overcoming the “cold start” problem.

As the results of this work, abstracts were submitted for the 17 th International Conference SAIT 2015 and 3rd international scientific conference COMINT-2015. Abstracts were published in collections of abstracts of relevant conferences.

DATA MINING, RECOMMENDATION SYSTEMS, BIG DATA, SOCIAL NETWORKS, DATA STRUCTURING

Зміст

1. ВСТУП

Безперервне збільшення кількості даних забезпечує величезні можливості для людства. Управління великими даними (big data), безперечно, буде відігравати важливу роль у майбутньому, знаходячи застосування в різних галузях. Є так багато потенційно надзвичайно корисних ідей, які приховані у великих даних. Різні методи аналізу, такі як: прогнозування, аналіз текстів, семантичний аналіз, створюють конкурентну перевагу за рахунок обробки даних з різними рівнями складності, швидкості і точності, які були раніше недоступні.

Нові методи отримання інформації, платіжні системи, інтернет - генерують щодня величезний потік інформації. Крім того, кожен автор (індивідуальна людина, організація, бізнес) є виробником нової неструктурованої інформації: персональних даних, географічно локалізованих даних, інформації в соціальних мережах, фотографій, блогів і т.д. Також, відбувається генерація великої кількості структурованих даних, таких як: інтернет речей, дані з різних датчиків, зондів, RFID карток, тощо.

Big Data дає можливість інтеграції, синхронізації, управління та оцінки цього потоку дуже різноманітної інформації. Мається на увазі, що скориставшись кількістю даних можна проаналізувати їх суть і значення. Дані більше не поділяються на структуровані і реляційні, а поділяються на неструктуровані і гетерогені (коментарі, відео, зображення, аудіо, дані з датчиків, тощо). Природа цих різнорідних даних не обмежена.

Спосіб ведення бізнесу, управління, та наукових досліджень був змінений з появою великих даних. Цей величезний обсяг інформації, показав межі можливостей її обробки традиційними методами. Справді, доступні інструменти аналізу не можуть впоратися зі зростанням розміру, складності і частоти змін даних. Тому постійно відбуваються зміни в архітектурі систем агрегації і обробки даних.

Отже, виділяють 5 основних властивостей великих даних:

- **Об'єм:** розмір отриманих даних. Уже існує більше даних, ніж будь-коли раніше, і їх розмір продовжує рости в геометричній прогресії: 90% всіх даних, наявних сьогодні були створені за останні два роки. Ще недавно, можна було говорити про гігабайти, зараз уже мова йде про терабайти, петабайти, ексабайти і навіть зеттабайти даних.
- **Швидкість:** швидкість, з якою дані генеруються і обробляються дані. Оскільки потоки даних надходять в реальному часі, то швидкість їх обробки стає вагомим фактором. Тому обробка даних в реальному часі стає однією з ключових проблем.
- **Різноманітність:** різниця типів зібраних даних. Дані, що аналізуються більше не структуровані, вони можуть бути текстом, зображеннями, мультимедіа контентом, даними з датчиків і т.д. Мова йде про додаткову цінність традиційних (структурованих) даних шляхом їх об'єднання з великою кількістю інших зовнішніх джерел.
- **Точність:** це найбільш важливий момент. Впевненість в точності зібраних даних зростає з кількістю їх джерел. Вона відображає акцент на необхідності якісних даних у системі.
- **Важливість:** важливість зібраних даних. Сортування даних є суттєвим. Важливо правильно вибрати дані, які будуть проаналізовані і відділити їх від не важливих даних.

Таким чином, реалізація стратегій взаємодії з клієнтами в бізнесі, стає чим далі більш персоналізованою. Цей процес використовує інформацію, яку генерує сам клієнт. Але такий підхід ускладнюється різнорідністю інформації про клієнта.

Для досягнення більшого ефекту персоналізації контенту, сьогодні багато де використовуються рекомендаційні системи.

Рекомендаційні системи (РС) — програми, які намагаються передбачити, який з

доступного контенту буде цікавий користувачу, базуючись на аналізі його профілю, та інформації від інших користувачів.

На сьогоднішній день майже всі РС тільки структуровані дані, представлені у реляційному вигляді. Для того, щоб покращити ці системи і перейти на наступний рівень рекомендацій - необхідно використання додаткових джерел неструктурованої інформації, що дозволить краще зрозуміти інтереси користувача.

Точні рекомендації дають користувачам можливість швидко знайти бажані речі без споживання великої кількості неактуальної інформації. Це також представляє великий інтерес для виробників та продавців, бо відкриває можливість персонально рекомендувати своїм клієнтам ті продукти, які відповідають їх інтересам.

Враховуючи кількість інформації про користувача, і її розрідженість та неоднорідність, гостро стоїть питання покращення точності рекомендацій. Не дивно, що в конкурсі Netflix[1], поліпшення точності рекомендацій на 10% було кращим результатом, і удостоєне призу 1 млн доларів США.

РС спираються на кілька суміжних дослідницьких дисциплін, таких як когнітивні науки, теорії наближень та пошуку інформації, і т.д. Через зростаючу важливість рекомендацій, цю галузь почали досліджувати незалежно з середини 1990-х років [8]. Взагалі, виділяють два основні підходи до рекомендацій: контент-орієнтовані підходи та колаборативна фільтрація (CF).

CF підходи можуть бути розбиті в свою чергу на основі моделі CF і підходи основані на аналізі найближчих сусідів (k nearest neighbour) [8] [35].

Підходи на основі моделі CF використовують для оцінки товару користувачем, щоб побудувати модель передбачення. Загальна ідея полягає в моделюванні взаємодії користувача та товару і визначається їх властивостями у системі, такими як клас користувача, до якого він належить, і категорія товару.

Нові можливості для подальшого підвищення точності РС відкриває активність в соціальних мережах. У реальному житті, люди часто питають порад друзів чи

знайомих перед покупкою продукту або послуги. Результати досліджень соціологів та психологів, показують, що люди схильні виказувати свою думку знайомим людям, і довіряти їхній рекомендації більше, ніж рекомендаціям незнайомих людей чи постачальників [35]. Популярні соціальні мережі, такі як Facebook [20], Twitter [19], і Youtube [17], забезпечують нові можливості для людей, у спілкуванні і побудові віртуальних спільнот. Соціальні мережі не тільки спростили можливість користувачам поділитися своєю думкою один з одним, але й служать як платформи для розробки нових алгоритмів для надання рекомендацій. РС з використанням соціальних мереж підвищують точність традиційних РС, враховуючи соціальні інтереси, довіру між користувачами в мережі, та інші відомості про користувача в якості додаткових даних.

Наприклад, користувач може читати конкретну новину про якусь подію тільки тому, що подія відбулася в тому місці, де він живе; користувачеві може подобається пісня, яку рекомендують його близькі друзі на Facebook. Суспільна довіра між парою друзів (u, v) може бути встановлена на основі явного зворотного зв'язку від споживача u щодо користувача v (наприклад, шляхом голосування), або це може бути виведено з неявного зворотного зв'язку (наприклад, частоти і кількості спілкування/обмінів електронної пошти між u і v). Різні алгоритми, що працюють з соціальними мережами, по-різному обробляють доступну інформацію і передають її в РС.

Метою даної роботи є комплексне дослідження можливих шляхів структурування і аналізу даних з соціальних мереж, для їх подальшого використання як додаткових джерел інформації для РС. Результатом проведених досліджень є практична частина роботи, що становить собою дослідження структуризації даних з соціальних мереж, їх аналізу і використання в якості додаткових даних для РС. В практичній частині використовуються відкриті набори даних з Kaggle[38], та Epinions[18]. Так як для дослідження довелося обробляти великі обсяги даних, то були використані методи розпаралелених обчислень.

2. ОГЛЯД ІСНУЮЧИХ РЕКОМЕНДАЦІЙНИХ СИСТЕМ ТА ЇХ АЛГОРИТМІВ

2.1. Рекомендаційні системи, що ґрунтується на зворотньому-зв'язку

Як уже було зазначено, є два основні варіанти рекомендаційних систем: орієнтовані на аналіз контенту і на колоборативну фільтрацію.

Основна ідея підходу на основі аналізу контенту - використання властивості елемента для передбачення інтересу користувача до нього. Наприклад, для книги, можна використовувати ім'я автора, жанр, ключові слова і мітки. Ці властивості потім співставляються з вподобаннями цільового користувача.

Основна ідея колаборативної фільтрації є використання оцінок від кожного окремого користувача. Оцінки користувача можна розділити на явні (наприклад, користувач оцінює річ на одиницю) і неявні (наприклад, користувач натискає на посилання, слухає пісні або купує річ). Коли в наявності є дані великої кількості оцінок користувачів, вони можуть бути використані для визначення вподобань подібних користувачів (наприклад, користувачі, які прослухали такий же набір пісень); користувачі з подібними смаками мають багато спільних пісень у своїх плейлістах, проте деякі пісні можуть відрізнятися - саме їх і можна рекомендувати користувачам, які мають такі ж смаки. На додаток, схожість між елементами може бути виявлено аналогічно. Наприклад, елементи можна назвати схожими, якщо вони, скажімо, прослухані чи переглянуті тими ж користувачами. Це основна ідея колаборативної фільтрації [12], [13].

Далі, мова буде йти про РС на основі колаборативної фільтрації.

Багато уваги в літературі було зосереджено на використанні даних явного оцінювання, зокрема рейтингів (наприклад, за шкалою від однієї до п'яти зірок), які присвоювалися речам чи місцям (наприклад, фільми, пісні, ресторани). Завдання в тому, щоб передбачити значення інших елементів, рейтинг яких

поставив користувач, і точність рекомендації, як правило, вимірюється в термінах кореня седньоквадратичної похибки (Root Mean Square Error, RMSE), та середньої абсолютної похибки (Mean Absolute Error, MAE). Згідно з основною концепцією колаборативної фільтрації, були запропоновані різні види алгоритмів "найближчих сусідів" (k nearest neighbours), які можуть бути використані в моделях користувач-користувач та продукт-продукт окремо чи комбіновано [12]. Одним з найкращих методів себе показав, метод розкладу матриці [11, 12]. Цей підхід добре себе зарекомендував у комп'ютерному зорі [33] та аналізі тексту [32]. Саме базове розкладання матриці, що використовується - сингулярне розкладання, також були розроблені численні більш складні підходи [11, 12]. Основна ідея полягає у відображенні користувачів і речей в низькорозмірних просторах, і в них визначення подібності.

Використання неявних данних отримало значно менше уваги в літературі. Відомі публікації в цій області [22, 23], спрямовані на рекомендації телепередач користувачам на основі їх попередньої історії перегляду, наприклад, як багато часу вони витратили на кожний вид телевізійних програм.

2.2. Соціальні мережі як додаткове джерело данних для РС

Соціальні мережі з великою кількістю користувачів, такі як Facebook, твіттер, чи Youtube, створюють нові шляхи для спілкування людей, і створення віртуальних спільнот. Результати в соціології та психології показують, що людські істоти схильні створювати зв'язки та ділитися інформацією з іншими людьми.

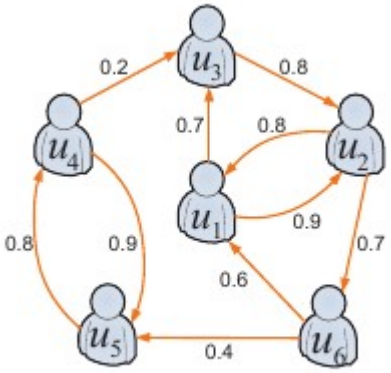

Рис. 2.2.1 - Соціальний граф

Можна побачити, як інформація з соціальних мереж може бути застосована для алгоритмів РС для підвищення точності рекомендації. Припускаємо, що користувачі є в соціальній мережі загального типу, такій як Facebook, або предметно-орієнтованій мережі, як Last.fm для отримання рекомендацій щодо кіно, чи Epinions для широкого діапазону рекомендацій.

Зобразимо соціальну мережу як орієнтований граф G = (U, F), де U є набір користувачів з | U | = u_0, а F - зв'язки дружби. Ця ж інформація представлена у вигляді як і наскільки користувач u довіряє або знає користувача v в соціальній мережі. Це можна зрозуміти з матриці S, розмірності $u_0 \times u_0$, де показані коефіцієнти довіри між усіма користувачами з U. Кожен користувач u має набір Fu безпосередніх сусідів, що їм довіряє, і в той же час, u довіряють Fu - користувачів. Прямі соціальні відносини користувача u з користувачем v (наприклад користувач u знає користувача v) представлені позитивним значенням $S_{u,v} \in (0, 1]$. Відсутні або приховані соціальні зв'язки - $S_{u,v} = s_m$, де, як правило, $s_m = 0$. соціальну вагу $S_{u,v}$ можна інтерпретувати як міру, що показує, наскільки користувач u довіряє користувачу v. Це може бути видно по явному зворотньому зв'язку користувача u та користувача v (наприклад, шляхом голосування), або з неявного зворотного зв'язку (наприклад, від ступеня взаємодії / зв'язку). Як правило, довіра між користувачами u та v - $S_{u,v}$, невід'ємна. В особливих випадках, вона може приймати негативні значення, наприклад, якщо два користувача мають суперечливі смаки. Рисунок 2.1

ілюструє приклад соціальної мережі з шести користувачів, кожен з яких має набір друзів. Кожна дружба, відмічена позитивним значенням довіри. Це також відображено у вигляді матриці S, в таблиці 2.2.1.

	U_1	U_2	U_3	U_4	U_5	U_6
U_1		0.9	0.7			
U_2	0.8					0.7
U_3		0.8				
U_4			0.2		0.9	
U_5				0.8		
U_6	0.6				0.4	

Таблиця 2.2.1 – таблиця довіри в соціальних мережах

Отже, існуючі алгоритми для РС пропонують рішення, що дозволяють використання даних з соціальних мереж:

- Створення РС для соціальних мереж на основі Байесового виведення (Bayesian-inference). Це забезпечує хороший компроміс між захистом конфіденційності користувача і точності рекомендації.
- Використання концепції «категорій друзів», для покращення точності рекомендацій.
- Вивчення онлайн соціального голосування

2.3. Рекомендаційні системи для соціальних мереж, на основі Байесового виведення

Розглянемо рекомендаційну систему на основі Байесового виведення для соціальних мереж. У цій системі, користувачі діляться своїми оцінками контенту з друзями. Подібність оцінок між парою друзів вимірюється набором умовних ймовірностей, отриманих від загальної історії їх оцінок. Користувач поширює запит на оцінку у соціальній мережі у своїх прямих і непрямих друзів. Ґрунтуючись на наданих відповідях інших користувачів, байєсівська мережа виводить рейтинг запитів користувача. Існують розподілені протоколи, які можуть бути легко реалізовані в соціальних мережах.

Алгоритм працює на основі основних соціальних мереж загального призначення, таких як Facebook, так і предметно-орієнтованих соціальних мережах, таких як Last.fm. Алгоритм може бути реалізований в стандартній окремий додаток.

Для роботи алгоритму, користувач має бути підключений до соціальної мережі. Друзі діляться один з одним своїми оцінками подій, фільмів, і т.д. Даний алгоритм відстежує рейтингові історії користувачів, і співвідносить їх між друзями. Кожен раз, коли користувач хоче отримати рекомендацію до фільму, програмний додаток надсилає запит до його прямих і непрямих друзів у соціальній мережі. Люди, які дивилися / оцінили фільм в минулому - діляться на запит з їх оцінками. На підставі зібраних оцінок між друзями, алгоритм обчислює бал рекомендації для запитів користувача. У цьому контексті робляться такі припущення:

- Рейтинг схожості між друзями в соціальній мережі пропонується вимірювати за допомогою набору рейтингових умовних ймовірностей, тобто побудовою мережі Байеса, що ґрунтується на дереві поширення запиту.
- Алгоритм розраховує найбільш релевантну рекомендацію та среднеквадратичну помилку Байєсової мережі (MSE) для запитів користувача.
- Використовується максимальна апостеріорна (MAP) оцінка, для того, щоб впоратися з проблемою холодного старту і розрідженостю оцінок користувачів.
- Алгоритм оцінюється з використанням двох різних наборів онлайн оцінок реальних користувачів. За рахунок цього, система побудована на Байєсовому виведенні, показує рекомендації, співставні з рекомендаціями систем, побудованих на основі колаборативної фільтрації. Це дозволяє мати гнучкий компроміс між якістю і кількістю рекомендацій.

2.3.1. Пов'язані роботи

Колаборативна фільтрація (CF) є найпопулярнішим підходом до побудови рекомендаційних систем і успішно застосовується в багатьох додатках. Для рекомендації за допомогою CF використовують думки кількох користувачів, щоб передбачити інтерес іншого користувача [1,4, 6, 32]. В роботі [4] запропонований імовірнісний підхід на основі CF до рекомендації оцінок для профіля нового користувача. За допомогою схеми активного навчання, профіль нового користувача може отримати достатньо даних для якісної рекомендації з мінімумом необхідного зусилля. В роботі [32] розглядаються ключові рішення в оцінці спільних систем фільтрації рекомендувача. В роботі [6] розглядаються моделі розкладу матриць в методах CF. За допомогою прикладу Netflix-конкурсу, автор [6] показує, що методи розкладу матриці стали домінуючою методологією в рекомендаціях за допомогою колаборативної фільтрації. Автори робіт [4,6] не займалися рекомендаціями в соціальних мережах і розподіленою рекомендаційною системою.

З швидким зростанням онлайнових соціальних мереж, довірчий підхід CF став визначальним. В цільові основі CF підходів, оцінки користувачів, із соціальних мереж порівняні з оцінками «еталонних» користувачів, які також враховуються для рекомендацій.

Основною передумовою CF підходу є те, що ті, хто погодився в минулому, як правило, погоджуються в майбутньому. Схема рекомендацій описана в цій главі відноситься до категорії CF.

2.3.2. Колаборативна фільтрація

У жовтні 2006 року, коли Netflix [1] розпочав свій «конкурс», інтерес до CF значно виріс. CF на основі аналізу сусідів і CF на основі прихованих факторів - два основні методи роботи РС.

Колаборативні методи аналізу сусідів в основі є найбільш ефективним при виявленні дуже локалізованих відносин, в той час як CF на основі прихованих факторів - як правило, ефективні при використанні всієї рейтингової інформації

користувача, як показано в роботі [6].

Позначимо число користувачів, як u_0 і кількость елементів, як i_0. Рейтинг $R_{u,i}$, який означає перевагу (експертність) користувача u, де високі значення означають більш сильну перевагу. Розрізняються передбачені оцінки із відомого числа, використовуючи позначення $\hat{R}_{u,i}$, для прогнозованого значення $R_{u,i.}$ Для методу, основаного на найближчих сусідах користувача:

$$\hat{R}_{u,i} = \bar{R}_u + \frac{\sum_{v \in N_u} sim(u,v)(R_{v,i} - \bar{R}_v)}{\sum_{v \in N_u} sim(u,v)} \tag{2.3.1}$$

де $\bar{R}_u$ є середня оцінка користувача u і N_u є вибрані сусіди користувача u. Використовується коефіцієнт кореляції Пірсона для вимірювання подібності двох користувачів. Коефіцієнт кореляції Пірсона між користувачем u і v визначається як:

$$sim(u,v) = \frac{\sum_{i \in I_C} (R_{ui} - \bar{R}_u)(R_{vi} - \bar{R}_v)}{\sqrt{\sum_{i \in I_C} (R_{ui} - \bar{R}_u)^2 \sum_{i \in I_C} (R_{vi} - \bar{R}_v)^2}} \tag{2.3.2}$$

де I_C набір загальних елементів, оцінених u і v, R_{ui} і R_{vi} їх рейтинги для фільму $i \in I_C$.. Назвемо цей метод **KNN** (K-Nearest Neighbor, K-найближчих сусідів).

Матричні моделі як користувачів, так і елементів (речей, подій) для спільного простору розмірності j_0, створюються таким чином, що взаємодії елемента(речі,події)-користувача моделюються як внутрішні в просторі фактора.

Відповідно, кожен елемент i пов'язаний з вектором $P_i \in \mathrm{R}^{1 \times j0}$, і кожен користувач u пов'язаний з вектором $Q_u \in \mathrm{R}^{1 \times j0}$. Скаляр $Q_u P_i^T$ описує взаємодію між користувачем u та елементом i - загальний інтерес користувача в елементі, визначений по його характеристиках.

Передбачений рейтинг моделюється як

$$\hat{R}_{u,i} = r_m + Q_u P_i^T , \tag{2.3.3}$$

де P_i, $Q_u \in \mathrm{R}^{1 \times j0}$ і $r_m \in \mathrm{R}$ - підгоночний параметр в моделі.

(u,i) - частини для яких $R_{u,i}$ є відомими (навчальний набір). Для запобігання

перенавчання(overfitting), додається термін регуляризації λ> 0 параметр регуляризації. Ми використовуємо норму Фробеніуса, яка позначається як $\|.\|_F$, щоб упорядкувати вивчені деталі і користувальницькі приховані фактори. Треба звернути увагу, що зазвичай ряд особливостей $j_0 \ll min(u_0,i_0)$. В наших експериментах припускається $j_0 = 10$. Позначимо цю модель як **SVD** (Singular Value Decomposition).

В принципі, рекомендаційні системи на основі Байєсового виведення і **SVD** – це два різні види систем. Байєсівське виведення являє собою розподілену систему, яка може бути легко реалізована на основі існуючих соціальних мереж. Користувачам не потрібно турбуватися про їхню оціночну конфіденційність, тому що вони діляться історією оцінок тільки з прямими друзями. У той час як в централізованих РС, користувацька конфеденційність є великою проблемою. В роботі [34] показано алгоритм , який зв’язує користувачів IMDB [23] з Netflix. У той час як середньостатистичний абонент Netflix не може слідкувати за своєю історією оцінок фільмів, її можуть використати.

Тим часом, РС на основі Байєсового виведення використовує тільки локалізовану інформацію, що надає користувачеві хороший захист приватних даних. Метод **SVD** більш точний у прогнозуванні та рекомендаціях. Крім того, в комерційних РС, користувачі весь час оновлюють свої рейтингові профілі. РС що використовує Байєсове виведення, можна ефективно інтегрувати в новий або оновлений користувацький профіль. Тим часом, як SVD потребує перерахунку оцінок всіх користувачів і їх даних, якщо потрібно інтегрувати новостворені профілі.

2.3.3. Рекомендація на основі довіри

Довіра (Trust) недавно була ідентифікована як ефективний засіб для того щоб використати соціальні мережі для поліпшення якості рекомендації. Емпіричні дослідження в [32,35] показали кореляцію між довірою і подібністю користувача. Різні методи були запропоновані для включення довіри в підходи CF. Як правило, схожість оцінок між друзями визначається числовим

значенням, причому більше значення вказує на більш високий рівень довіри. Потім рекомендація розраховується для користувача як залежність від оцінок і пов'язується з значеннями довіри його друзів. На відміну від рекомендації основаній на довірі, використовується умовний розподіл ймовірностей для визначення схожості між користувачами. Розподіл ймовірностей надає більш багату інформацію, ніж значення довіри і дозволяє використовувати байесівське виведення мережі для проведення декількох непрямих рекомендацій в соціальних мережах.

Автор [35] запропонували імовірнісний алгоритм на основі поширення довіри для проектування та оцінки рекомендаційних систем. Оцінка інформації користувачів представлена дводольним графом, на якому поширюється довіра. [39] досліджує підхід на основі моделі для рекомендації в соціальних мережах, що використовують методи розкладу матриць. Він вивчає приховані риси користувача з користувацької матриці оцінок і підвищує точність латентних функцій користувача через поширення латентних функцій над соціальною мережею.

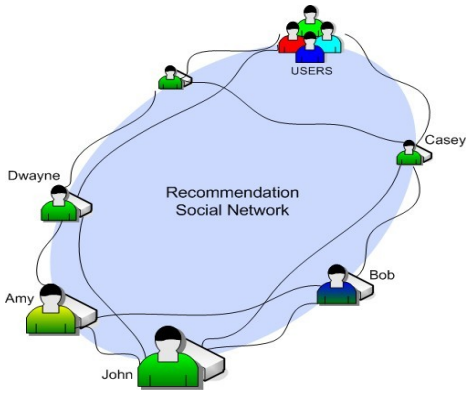

Рис 2.3.1 – Рекомендація в соціальній мережі

Нарешті, байесовскі підходи були використані в минулому в рекомендаційних системах. Попередня робота по використання байєсівського припущення, для рекомендації в соціальних мережах вивчається в роботі [35]. Зокрема, [35] утворює байєсівську мережу з вузлом, відповідним до елементу соціальної мережі. Стан вузлів відповідає можливим значенням оцінок. Алгоритм

проводить пошук на різних модельних структурах і вибирає кращу. У цій конструкції, вузол мережі Байєса - користувач. Структура байєсівської мережі визначається соціальною мережею, в якій подібні користувачі підключені один до одного. Крім того, рекомендація здійснюється в розподіленому режимі, таким чином, більш масштабована, ніж централізовані рекомендації.

2.3.4. Рекомендація на основі виведення Байєса

Як показано на малюнку 2.3.3.1 користувачі підключаються до соціальної мережі, використовуючи посередників - свої власні пристрої, наприклад, ПК, мобільні телефони, приставки і т.п. Після перегляду фільму, користувач може надати йому (фільму) рекомендацію / оцінку для своїх друзів. Користувач може також попросити у своїх друзів дати оцінку конкретного фільму. Наприклад, на малюнку 2.3.3.1, Джон є другом Емі, Боба Кейсі, і Даєн. Припустимо, Джон думає про те, чи слід йому дивитися новий реліз. Він запитує своїх друзів. Виявляється, що Емі, Боб, і Кейсі переглянули фільм, і поставили свої оцінки п'ять зірок, три зірки, і п'ять зірок. На підставі повернутих оцінок, Джоновий локальний рекомендаційний агент обчислює бал рекомендації, що, швидше за все, відображає інтерес Джона, і рекомендує фільм Джону, якщо рекомендаційна оцінка є високою.

Ключовою проблемою в цьому прикладі є оцінка рейтингу Джона з низькою похибкою. Найпростіший, наївний підхід – це рекомендувати Джонові середню оцінку своїх друзів. Основне припущення цього підходу є те, що оцінки друзів Джона в рівній мірі важливі для Джона. Насправді, смак Джона, може бути, "ближче" до смаку Боба, ніж до смаку Емі і Кейсі. Інтуїтивно, рейтинг Боба повинен мати більшу вагу, при оцінці рейтингу для Джона. Можна ввести значення "довіри" між друзями і використовувати «довірозважену» суму в якості рекомендації [35]. Тим не менш, це є складним завданням, представляти близькі рейтинги між друзями за допомогою одного числового значення. Важко пропагувати величини довіри через соціальну мережу, коли деякі рекомендації прийшли від непрямих друзів з різних соціальних мереж. Більш вишуканий

підхід – це заглянути в попарні кореляції рейтингів фільмів між Джоном і його друзями у минулому, і вивести "найбільш ймовірний" рейтинг Джона для поточного фільму. Більш конкретно, якщо Джон і Емі першу дивилися і оцінили загальний набір фільмів, рейтинг кореляція між ними може бути виміряна за допомогою набору умовних ймовірностей $P(R_J|R_A)$ і $P(R_A|R_J)$, де R_J і R_A є випадковими величинами, що представляють рейтинги Джона і Емі відповідно. Якщо Емі дає оцінку r_A для нового релізу, на основі рейтингової історії, найбільш імовірний рейтинг Джона для фільму може бути оцінений як

$$\hat{R_J}|R_A = r_A, \textbf{argmax}P(R_J = r|R_A = r_A) \qquad (2.3.4)$$

Точно так само, базуючись на рейтингах Боба і Кейсі, можна обчислити ще дві оцінки Джона $\hat{R_J}|R_B=r_B$ і $\hat{R_J}|R_C=r_C$.. Зрештою, необхідно зробити одну рекомендацію для Джона, використовуючи три оцінки на основі граничних умовних розподілів. В ідеалі, необхідно оцінити найбільш ймовірний рейтинг Джона за допомогою спільних рейтингів його друзів

$$\hat{R_J}|\{r_A,r_B,r_C\} , \textbf{argmax}_r P(R_J = r|r_A,r_B,r_C) \qquad (2.3.5)$$

де R {x} - абревіатура для події R {x} = r {x}, x =A, B, C. На жаль, на практиці, важко оцінити спільний умовний розподіл між Джоном і всіма його друзями. Замість цього використати байєсівську мережу, щоб відновити спільний умовний розподіл на основі граничних умовних розподілів між Джоном і кожним з його друзів. Точніше, побудувати одно-рівневе дерево Байеса між Джоном і його друзями: а) Джон корінь дерева; б) кожен з друзів Джона є прямим нащадком Джона в дереві. Після визначення байєсівської мережі, по суті, припускається, що рейтинги друзів Джона є незалежними один від одного: $\{R_A \perp R_B \perp R_C|R_J\}$. Отже, припускаємо, що рейтинг розбіжності між Джоном і його друзями є незалежними. У цьому припущенні, маємо

$$P(r_A,r_B,r_C|r_J) = P(r_A|r_J)P(r_B|r_J)P(r_C|r_J) \qquad (2.3.6)$$

За правилом Байеса, можна обчислити умовну ймовірність рейтингу Джона на основі оцінок своїх друзів, як

$$P(r_J|r_A, r_B, r_C) = \frac{P(r_A, r_B, r_C|r_J)P(r_J)}{\sum_r P(r_A, r_B, r_C|R_J = r)P(R_J = r)} \quad (2.3.7)$$

2.3.5. Проектування Рекомендаційної системи

У більш загальних настройках, коли користувач хоче мати рейтинг рекомендації для фільму, може статися, що жоден з його друзів безпосередньо не дивився / оцінював фільм. Для збільшення шансів на рекомендації, необхідно дозволити користувачу поширювати його запит через соціальну мережу і збирати рейтинги непрямих друзів з різних соціальних мереж. На підставі зібраних рейтингів, будується багаторівнева мережа Байєса , щоб оцінити найбільш ймовірний рейтинг для користувача, який дає запит.

Більш конкретно, розподілена рекомендаційна система працює в наступних рамках

- Люди пов'язані в соціальній мережі G = (V, E), де V є сукупність користувачів, E набір дружніх зв’язків.
- Кожен користувач ділиться оцінками з його безпосередніми друзями;
- Кожна пара друзів $(u,v) \in E$ вимірює їх рейтинги схожим набором умовних розподілів $P(u|v)$ and $P(v|u)$, кожен з яких є користувацьким розподіленням рейтингу обумовлений рейтингом інших користувачів;
- Користувач відсилає запит на рейтинг для фільму до його прямих друзів у соціальній мережі *G*. При отриманні запиту для фільму, користувач повертає його рейтинг, якщо він оцінив фільм раніше, в іншому випадку він ретранслює запит до своїх друзів. Щоб обмежити діапазон розповсюдження запиту, запит буде видалений після певного числа стрибків. Рекомендаційний агент оцінює результат для користувача запитувача, використовуючи байєсівську мережу.

Дерево Рекомендаційного Поширення

Нехай S ∈ V буде користувач, який згенерував запит для фільму. Нехай L = {L_i ∈ V,i = 1,2,...,k} проіндексовані прямі і непрямі друзі S, які відповідають на запит. Соціальна мережа зазвичай має багато з'єднань. Щоб уникнути зайвих відповідей на запит, створюється унікальний ідентифікатор для кожного запиту. Кожен користувач в L реагує тільки на запит, коли він приймає запит в перший раз. Припустимо далі, що відповіді на запит будуть повернені до S тим же шляхом, що і шлях поширення запиту. В результаті, об'єднання всіх шляхів реагування на запит, створюється дерево найкоротших шляхів з усіх рекомендацій в L із загальним коренем C.

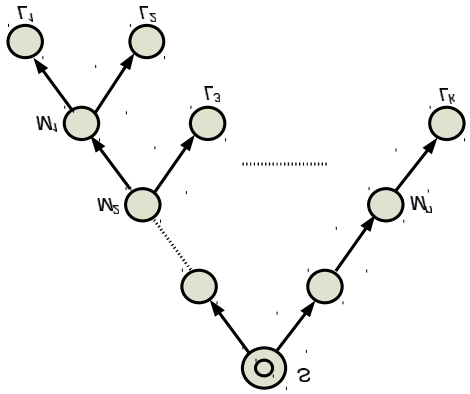

Рис 2.3.2 – Дерево рекомендаційного розповсюдження як Байєсівська мережа.

Малюнок 2.3.2 ілюструє приклад дерева рекомендаційного поширення, де S — виступає в якості кореня, а L — вузлами дерева. Висота дерева обмежена обсягом захоплення запиту. Тут є три типи користувачів: ініціатор запиту S, ті, що роблять рекомендації {L_i}, і проміжні користувачі {M_i}. Ті, що роблять рекомендації {L_i}, мають рейтинг рекомендації r_i для запитуваного фільму, і передають свій рейтинг до свого основного користувача. Проміжний користувач знаходиться на шляху відповіді того користувача, що робить рекомендації, і ініціатора запиту. Він збирає рекомендаційну інформацію від «своїх дітей», агрегує інформацію, збирає в одно ціле, і передає зведену інформацію до його «батьків». Нарешті, враховуючи структуру дерева рекомендаційного розповсюдження і рейтинги всіх користувачів листків дерева, ініціатор обчислює остаточний рейтинг рекомендації.

Байєсівська Мережа

Для роботи з рейтингами від прямих і непрямих друзів, використовується байєсівська мережа для розрахунку найбільш ймовірного рейтингу в корені. Побудуємо мережу Байєса з дерева рекомендаційного розповсюдження. Кожен користувач має свій наступний вузол на найкоротшому шляху до кореня в якості єдиного батька в байєсівській мережі. Іншими словами, припускаємо, що рейтинг користувача не залежить від рейтингів тих користувачів, які не брали участі в поширенні умовного рейтингу. Інтуїтивно кажучи, будь-який користувач в дереві рекомендаційного розповсюдження має сильнішу рейтингову кореляцію з його «батьком» ніж всі його користувачі які не брали участі в поширенні рейтингу. Його рейтингова залежність від користувачів не-нащадків "затемнена" рейтинговою залежністю його батьків. Використовуючи байєсівську мережу, можна обчислити умовну ймовірність $Pr(R_S = s|R_{Li} = r_i, 1 \le i \le K)$ на основі ймовірнісного розподілу рекомендаційного рейтингу та граничних умовних розподілів ймовірностей на всіх дружніх зв’язках в дереві розповсюдження.

Для будь-якого вузла m в дереві розповсюдження, визначаємо C_m, набір своїх прямих дітей.

Визначимо D_m як набір рекомендацій в корені піддерева при m:

$$D_m = \{L_i \in \mathrm{L} : L_i \textit{ в піддереві з коренем } m\} \qquad (2.3.8)$$

Рекомендації, згенеровані користувачами в D_m передаються вузлом m до кореня. Очевидно, що $D_S = \mathrm{L}$. Відповідно до протоколу пересилання запитів, якщо сам m є рекомендатором, він не буде направити запит далі до його друзів, і буде листковим вузлом в дереві розповсюдження: $D_m = \{m\}$, якщо $m \in \mathrm{L}$

Визначимо імовірнісну подію, що рекомендатор L_i дає рейтинг r_i як $\Phi(L_i)$, $\{R_{Li} = r_i\}$. Тоді загальний рейтинг всіх рекомендаторів під m може бути записаний як

$$\Psi_m \triangleq \{R_{L_i} = r_i, L_i \in D_m\} = \bigcap_{L_i \in D_m} \Phi(L_i). \tag{2.3.9}$$

Мета полягає в оцінці $P(R_S = s|\Psi_S)$. Враховуючи діапазон оцінок [1, N], за правилом Байеса, маємо:

$$P(R_S = s|\Psi_S) = \frac{P(\Psi_S|R_S = s)P(R_S = s)}{\sum_{r=1}^{N} P(\Psi_S|R_S = r)P(R_S = r)} \tag{2.3.10}$$

Оскільки $D_S = \cup_{c \in C_S} D_c$ і $\Psi_S = \cap_{c \in C_S} \Psi_c$, маємо

$$P(\Psi_S|R_S = s) = P(\cap \Psi_c|R_S = s) = \prod P(\Psi_c|R_S = s), \tag{2.3.11}$$

де остання еквівалентність визначається за умовної незалежності в байєсівській мережі. Якщо дитина c є Рекомендатором, тобто, $c \in$ L, then $D_c = \{c\}$, and $P(\Psi_c|R_S = s) = P(R_c = r_c|R_S = s)$, які можуть бути отримані безпосередньо з умовно імовірності між c та його батьків S. Якщо c є справжній вузол, тобто $c \notin$ L, то маємо:

$$\begin{aligned} P(\Psi_c|R_S = s) &= \sum_{i=1}^{N} P(\Psi_c \cap \{R_c = i\}|R_S = s) \\ &= \sum_{i=1}^{N} P(\Psi_c|R_c = i)P(R_c = i|R_S = s) \end{aligned} \tag{2.3.12}$$

де остання еквівалентність визначається за умовної незалежності рейтингів нащадків ˙ c’ в байєсівській мережі з S умовного рейтингу на c’. Останній член $P(R_c = i|R_S = s)$ при підсумовуванні легко отримати з маргінальних розподілів між c і S. Перший член $P(\Psi_c|R_c = i)$ може бути рекурсивно обчисленим за аналогічним процесом при зсуві вгору на один рівень в байєсівській мережі.

Розрахунки рекомендацій

Базуючись на байєсівській мережі, можна розрахувати один рекомендаційний рейтинг для ініціатора запиту S. Пропонуємо дві схеми розрахунку

рекомендації. Перший найбільш імовірна рекомендація, $\hat{S}^{MP}$, який максимізує загальну умовну ймовірність

$$\hat{S}^{MP} \triangleq \underset{1 \leq s \leq N}{\mathbf{argmax}} P(R_S = s|\Psi_S). \tag{2.3.13}$$

Друга визначає Байесові среднеквадратичні похибки , $\hat{S}^{MSE}$, і зводить їх до мінімуму.

$$\hat{S}^{MSE} \triangleq E[S|\Psi_S] = \sum_{s=1}^{N} sP(R_S = s|\Psi_S) \tag{2.3.14}$$

Відмітим, що, $\hat{S}^{MP}$ розраховується за (2.7) і завжди ціле число, але $\hat{S}^{MSE}$ розраховується за (2.8) і може бути дробовим.

Протокол Розподілення

Конструкція складається з двох основних компонентів:

- Навчальний Модуль: він спілкується з прямими друзями, обмінюється рекомендаціями і будує імовірнісний розподіл, необхідний для мережі Байеса. Навчальний модуль оцінює розподілення рекомендаційного рейтингу місцевого користувача, і умовне розподілення між локальним користувачем, і кожним з його друзів.
- Рекомендаційний модуль: він обчислює рекомендації для запитів користувачів на основі наявних оцінок всередині соціальної мережі. Запит поширюється на друзів, які можуть ретранслювати запит їх власним друзям. Кожна доступна рекомендація поширюється назад до ініціатора запиту в зворотному напрямку шляху поширення запиту. Байєсівський висновок на основі мережі проводиться в розподіленому режимі для обчислення оцінки рекомендації.

Нижче описані деталі модуля навчання для рекомендаційного двигуна.

2.3.6. Вивчаючи друзів

Індивідуальні користувачі повинні оцінити їх власне розподілення рейтингу

імовірності, і рекомендаційний рейтинг вірогідності їх прямих друзів, що зумовлений їх власними оцінками. Відповідно, навчання модуля підтримує базу даних, яка зберігає наступні лічильники: $\{n_i\}$ і $\{n^T_{j,i}\}$, де n_i є число випадків рейтингу *i*, зроблених локальним користувачем, і $n^T_{j,i}$, - це число випадків, що друг *T* дає рейтинг *j* для фільму, а локальний користувач дає рейтинг *i* однойменного фільму, де *i*, *j* = 1,2, ..., N. Емпіричні розподіли можна оцінити як $P(i) = n_i / \sum_{i=1}^{N} n_i$ і $P^T(j|i) = n^T_{j,i} / \sum_{j=1}^{N} n^T_{j,i}$.

Слід зазначити, що оцінка, яка базується на частоті, є максимальною оцінкою правдоподібності (MLE).

Всі лічильники напочатку рівні нулю. Коли користувач S робить рекомендацію {*movie id,s*}, навчальний модуль зберігає рекомендаційну пару в таблиці рекомендацій. Лічильник n_s збільшується на одиницю. Навчальний Модуль посилає {*S, movie id s*} до всіх прямих друзів. Коли друг T отримує повідомлення, його навчальний модуль проводить сервіс пошуку по його рекомендаційній таблиці, і визначає, чи T вже оцінив цей фільм. Якщо T ще не оцінив фільм, повідомлення відкидається без подальших дій. Якщо T оцінив фільм з рейтингом *t*, навчальний модуль T збільшує лічильник $n^S_{s,t}$ на одиницю. Крім того, T відправляє назад повідомлення про {*T,movie id,t*} до S. Це повідомлення, відповідно, дозволяє користувачеві S оновити свій умовний лічильник.

2.3.7. Розподілений модуль рекомендацій

Генерація запиту: Припустимо, користувач S запрошує рекомендацію. Рекомендаційна система посилає повідомлення запиту до сусідів по соціальній мережі. Запити бувають трьох типів [послідовність *id, movie id, TTL*]. Послідовність *id* – це унікальний ідентифікатор, ідентифікуючий цей запит. Наприклад, ідентифікатор може бути створений шляхом хешування ID фільму разом із власним ID користувача соціальної мережі. *movie id* ідентифікує фільм, для якого потрібна рекомендація. TTL, або time-to-live, визначає область

пошуку запиту. Запит затухає, коли його TTL йде до нуля.

Розповсюдження Запиту: Після отримання запиту, користувач спочатку перевіряє, чи він вже оцінив фільм. Якщо так, його рейтинг фільму повертається назад користувачеві, від якого отримано повідомлення запиту. Повідомлення запиту відкидається без подальшої переадресації. Якщо користувач ще не оцінив фільм, і значення TTL запиту є позитивним, користувач зменшує TTL запиту і пересилає запит до сусідів, за винятком того, від якою приймався запит. Нарешті, якщо TTL запиту дорівнює нулю, то він буде відкинутий. Користувач, який прийняв запит, відправляє повідомлення *DROP* назад користувачеві, від якого приймається запит, вказуючи, що запит був відкинутий без рекомендації. Соціальна мережа може мати петлі і користувач може отримати той же запит від кількох сусідів. Для спрощення виводу, користувач реагує тільки на сусіда, від якого запит приймається протягом першого часу і посилає повідомлення DROP для всіх інших сусідів.

Генерація Відповіді: Після процесу запиту, будується рекомендаційне дерево. Відповіді на Запит будуть поширюватися назад користувача, який дав запит, по дереву. Метою є дозволити кореневі (користувач, який робить запит) розрахувати умовні ймовірності, визначені в (2.3.4), (2.3.5) і (2.3.6). З точки зору кореня, умовні ймовірності в (2.3.5) і (2.3.6) розраховуються рекурсивним чином. Розрахунок реалізований у вигляді ітеративного алгоритму в байєсівській мережі, починаючи з кінцевих вузлів. Зокрема, всі листкові вузли відповіли на запит, відправивши свої рейтинги до їхніх батьків. Для проміжного користувача m, з урахуванням визначень в розділі 2.3, він відповідає за відправку набору умовних рейтингових ймовірностей , $P(\Psi_m|R_m = r), 1 \le r \le N$, до його батька. Щоб зробити це, m чекає відповідей після пересилання запиту на його сусідів. Якщо сусід повертає повідомлення DROP, ніяких дій не потрібно. Всі сусіди, що повертаються рейтинг або умовний імовірносний рейтинг діти m's в дереві розповсюдження. Подібно до формул (2.3.10) і (2.3.11)

$$P(\Psi_m|R_m = r) = \prod_{c \in C_m} P(\Psi_c|R_m = r$$

Якщо дитина m's $c \in C_m$ є рекомендатором, тобто $c \in$ L, then $P(\Psi_c|R_m = r) = P(R_c = r_c|R_m = r)$; в іншому випадку

$P(\Psi_c|R_m = r) = \sum_{i=1}^{N} P(\Psi_c|R_c = i)P(R_c = i|R_m = r)$.

Після того, як m збере рейтинги $R_c = r_c$ від його дітей, які дають рекомендації, і умовні ймовірності $P(\Psi_c|R_c = i)$ від його ретрансляційних дітей, він може обчислити спільну умовну ймовірність $P(\Psi_m|R_m = r)$, яка базується на граничній умовній ймовірності $\{P(R_c = r_c|R_m = r), \forall c \in C_m\}$, яка була обчислена Навчальним Модулем. Розрахована умовна ймовірність буде послана m's батькові в якості відповіді на запит.

Рекомендаційний розрахунок: Після того, як користувач S, який відправив запит, отримує відповіді від усіх своїх дітей, він проведе обчислення $P(R_S = s|\Psi_S)$ слідом за (2.4), (2.5) і (2.6). Нарешті, рекомендаційна машина на S рекомендує користувачеві найбільш імовірний рейтинг $S^{\hat{}MP}$ або Рейтинг квадратичної Похибки Байеса $S^{\hat{}MSE}$, відповідно до формул (2.3.7) і (2.3.8).

2.3.8. Проблема «холодного старту» і розрідженості оцінок

Навчальний модуль використовує рейтинги друзів, які часто дивиляться фільми, щоб побудувати умовний розподіл ймовірності. Заснований підхід часто використовується в оцінці ймовірності. Оцінка на основі частоти MLE не точна, коли число загальних фільмів, які спостерігались парою друзів, мале. Крім того, користувач, який нещодавно приєднався, не зробив будь-який рейтинг, таким чином, друзі не можуть побудувати історії, засновані на умовних ймовірностях проти нового користувача. Питання рейтингової розрідженості і холодного старту перешкоджають рекомендаційним підходам заснованим на колаборативній фільтрації.

Соціальна мережа пропонує нове місце, яке використовує користувачів для вирішення вищевказаних питань. У контексті соціальної мережі, новий користувач, швидше за все, знаєте, його / її друзів, і має спільні уявлення / думки про них. Новий користувач може дати керівництво про те, як повинен виглядати розподіл умовної ймовірності. Наприклад, новий користувач може

вказати рекомендаційній машині з імовірністю вісім (в діапазоні від 1 до 10), що його рейтинг такий же, як у сусіда, і з імовірністю п'ять, його рекомендаційний рейтинг, ймовірно одиниця, і так далі. За допомогою цієї інформації, рекомендаційна машина може побудувати інформативних апріорний розподіл для сусідів нового користувача. Точно так само, до сусідів нового користувача застосовують той же принцип і будують інформативні апріорні розподіли для нового користувача. Попереднє розподілення пояснюється невизначеністю у розподілі ймовірностей. По мірі того як фактичні рекомендаційні рейтинги накопичуються з плином часу, попереднє розподілення виправляється фактичними рейтингами, що відображає справжнього подібность між двома користувачами. Нижче описано оцінку ймовірності на основі максимального априорного розподілу (MAP).

Нехай $\mathbf{p} = [p_1,p_2,...,p_n]$ є масова функція (PMF) n-стану дискретної ймовірності, і $\mathbf{x} = \{x_i\}$, $i = 1,2,...,n$, число зразків для кожного стану. Якщо $g(\mathbf{p})$ - попереднє розподілення p, і MAP оцінка, $\mathbf{p}^{\mathbf{MAP}}$, визначається як розподілення, який максимізує наступне **p**.

$$\mathbf{p}^{\mathbf{MAP}} = \mathbf{argmax}\, f(\mathbf{x}|\mathbf{p})g(\mathbf{p}), \tag{2.3.16}$$

де $f(\mathbf{x}|\mathbf{p})$ - ймовірність **x** дана **p**.

Для попереднього розподілу вибираємо розподіл Діріхле. Розподіл Діріхле пов'язаний з дискретним розподілом ймовірностей, спрощує висновок.

Розподіл Діріхле визначається як

$$g(p_1, p_2, \ldots, p_{n-1}; \alpha_1, \alpha_2, \ldots, \alpha_{n-1}, \alpha_n) = \frac{1}{Z}\Pi_{i=1}^{n} p_i^{\alpha_i - 1}$$

де $\alpha_i > 0, p_i > 0, \sum_{i=1}^{n-1} p_i < 1, p_n = 1 - \sum_{i=1}^{n-1} p_i$ і Z є нормалізаційною постійною

Розподіл Діріхле визначає розподіл **p** заданими параметрами $\{\alpha_i\}$. Таким чином,

$$\begin{aligned} f(\mathbf{x}|\mathbf{p})g(\mathbf{p}) &= \Pi_{i=1}^{n} p_i^{x_i} \cdot \frac{1}{Z}\Pi_{j=1}^{n} p_j^{\alpha_j - 1} \\ &= \frac{1}{Z}\Pi_{i=1}^{n} p_i^{x_i + \alpha_i - 1}. \end{aligned} \tag{2.3.17}$$

Вирішуючи (4.16), отримаєм:

$$p_i^{MAP} = \frac{x_i + \alpha_i - 1}{\sum_{i=1}^{n}(x_i + \alpha_i - 1)}. \qquad (2.3.18)$$

На початку, без будь-яких зауважень / зразків,

$$p_i^{MAP} = \frac{\alpha_i - 1}{\sum_{i=1}^{n}(\alpha_i - 1)}. \qquad (2.3.19)$$

Параметри $\{\alpha_i\}$ повиненні бути встановлені на основі користувацьких входів. Значення $\sum_{i=1}^{n}(\alpha_i - 1)$ відіграє важливу роль у визначенні, наскільки швидко вплив попереднього розподілу зменшується з збільшенням кількісті доступних зразків.

• **Налаштування параметрів попереднього розподілу.** Параметри попереднього розподілу Діріхле, $\{\alpha_i\}$, повинні бути встановлені таким чином, що (i), вони відображають рівень довіри користувача до свого сусіда в термінах схожості їх думок про фільми; і (ii) вплив попереднього розподілу повиннен зменшуватися, оскільки назбирується більше прикладів. Враховуючи рейтингову шкалу [1, N], для кожного друга, користувач вибирає N довірчих значень $\{c_i, 0 \le i \le N - 1\}$, де c_i представляє упевненість, що абсолютна різниця рейтингів між ним і другом = i. Наприклад, c_0 представляє його рівень довіри, що їхні рейтинги в повній мірі узгоджуються один з одним. Рівень довіри кількісно може бути від 1 до 10: чим більше число, тим вище впевненість. Користувач може вводити свої рівні довіри до своїх сусідів, використовуючи простий GUI інтерфейс.

Налаштування параметрів Діріхле досить просте. Нехай користувач X має сусіда Y. Позначимо через R_X і R_Y рекомендаційні рейтинги користувачів X і Y . Користувач X встановлює початковий умовний розподіл ймовірності $P(R_Y = h|R_X = l)$, $l,h = 1,2,...,N$, як

$$P(R_Y = h|R_X = l) = c_{|l-h|}/\sum_{i=1}^{N} c_{|l-i|}$$

Позначимо $\{\alpha_i^l, i = 1, 2, \ldots, N\}$ через параметри попереднього розподілу Діріхле для P(RY |RX = l). αhl = KP(Ry = h|Rx = l) + 1. Оцінка ймовірності на основі MAP може бути виконана у відповідності з рівнянням (2.3.11). Величина K

обрається так, щоб правильно управляти впливом попереднього розподілу на актуальні приклади.

Порівняння з колаборативною фільтрацією

По-перше, порівняємо продуктивність байєсівського припущення на основі KNN моделі, змінюючи величину K і будуючи криву рекомендації в залежності від MAE для KNN. Результат показаний на рис. 2.3.1. Згідно з малюнком 2.3.3, навчання через 3 рукостискання (тобто коли соціальний ланцюжок запитів від користувача має довжину 3) може досягти MAE з 0,791 з охопленням 79,7% при завданні порогу ймовірності 0. Із рис 2.3.2, видно, що KNN наближається до MAE з 0,765 з охопленням 81,9%, коли розмір околиці K = 1700. З моделі через 3 рукостисканнями з результатом навчання, можна досягти такої ж MAE, як в моделі KNN з охопленням 73% при установці порога ймовірності до 0,37. Якщо граничне значення ймовірності встановити вище, система може забезпечити ще більш точні рекомендації.

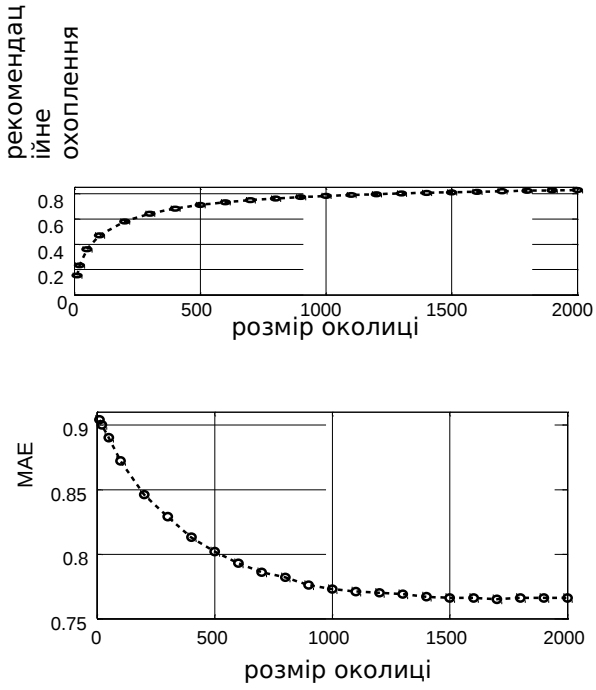

Рис.2.3.3 - MAE і результат охоплення по методу KNN. Верхній Рис. рекомендаційне охоплення в залежності CF розміру околиці. Нижній Рис залежність MAE від CF розміру околиці.

По-друге, порівнюємо припущення Байєса як основу методу з SVD. Для методу SVD, MAE найкраще, коли установка λ = 0,05, RM = 3,6. Краще значення MAE = 0,755. Можна звернути увагу, що rm = 3,6, що дорівнює середньому значенню всіх рейтингів навчання. Також важливо, що рекомендаційне охоплення для методу SVD 100%. Рекомендації на основі припущення Байєса, з Зма-рукостисканнями з навчанням може досягти того ж MAE як SVD з охопленням 68%, якщо встановити поріг ймовірності 0,42.

Холодний старт

Тепер оцінюємо, чи пом'якшується ефект холодного старту. Використовуємо ту ж топологію, як і у попередніх експериментах, але змоделюємо весь набір даних трьома різними апріорними розподілами. Перший розподіл використовує рівні довіри {3,1.5,1,1,1}. Оскільки прямі друзі схожі один на одного.

Другий розподіл - половина користувачів (випадково обрані) інша половина використовує розподіл з усіма рівнями довіри, рівних одиниці. Це вказує на те, що користувачі не мають знань про рейтингові звички своїх друзів.

Третій розподіл - половина користувачів використовують добрий розподіл (як у першому випадку), половина користувачів використовують поганий розподіл, побудований з використанням рівнів довіри {1,1,1,1.5,3}. Поганий розподіл навмисно встановлює високі значення з імовірності рекомендацій.

Ділимо рекомендації вздовж лінії часу на п'ять рівних груп. Встановлюємо поріг ймовірності рівним 0,5. MAE хорошого розподілу приблизно залишається постійним, з незначним зростанням у зв'язку зі збільшенням числа рекомендацій. При поганому розподілі, початкова якість рекомендацій погана. Поганий розподіл генерує деякі рекомендації, які не узгоджуються з інтересами користувачів, отже з високою MAE. Значення MAE покращується протягом довгого часу, по міру того як все більше прикладів виправляють поганий розподіл.

Хороший розподіл здатний забезпечити набагато більше рекомендацій, чим інші два випадки. Люди з що не мають ніякого знання звичок своїх друзів, що

призводить до рекомендацій, пов'язаних з низькою ймовірністю і, таким чином, фільтруються на порозі ймовірності. З часом, умовний розподіл ймовірностей між друзями стає точнішим з збільшенням прикладів. Рекомендаційне охоплення відповідно збільшується.

Поганий розподіл генерує найменше рекомендацій. Удосконалення розподілу імовірності, дивись рівняння (2.3.11), перетворює вихідну умовну ймовірність для рівномірного розподілу, а потім до більш точного розподілу з усе більшою і більшою кількістю зразків. На протязі цього процесу, щільність ймовірності спочатку «розтягується», що призводить до меншого числа рекомендацій у порівнянні з двома іншими розподілами. В цілому, досвід показує, що установка точнного розподілу допомагає новим користувачам отримати широкі, точні рекомендації з самого початку, і рівномірний розподіл краще, ніж поганий

Розглянуто РС на основі Байесового виведення, яка використовує соціальну мережу, для забезпечення точних і персоналізованих рекомендацій. Використання умовних розподілів ймовірностей для вимірювання подібності рейтингу між друзями, в рамках байєсівської мережі на основі висновку призначена для розрахунку найбільш ймовірної рекомендації. Експерименти показують, що точність висновоку на основі байєсівської рекомендації краще або співставна з централізованими підходами, заснованих на CF і цільовими підходами, і може гнучко балансувати кількістю рекомендацій щодо точності рекомендації. Крім того, соціальна мережа дає унікальну можливість для вирішення питання холодного старту і питання розрідженості оцінок.

2.4. Рекомендаційні системи, що ґрунтуються на категорізації користувачів в соціальних мережах

Традиційні підходи колаборативної фільтрації передбачають інтереси користувачів за історією оцінок, які поставив користувач [1], [8], [9], [10], [13], [17] і [18]. Популярні соціальні мережі надають додаткову інформацію для

покращення рекомендацій, в основі яких лежать рейтингові оцінки користувача. Деякі РС використовують інформацію з соціальних мереж для підвищення точності рекомендації. Загальна концепція – що вподобання користувача аналогічні вподобанням його друзів в соціальних мережах.Тим часом, життя користувачів в мережі, та поза її межами, може суттєво відрізнятися. Багато соціальних мереж, на сьогодні мають фунцію категоризації друзів. Найпершою мережею, що ввела таку фунцію була Google+, в якої це реалізовано в вигляді “кіл” користувачів. Користувач може розподіляти своїх друзів по різним колам, таким, як сім'я, одногрупники, колеги, друзі по рибалці, тощо. Facebook також запровадила схожу функцію, за допомогою якої користувач може сортувати своїх друзів за групами, і ділитися контентом з певним списком друзів. У Twitter, користувачі можуть організувати людей, на яких вони підписані (followees) в "списки". Коли користувач, переглядає певний список, він побачить твіти тільки людей, які є в цьому списку.

РС також мають вигравати від категорізації користувачів. Користувач довіряє різним людям з різних причин. Наприклад, в контексті рекомендацій в різних категоріях, користувач може довіряти іншому користувачу у всьому, що стосується автомобілів, але не довіряти в тому, що стосується фільмів. Очевидно, що в такому разі, при рекомендації першому користувачеві в категорії автомобілів необхідно враховувати оцінки, які поставив другий користувач в цій категорії, і не враховувати оцінки цього користувача при рекомендаціїї, що стосується фільмів. В ідеальному випадку, до рекомендації, відомо, які “кола” необхідно враховувати під час рекомендаціїї, а які ні. Нажаль, в більшості випадків (як і з доступними датасетами, так і у випадку реального використання), категорії не завжди чітко розділені між собою. Тому, якщо використовувати тільки одну категорію для передбачення, то можна пропустити важливу інформацію, що міститься в іншій категорії, хоча на перший погляд, інша категорія не повинна мати відношення до даної рекомендації. Тим не меньше, навіть змішані категорії можуть допомогти при рекомендації, тому що

таким категоріям як “сім'я” чи “близькі друзі”, людина схильна довіряти в багатьох питаннях.

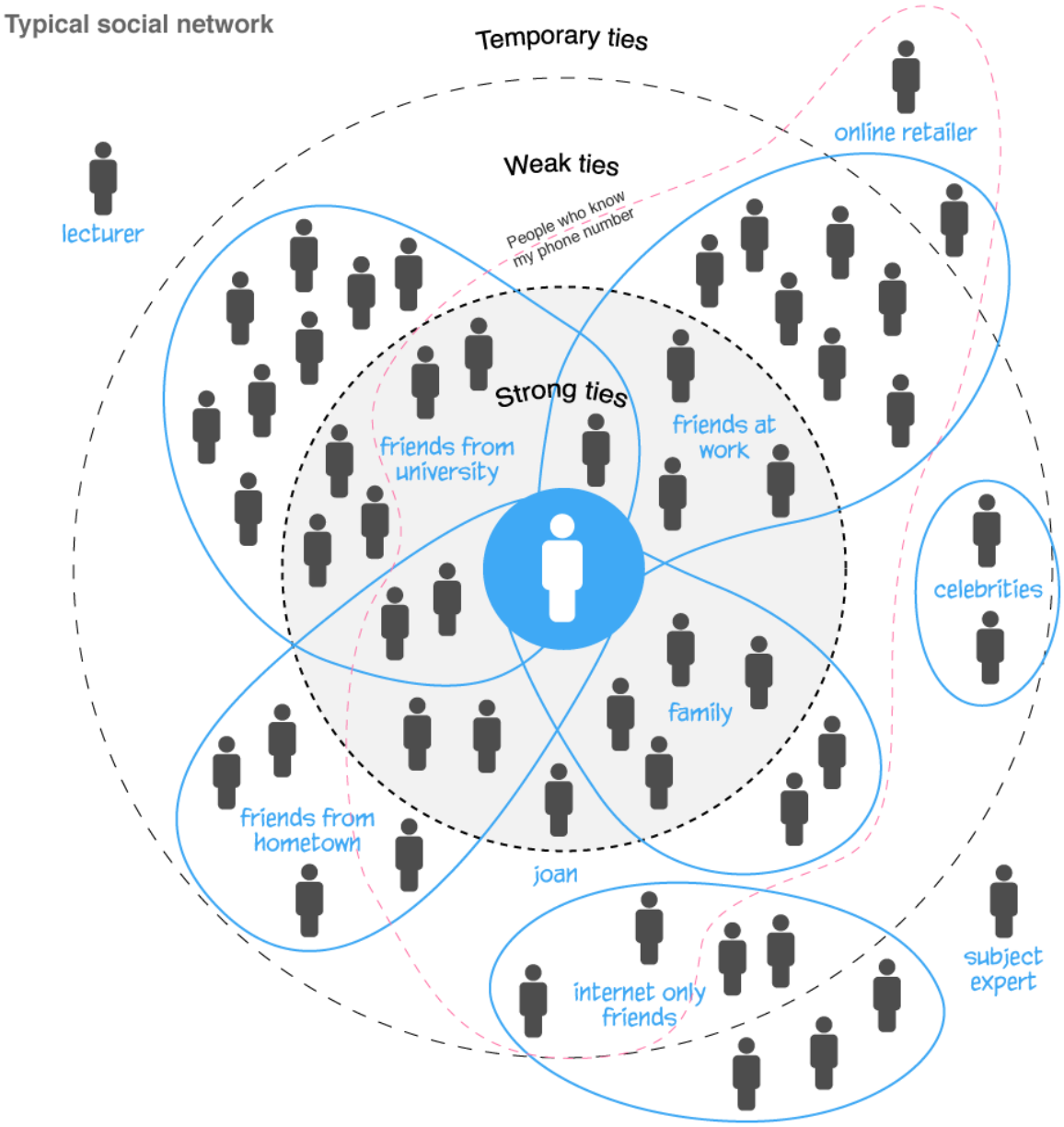

Рис. 2.4.1 – Розподіл друзів за категоріями в соціальній мережі

2.4.1. Споріднені дослідження

Моделі, які основані на розкладанні низькорангових матриць, вважаються одними з кращих для колаборативної фільтрації, так як у них найнижчий RMSE. Нижче наведено стислий огляд таких моделей.

Використання даних з соціальних мереж було використане для підвищення

точності прогнозування значень оцінок. Також різні моделі для інтеграції цих двох джерел даних були запропоновані, такі як Social Recommendation (SoRec) [40], Social Trust Ensemble (STE) [41], Recommender Systems with Social Regularization [42], Adaptive social similarities for recommender systems [43]. Для досягнення особливо низького значення RMSE була запропонована модель SocialMF [39] і була використана як базова модель.

В цій моделі, дані з соціальних мереж представляються у вигляді матриці $S \in \mathrm{R}^{u0\times u0}$, де u_0 це кількість користувачів. Прямі і зважині соціальні зв'язки користувача u з коритувачем v (u знає, довіряє, підписаний на v) це невід'ємне значення $S_{u,v} \in (0,1]$. *Відсутні або приховані зв'язки, позначені значенням 0. Кожен рядок нормалізовано до 1, і утворено нову матрицю* $S*$ *з* $Su,v* \propto Su,v$, *and* $\sum v\, Su,v* = 1$ *для кожного користувача u.*

Матриця передбачиних оцінок $\hat{R} \in \mathrm{R}^{u0\times i0}$, де u_0 це кількість користувачів, і i_0 – кількість речей, для яких будується передбачення, змодельована, як:

$$\hat{R} = r_m + QP^T \qquad (2.4.1)$$

Ідея, яка лежить в основі SocialMF – те, що сусіди в соціальній мережі можуть мати схожі інтереси. З цього випливає, що профіль користувача Q_u має бути в середньому схожим на профіль його друзів, в межах середньоквадратичної похибки

$$\begin{aligned} & \frac{1}{2}\sum_{(u,i)obs.}\left(R_{u,i}-\hat{R}_{u,i}\right)^2 \\ + \quad & \frac{\beta}{2}\sum_{\text{all } u}\left((Q_u-\sum_v S^*_{u,v}Q_v)(Q_u-\sum_v S^*_{u,v}Q_v)^\top\right) \\ + \quad & \frac{\lambda}{2}\left(||P||_F^2+||Q||_F^2\right), \end{aligned} \qquad (2.4.2)$$

де рейтинг $\hat{R}_{u,i}$, передбачений за цією моделлю (2.4.1).

Також треба звернути увагу, що при $\beta = 0$, вплив інформації, отриманої з соціальних мереж не враховується, і вага такої інформації збільшується, при збільшенні β.

Після того, як модель навчилася, обмеження, що користувачі мають мати схожі

профілі буде відображатися в матриці Q. І рейтинг будь-якого користувача може бути передбачино за допомогою рівняння (2.4.1)

2.4.2. Моделі рекомендацій на основі категорізації користувачів

Перша розглянута модель - розширена модель SocialMF [39] з включенням категорій користувачів.

Гіпотеза заключається в тому, що користувач довіряє різним друзям в оцінці предметів, що можуть бути розділені на різні категорії, і що користувач може довіряти кожному другу стосовно тільки однієї категорії, а не багатьох. Наприклад коло друзів, яким довіряють стосовно автомобілів, може бути зовсім інше, ніж те, кому довіряють в виборі фільму.

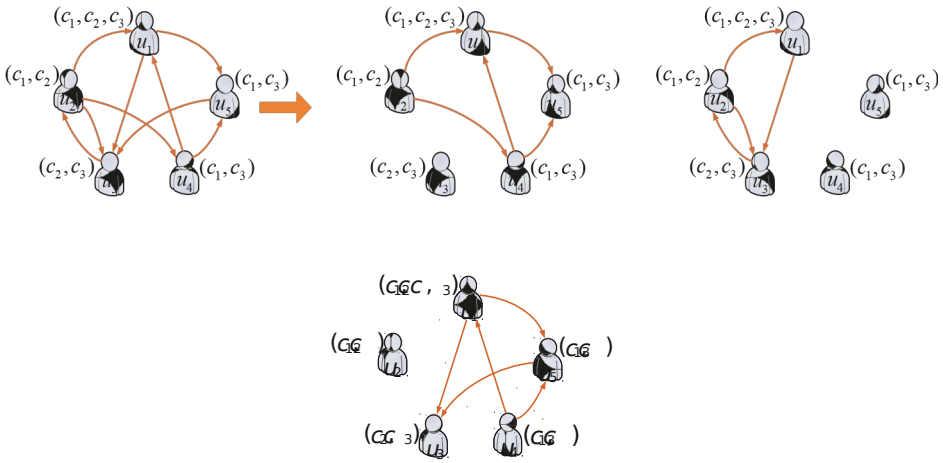

Рис. 2.4.2: Ілюстрація кіл, кожен користувач помічений в категорії, в якій він має рейтинг. а): оригінальна соціальна мережа; б), в) і г): передбачувані кола для категоріям С1, С2 і С3 відповідно.

З малюнку 2.4.2 видно, що мережа S розділяється на підмережі S(c), кожна з яких відповідає за певну категорію. Тобто, користувач v потрапляє в певну категорію користувача u, якщо виконуються наступні вимоги:

- Su,v > 0 в початковій мережі

- $N_u^{(c)} > 0$ і $N_v^{(c)} > 0$ в даних оцінок, які виставив користувач v

де Nu – кількість рейтингів, які користувач u поставив в даній категорії

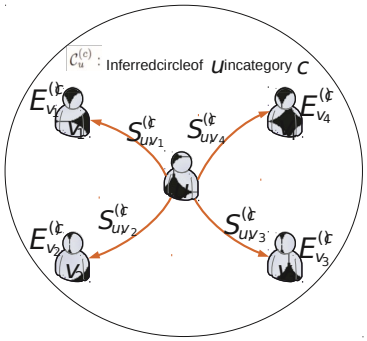

Рис. 2.4.3: Ілюстрація експертизи на основі оцінок довіри в категорії C

В іншому випадку, користувач v не знаходиться в категорії друзів користувача u, яким він довіряє в категорії C.

Значення ваги довіри між друзями в одній категорії C записується в матрицю S(c), так, що Su,v(c) = 0 if v ∈ Cu(c), Su,v(c) > 0, і якщо v ∈ Cu , то є декілька варіантів коли значення в матриці позитивне. Su,v > 0, і користувач v у відповідній категорії користувача u.

Умова1: однакова довіра

Найпростіший варіант визначення рівня довіри Su,v > 0 в категорії друзів користувача u в категорії C: кожен користувач v в “колі” користувача u отримує однарове значення довіри, тобто Su,v(c)∗ = const if v ∈ Cu(c)

Умова2: експертна оцінка

Найбільший рівень довіри надається користувачу, у якого найбільше досвіду в даній категорії. Як рівень його досвіду використовується кількість його оцінок в даній категорії. Суть полягає в тому, що більш досвідчений користувач виставляє більше оцінок в даній категорії.

З цього можна зробити висновок, що якщо користувач u довіряє користувачу v в категорії C, то u підписаний на v.

- Досвід v в категорії C рівний кількості виставленийх ним оцінок, тобто

$E_v^{(c)} = N_v^{(c)}$,

$$S_{u,v}^{(c)} = \begin{cases} N_v^{(c)} & \text{if } v \in \mathcal{C}_u^c \\ 0 \end{cases} \tag{2.4.3}$$

Тоді нормалізація кожного рядка матриці S(c) відбувається за формулою:

$$S_{u,v}^{(c)*} = \frac{S_{u,v}^{(c)}}{\sum_{v \in \mathcal{C}_u^{(c)}} S_{u,v}^{(c)}}, \tag{2.4.4}$$

що забезпечує нормалізацію для кожної категорії до 1, для всіх користувачів u, v:

$$\sum_{v \in \mathcal{C}_u^{(c)}} S_{u,v}^{(c)*} = 1$$

- У цьому випадку, рівень експертизи користувача v в категорії C - сума двох компонентів: перший компонент є кількість рейтингів, присвоєних у V категорії C, другий компонент є деяке значення оцінок в категорії C від усіх своїх послідовників v , якщо більшість послідовників V мають дуже багато рейтингів у категорії C, і всі вони довіряють v, це ознака того, що v є експертом в категорії C.

Якщо позначити оцінки користувача *v* в категорії c, як f(v,c), то для кожного, хто на нього підписаний, обчислюється розподіл його рейтингів в кожній категорії. Позначимо вектор розподілу як D_w

$$\mathcal{D}_w = \left(\frac{N_w(1)}{N_w}, \frac{N_w(2)}{N_w}, ..., \frac{N_w(m)}{N_w} \right), \tag{2.4.5}$$

де m — кількість категорій, N_w(c), де c = 1,...,m — кількість оцінок, які поставив користувач *w* в категорії c. N_w — загальна кількість оцінок, які поставив користувач *w*, $N_w = \sum c\ N_w(c)$. Тому D*w* показує пропорційний розподіл оцінок користувача *w* по категоріям. Тобто він відображає рівень компетентності користувача *w* у всіх категоріях.

Інший компонент, який можна назвати цінністю оцінки усіх підписників можна визначити за формулою:

$$\phi_v^{(c)} = \sum_{w \in \mathcal{F}_v^{(c)}} \mathcal{D}_w(c). \tag{2.4.6}$$

Скомпонувавши обидва компоненти, рівень компетентності (експертності) користувача визначається так:

$$E_v^{(c)} = N_v^{(c)} \cdot \sum_{w \in \mathcal{F}_v^{(c)}} \mathcal{D}_w(c) \tag{2.4.7}$$

Тобто значення для матриці довіри можна записати у такому вигляді:

$$S_{u,v}^{(c)} = \begin{cases} N_v^{(c)} \cdot \sum_{w \in \mathcal{F}_v^{(c)}} \mathcal{D}_w(c) & \text{if } v \in \mathcal{C}_u^{(c)} \\ 0 \end{cases} \tag{2.4.8}$$

Далі, пронормалізувавши кожен рядок матриці $S^{(c)}$, отримуємо:

$$S_{u,v}^{(c)*} = S_{u,v}^{(c)} / \sum_{v \in \mathcal{C}_u^{(c)}} S_{u,v}^{(c)}$$

Умова3: розділення довіри

Попередні висновоки базуються на припущенні, що, якщо йдеться про оцінку довіри до користувача v, то користувач u та користувач v мають оцінки в категорії c, то u довіряє v в c. Визначення значення довіри проведено окремо в кожній категорії друзів. На практиці, користувач u може визначити користувача v як експерта в категорії c тільки тому, що v має оцінки в підгрупі категорій, в яких вони одночасно мають оцінки. Іншими словами, враховуючи, що u довіряє

v, якщо v має більше оцінок у категорії c1 ніж в c2 ніж, то дуже імовірно, що u довіряє v через оцінки v в c1, ніж рейтинги v в c2. Тепер, якщо u і v одночасно мають рейтинги в кількох категоріях, то довіра u до v к повинна бути розділена між цими категоріями.

Тобто, якщо нормалізувати значення довіри між категоріями, то отримаємо:

$$S_{u,v}^{(c)} = \begin{cases} \frac{N_v^{(c)}}{\sum_{c:v \in \mathcal{C}_u^{(c)}} N_v^{(c)}} & \text{if } v \in \mathcal{C}_u^{(c)} \\ 0 \end{cases} \tag{2.4.9}$$

Для того, щоб проілюструвати розподіл, розглянемо рисунок 3.1. Користувач u_2 довіряє користувачу u_2 і вони обоє мають оцінки в категоріях c1 і c2. Допустимо, що кількість оцінок користувача u_1 – 9 в категорії c1 і 1 в категорії c2. Без нормалізації, отримуємо $S_{u2,u1} = 1$. Тим часом, використовуючи правило розподілу довіри, маємо $S^{c1}{}_{u2,u1} = 0.9$, а $S^{c2}{}_{u2,u1} = 0.1$.

Нормалізуючи кожен рядок матриці $S^{(c)}$, отримуємо:

$$S_{u,v}^{(c)*} = S_{u,v}^{(c)} / \sum_{v \in \mathcal{C}_u^{(c)}} S_{u,v}^{(c)}. \tag{2.4.10}$$

Можна сказати, що нормалізація значень довіри по категоріях c та користувачах v, можна розглядати як перший крок до тренування моделі.

3. ТЕХНОЛОГІЧНИЙ ПРОЦЕС РОЗРОБКИ

3.1. Постановка мети експеременту

Експерементальна частина роботи полягає у доведенні гіпотези, що модель SocialMF[39] можна розширити, включивши в розрахунок рекомендвацій додаткові дані з соціальної мережі. Мається на увазі, що крім вже згаданої категорізації друзів користувача і визначення довіри до них, пропонується додатково враховути інформацію, що знаходиться в профілі користувача.

Це включає в себе таку інформацію як місце розташування користувача (наприклад, йому не треба рекомендувати події, що відбудуться за тисячі кілометрів від нього, але якщо мова йде про новий фільм, то місце розташування користувача не має значення), його власні вподобання, вікову групу, тощо.

3.2. Системи керування версіями

Так як виконання магістерської дисертації супроводжувалось розробкою програмного коду, мною було прийнято рішення розгорнути інфраструктуру керування проектом, та систему керування версіями.

Система контролю версій дозволяє зберігати попередні версії файлів та завантажувати їх за потреби. Вона зберігає повну інформацію про версію кожного з файлів, а також повну структуру проекту на всіх стадіях розробки. Місце зберігання даних файлів називають репозиторієм. Всередині кожного з репозиторіїв можуть бути створені паралельні лінії розробки — гілки.

Гілки зазвичай використовують для зберігання експериментальних, незавершених (alpha, beta) та повністю робочих версій проекту (final). Більшість систем контролю версії дозволяють кожному з об'єктів присвоювати

теги, за допомогою яких можна формувати нові гілки та репозиторії.

Використання системи контролю версії є необхідним для роботи над великими проектами, над якими одночасно працює велика кількість розробників. Системи контролю версії надають ряд таких можливостей:

- Можливість створення різних варіантів одного документу;
- Документування всіх змін (коли ким було змінено/додано, хто який рядок змінив);
- Реалізацію функції контролю доступу користувачів до файлів.
- Можливість створювати документацію проекту з поетапним записом змін в залежності від версії;
- Можливість давати пояснення до змін та документувати їх.

Нижче буде розглянуто найпопулярніші на даний момент системи контролю версій.

Subversion (SVN)— централізована система. Дані зберігаються в єдиному сховищі. При збереженні нових версій використовується дельта-компресія, тобто система знаходить відмінності нової версії від попередньої і записує тільки їх, уникаючи непотрібного дублювання даних. Сховище може розташовуватися на локальному диску або на мережевому сервері. До локального сховища клієнт Subversion звертається безпосередньо.

Клієнти копіюють файли з сховища, створюючи локальні робочі копії, потім модифікують їх і публікують зміни в сховищі. Декілька клієнтів можуть одночасно звертатися до сховища.

Основними недоліками SVN є централізованість репозиторію, низька швидкість роботи, наявність помилок при змінах імені файлів та директорій і низька гнучкість роботи.

GIT – розподілена система контролю версій, основними перевагами якої є надзвичайно висока в порівнянні з SVN швидкість роботи, гнучкість, обмін

змінами напряму між учасниками розробки (peer-to-peer) та зручність використання у великих проектах.

Проте є й певні недоліки – перш за все це велика кількість —зайвих опцій та складність workflow, неефективність в невеликих проектах.

Mercurial — розподілена система керування версіями файлів та спільної роботи, розроблена для ефективної роботи з дуже великими репозиторіями коду. Mercurial спочатку був написаний для Linux, та пізніше портований під Windows, Mac OS X і більшість Unix-систем. У першу чергу він є консольною програмою. Всі його операції запускаються параметрами програми `hg`.

Так як у мене, як єдиного розробника, є досвід використання Git, він безкоштовний і підходить під поставлені задачі, то його було вибрано в якості системи контролю версій для даної роботи.

3.3. Керування проектом та задачами

Контроль проекту — це елемент, який забезпечує відповідність проекту графіку виконання. Контроль проекту починається з планування та закінчується звітом з виконання проекту, пронизуючи кожен елемент процесу керування проектом. Кожен проект має бути оцінений щодо рівня необхідного контролю: забагато контролю означає втрату часу, замало контролю означає збільшення ризиків. Системи контролю необхідні для оптимізації
витрат, ризиків, якості, комунікацій, часу, змін, закупівель та людських ресурсів. Якісний план впровадження системи характеризує:

- Стратегію, задля приведення розробки до загальніших цілей організації.
- Стандарти для нових систем.
- Політику керування проектом щодо часу та бюджету.
- Процедури, які описують процес.
- Оцінку якості змін.

Для автоматизації керування проектом використовують спеціалізоване

програмне забезпечення. Лідерами ринку в цій галузі на даний момент є YouTrack, Jiira та Redmine. Так як Youtrack та Jiira не є безкоштовним програмним забезпеченням, було обрано Redmaine, котрий поширюється за умовами ліцензії GPL. Крім безкоштовного надання використання відкритого програмного забезпечення, він має безліч переваг, зокрема:

- Безпеку та надійність, адже його код перевіряється користувачами;
- Велику кількість написаних користувачами розширень та гнучкість його настройки.

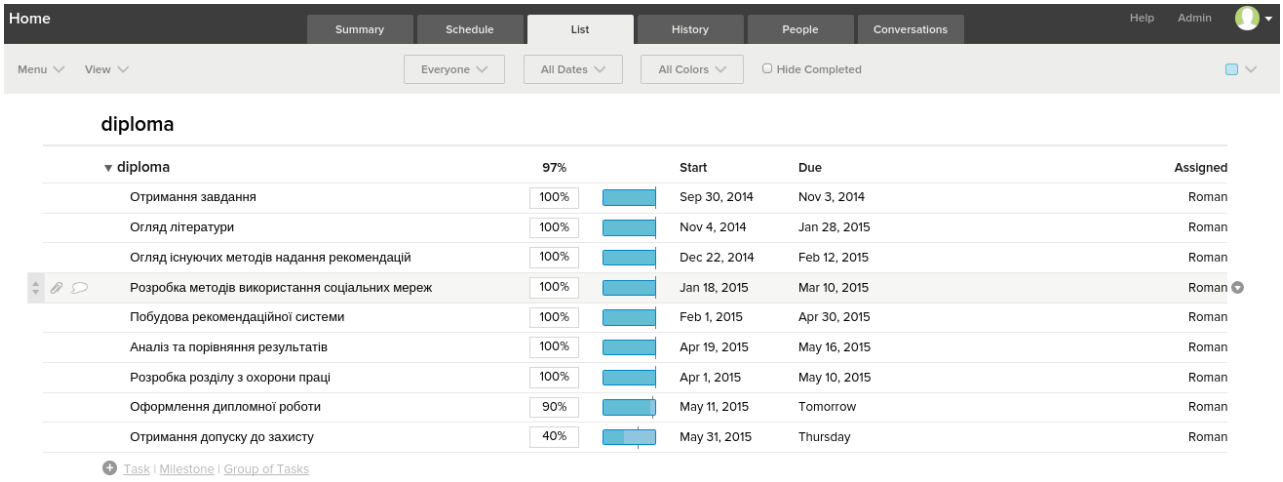

Рис 3.3.1 – Скріншот розгорнутого Redmine

Діаграма Ганта представляє собою відрізки (графічні плашки), розміщені на горизонтальній шкалі часу. Кожен відрізок відповідає окремому завданню або підзадачі. Завдання і підзадачі, складові плану, розміщуються по вертикалі. Початок, кінець і довжина відрізка на шкалі часу відповідають початку, кінцю і тривалості завдання. На деяких діаграмах Ганта також показується залежність між завданнями.

Завдяки використанню Діаграми Ганта стало можливим якісніше планування графіку виконання дипломної роботи.

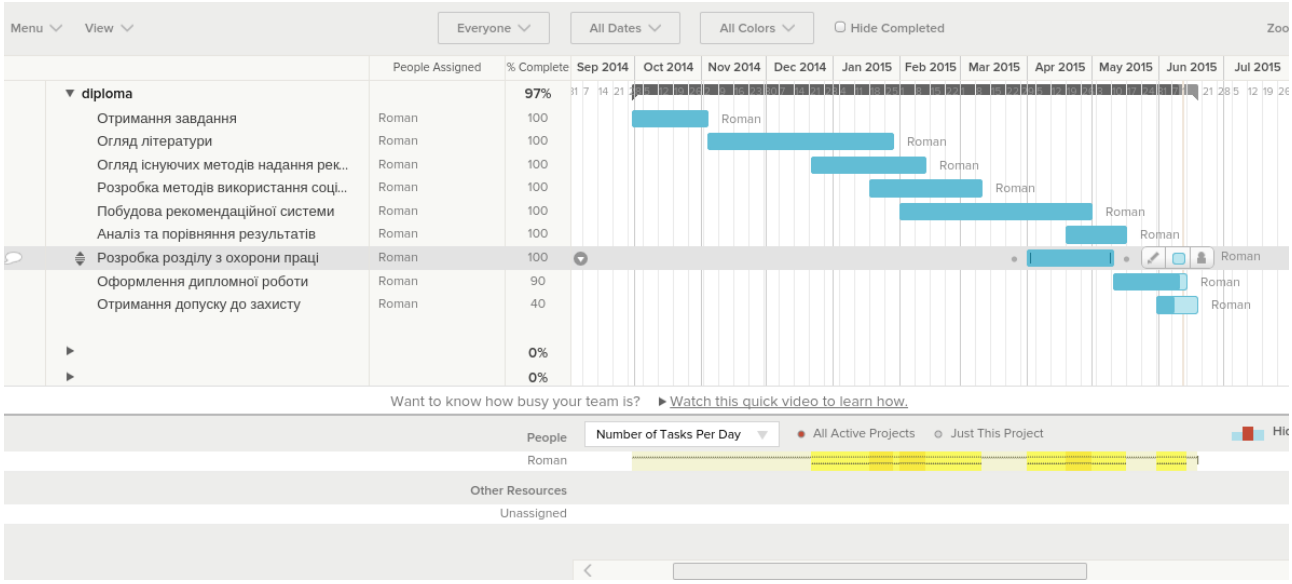

Рис 3.3.2 – Діаграма Ганта, що відображає виконання магістерської дисертації

В якості платформи для Redmine було обрано власний ноутбук. Даний вибір дозволив з легкістю сконфігурувати Redmine. Репозиторій було вирішено розмістити на платформі BitBucket, котра дозволяє безкоштовне користування в некомерційних цілях

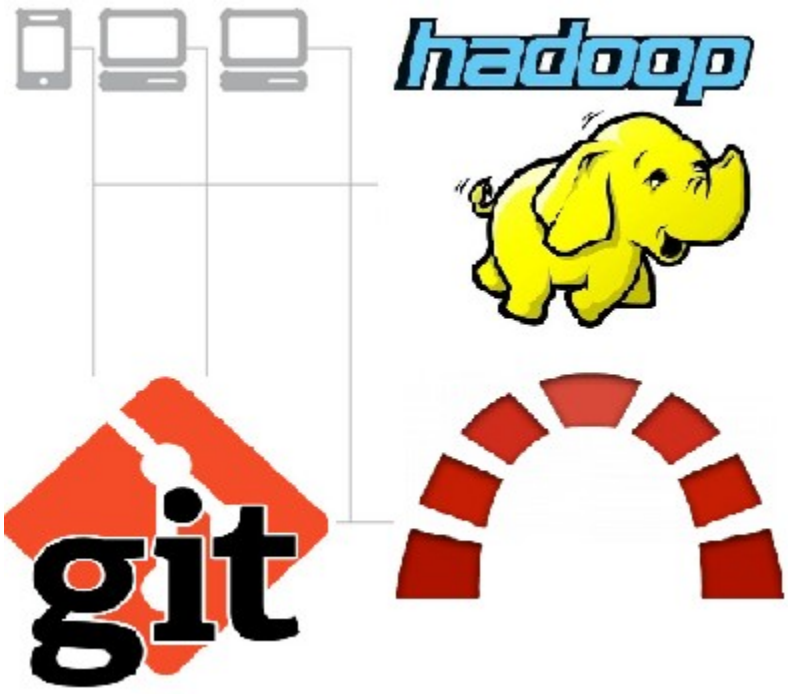

Рис 3.3.3 – Архітектура системи забезпечення технологічного процесу розробки дипломного проекту

3.4. Менеджмент даних

Так, як я не маю можливості перевірити гіпотизу на результатах роботи якогось великого інтернет-магазина, чи іншого високонагруженого(high load) порталу, де можна було б зібрати достатньо статистики для аналізу результатів, мною було вирішено провести тест гіпотези за допомогою відкритих наборів даних (детальніше описаних в розділі 5). Оскільки, в такому випадку відпадає потреба в рекомендаціях в реальному часі, а набори даних досить великі, я вирішив використати платформу розпаралелених обчислень Hadoop, для отримання результатів за розумний час.

3.4.1. Програмна платформа Hadoop

Hadoop - це програмна платформа (Software Framework) побудови розподілених додатків для масово-паралельної обробки (Massive Parallel Processing, MPP) даних.

Apache Hadoop включає в себе 3 підпроекти:

1. Hadoop Common: бібліотеки і сценарії керування розподіленою обробкою, файловою системою, розгортанням інфраструктури;
2. Hadoop Distributed File System: розподілена файлова система, яка забезпечує високошвидкісний доступ до даних додатка;
3. Hadoop MapReduce: програмна платформа для розподіленої обробки великих обсягів даних на обчислювальному кластері.

До основних технічних характеристик платформи Hadoop відносять:

- Масштабованість: платформа масштабується лінійно і дозволяє зберігати й обробляти петабайт даних;
- Стійкість до збоїв: всі надлишкові дані зберігаються, всі провалені завдання з обробки даних перезапускаються;
- Кросплатформеність: бібліотеки Hadoop написані (в основному) на Java, і можуть виконуватися в будь-якій операційній системі, що підтримує JVM (Java VM);
- Автоматичне розпаралелювання виконання завдання: Hadoop створює

«чисті» абстракції для розробників, знімаючи з них роботу з планування, контролю і агрегатування результатів паралельної обробки даних.

Центральне місце екосистеми Hadoop займає сховище даних (Data Storage). Hadoop підтримує зберігання як неструктурованих даних з розподіленої файлової системі HDFS, так і структурованих даних у нереляційних базі даних HBase.

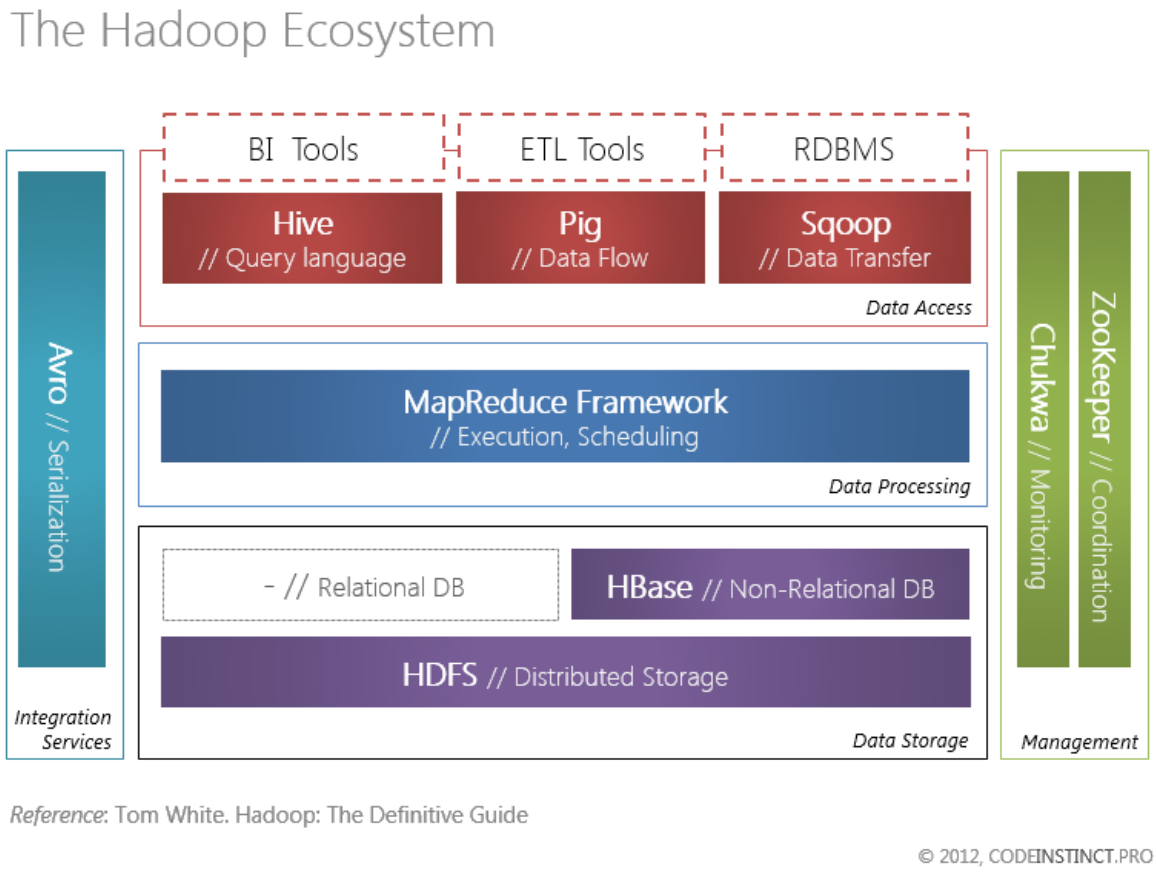

Рис 3.4.1 – Архітектура екосистеми Hadoop

Фреймворк MapReduce відповідає за планування завдань (Job Scheduling) і виконання розподілених обчислень.

Запити до наборів даних, які зберігаються на Hadoop-кластері, роблять за допомогою наступних інструментів, що входять в екосистему Hadoop: Pig, Hive (має свій SQL-подібну мову запитів HiveQL).

Для передачі великої кількості даних, що зберігаються на кластері Hadoop, до

сховищ структурованих даних, таких як, реляційні бази даних, розроблений інструмент Sqoop.

Компоненти, що відноситься до взаємодії (Integration Services), керування (Management), що відповідають за доступ до даних (Data Access), а також нереляційних БД HBase представлені окремими проектами Apache Foundation (часто верхнього рівня).

Hadoop Distributed File System (HDFS) - розподілена файлова система, яка забезпечує високошвидкісний доступ до додатка даних.

3.4.2. Концепції та структура HDFS

HDFS є ієрархічною файлової системою, в якій реалізована підтримка вкладення каталогів. У каталозі може розташовуватися нуль або більше файлів, а також будь-яку кількість підкаталогів.

HDFS складається з наступних обов'язкових компонентів:

- Вузол імен (NameNode) - програмний код, що виконується, в загальному випадку, на виділеній машині примірника HDFS і відповідає за файлові операції (роботу з метаданими);
- Вузол даних (DataNode) - програмний код, як правило, виконується на виділеній машині примірника HDFS і відповідає за операції рівня файлу (робота з блоками даних).

Hadoop містить єдиний вузол типу NameNode і довільну кількість вузлів типу DataNode.

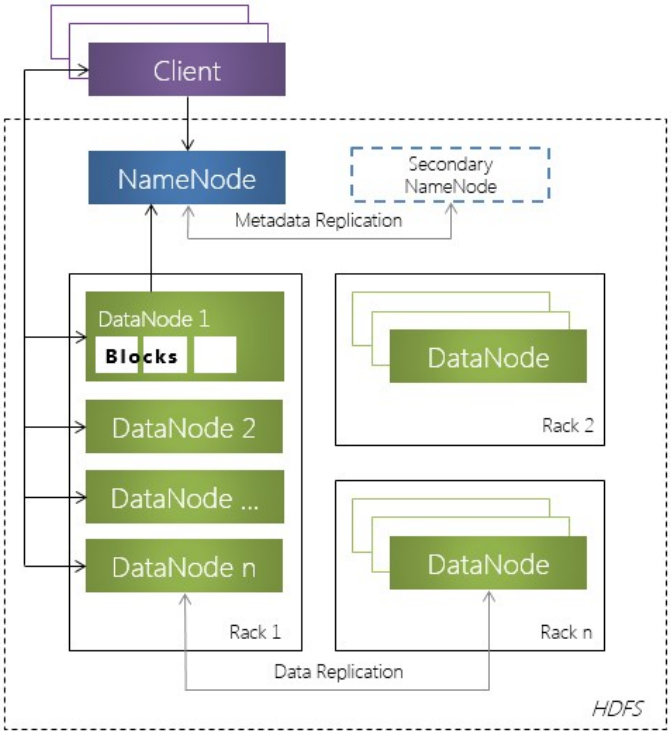

Рис 3.4.2 – Архітектура розгорнутої системи Hadoop

Основні концепції, закладені при проектуванні HDFS, і архітектурні рішення, застосовувані для реалізації цих концепцій, наведено нижче:

Обсяг даних

HDFS не повинна мати досяжних в осяжному майбутньому обмежень на обсяг збережених даних.

Архітектурні рішення:

- HDFS зберігає файли поблочно. Блоки в HDFS розподілені між вузлами даних обчислювального кластера. Всі блоки (крім останнього блоку файлу) мають однаковий розмір, крім того блок може бути розміщений на декількох вузлах.

Відмовостійкість

HDFS розцінює вихід з ладу вузла даних як норму, а не як виняток (і, дійсно, ймовірність виходу хоча б одного вузла з тисячі навіть на надійному фізичному обладнанні істотна).

Архітектурні рішення:

- Для забезпечення відмовостійкості всі дані в HDFS реплікуються настроюються кількість разів. Докладніше про те, як забезпечується реплікація в Hadoop-кластері на рівні файлової системи, описано в розділі «Файлові операції і реплікація».
- Захист від копіювання пошкоджених даних вирішена за допомогою зберігання контрольних сум в окремому прихованому файлі.
- Копіювання метаданих за допомогою вторинного вузла імен.

Самодіагностика

Діагностика справності вузлів у Hadoop-кластері не повинна вимагати додаткового адміністрування.

Архітектурні рішення:

- Кожен вузол даних через певні інтервали часу відправляє діагностичні повідомлення вузлу імен (вузли даних докладно будуть розглянуті нижче).
- Логування операцій над файлами в спеціальний журнал вузла імен.

Продуктивність

У квітні 2008 року Hadoop побив світовий рекорд продуктивності в стандартизованому тесті продуктивності з сортування даних - 1 Тбайт був оброблений за 309 сек. на кластері з 910 вузлів.

Архітектурні рішення:

- Принцип «один раз записати - багато разів прочитати» (Write-once and read-many, WORM) повністю звільняє систему від блокувань типу «запис-читання». Позбутися від конфліктів множинного запису проектувальники вирішили, дозволивши запис у файл в один час тільки одному процесу.
- HDFS оптимізований під потокову передачу даних.
- Знизити навантаження на канали передачі даних (а саме ці канали найчастіше є вузьким місцем в розподілених середовищах), а також більш раціонально використовувати місце на жорстких дисках дозволило стиснення даних.
- Реплікація відбувається в асинхронному режимі.
- Зберігання всіх метаданих вузла NameNode в оперативній пам'яті.

Вузли імен

Вузол імен (NameNode) являє собою програмний код, що виконується, в загальному випадку, на виділеній машині примірника HDFS і відповідає за файлові операції, такі як відкриття та закриття файлів, створення і видалення каталогів. Крім того, NameNode відповідає за:

- керування простором імен файлової системи;
- керування доступом з боку зовнішніх клієнтів;

Hadoop містить єдиний вузол типу NameNode. Що породжує вразливість * всього кластера, викликану виходом вузол типу NameNode (одинична точка відмови). HDFS підтримує вторинний вузол імен - Secondary NameNode. Часто це факт є причиною омани, що при відмові первинного вузла імен, його автоматично замінить вторинний вузол імен. Насправді підтримки автоматичного відновлення кластера після відмови первинного вузла NameNode у версії 1.0.0 немає.

Вторинний вузол імен виконує наступні функції:

- копіює образ HDFS (розташований у файлі FsImage) і лог транзакцій операцій з файловими блоками (EditLog) в тимчасову папку;
- застосовує зміни, накопичені в балці транзакцій до образу HDFS;
- записує новий образ FsImage на вузол NameNode, після чого відбувається очищення EditLog.

Вузли даних

Вузол даних (DataNode), як і вузол NameNode, також являє собою програмний код, що виконується, як правило, на виділеній машині примірника HDFS і відповідає за операції рівня файлу, такі як:

запис і читання даних,

виконання команд створення, видалення та реплікації блоків, отриманих від вузла NameNode.

Крім того, вузол DataNode відповідає за:

– періодичну відправку повідомлення про стан (heartbeat-повідомлення);

- обробку запитів на читання і запис, що надходять від клієнтів файлової системи HDFS, так як дані приходять з решти машин кластера до клієнта повз вузла NameNode.

Клієнти HDFS

Клієнти є програмними клієнтами, що працюють з файловою системою. У ролі клієнта може виступати будь-який додаток або користувач, який взаємодіє через спеціальний API з файлової системою HDFS.

Для клієнта HDFS виглядає як звичайна ** файлова система - ієрархія каталогів з вкладеними в них підкаталогами і файлами. Як і у файлових системах загального призначення, клієнту, за наявності достатніх прав, дозволені наступні операції: створення, видалення, перейменування, переміщення. Вищеназвані операції застосовні до каталогів і файлів.

Найбільш істотна відмінність роботи клієнта з файлової системою HDFS від роботи з файловою системою загального призначення - це те, що при створенні файлу клієнт може явно вказати розмір блоку файлу (за замовчуванням 64 Мб) і кількість створюваних реплік (за замовчуванням значення дорівнює 3-му) .

Взаємодія компонентів HDFS

Взаємодія вузлів імен, вузлів даних і клієнтів здійснюється за протоколами, які базуються на протоколі TCP / IP.

Клієнт створює з'єднання через спеціально сконфігурований для взаємодії TCP-порт на цільовому вузлі NameNode. Взаємодії клієнта з вузлом NameNode відбувається по протоколу ClientProtocol. Вузли DataNode взаємодіють з вузлом DataNode, використовуючи протокол DataNode Protocol [6].

І ClientProtocol, і DataNode Protocol «обгорнуті» у Remote Procedure Call (RPC). Вузол NameNode ніколи не ініціалізує виклики RPC - він тільки відповідає на RPC-виклики вузлів DataNode і клієнтів.

Файлові операції та реплікація

Набір допустимих файлових операцій у розподіленої файлової системи HDFS схожий з набором файлових операцій в «локальних» файлових системах за

винятком операції модифікації файлу - модифікація в HDFS не дозволена з причин, пов'язаних з архітектурними особливостями (в тому числі і питаннями продуктивності і блокувань) цієї файлової системи.

За всі файлові операції відповідає вузол NameNode. Операції з конкретними файлами знаходяться в зоні відповідальності вузла DataNode, на якому ці файли знаходяться. Детальніше про функції, що виконуються на вузлах імен і вузлах даних буде описано нижче.

Цікаві архітектурні рішення були знайдені для створення файлів.

Спочатку клієнт кеширує необхідну для запису інформацію десь в тимчасовому (або постійному – вибір клієнта) сховище. Після того, як обсяг інформації досягає передбачуваного клієнтом розміру блоку в HDFS, клієнт відправляє на вузол NameNode запит на створення файлу, опціонально вказавши розмір блоку для створюваного файлу і кількість реплік.

Вузол NameNode відповідає клієнту, відправивши у відповідь ідентифікатор вузла даних і блок призначення, на який буде вестися запис. Також вузол NameNode повідомляє інші вузли DataNode, на які будуть писатися репліки файлового блоку. Після початку передачі файлового потоку вузлу DataNode, що приймає вузол починає автоматичну ретрансляцію файлового блоку на інші вузли репліки. Закінчення запису файлового блоку фіксується в журналі вузла імен.

Всі файлові блоки реплікуються вказане клієнтом при створенні разів. Друга репліка файлового блоку зберігається на іншому вузлі, а третя - на вузлі, розташованому на іншій стійці. Розміщення наступних реплік обчислюється довільно.

Якщо вузол NameNode не приймає від вузла DataNode heartbeat-повідомлень, то вузол імен позначає це вузол DataNode як «померлий» і реплікує дані, що зберігаються на «померлому» вузлі з копій «що залишилися в живих».

Обмеження HDFS

Файлова система HDFS має наступні обмеження:

- вузол імен NameNode є єдиною точкою відмови;
- відсутність повноцінної реплікації Secondary NameNode;
- відсутність можливості дописувати або залишити відкритим для запису файли в HDFS (як наслідок, в plain Hadoop відсутня підтримка оновлюваних і потокових даних);
- відсутність підтримки реляційних моделей даних;
- відсутність інструментів для підтримки посилальної цілісності даних;
- низька безпека даних.

3.4.3. Програмна модель map / reduce

Виконання розподілених завдань на платформі Hadoop відбувається в рамках парадигми map / reduce (програмної моделі).

map / reduce - це парадигма виконання розподілених обчислень для великих обсягів даних.

У загальному випадку, для map / reduce виділяють 2 фази:

map (*f*, c)

- Приймає функцію f і список c. Повертає вихідний список, який є результатом застосування функції f до кожного елементу вхідного списку c.

reduce (*f*, c)

- Приймає функцію f і список c. Повертає об'єкт, утворений через згортку колекції c через функцію f.

Програмна модель map / reduce була запозичена з функціонального програмування, хоча в реалізації Hadoop і має деякі семантичні відмінності від прототипу в функціональних мовах.

Як і у функціональних мовах, при використанні програмної моделі map / reduce:

- вхідні дані не змінюються;
- розробник кодує, що потрібно зробити, а не як потрібно зробити.

На даний момент широко відомі наступні програмні реалізації моделі map / reduce:

- Google MapReduce - закрита реалізація від Google на C + +;
- CouchDB і MongoDB - реалізації для NoSQL баз даних;

- Hadoop MapReduce - відкрита реалізація на Java для Apache Hadoop.

Роботу Hadoop MapReduce можна умовно поділити на наступні етапи:

Input read

Вхідні дані діляться на блоки даних зумовленого розміру (від 16 Мб до 128 Мб) - сплити (від англ. Split). MapReduce Framework закріплює за кожною функцією Map певний сплит.

Map

Кожна функція Map отримує на вхід список пар «ключ / значення» <k,v>, обробляє їх і на виході отримує нуль або більше пар <k',v'>, які є проміжним результатом.

Partition / Combine

Метою етапу partition (поділ) є розподіл проміжних результатів, отриманих на етапі map, по reduce-завданнях.

Основна мета етапу partition - це балансування навантаження. Некоректно реалізована функція partition може призвести до нерівномірного розподілу даних між reduce-вузлами.

Copy / Sort / Merge

На цьому етапі відбувається:

Copy: копіювання результатів, отриманих у результаті роботи функцій map і combine (якщо така була визначена), з map-вузлів на reduce-вузли.

Sort: сортування, угруповання по ключу k отриманих в результаті операції copy проміжних значень на reduce-вузлі.

Merge: «злиття» даних, отриманих від різних вузлів, для операції згортки.

Reduce

Framework викликає функцію reduce для кожного унікального ключа k 'в відсортованому списку значень.

Output write

Результати, отримані на етапі reduce, записуються в вихідний потік (у загальному випадку, файлові блоки в HDFS). Кожен reduce-вузол пише у

власний вихідний потік.

3.5. Вибір мови обробки даних

Для аналіза даних і статистичної обробки існує багато бібліотек та додатків в різних мовах програмування. Перед початком експеременту, я зробив огляд можливостей різних мов для обробки статистичних даних:

- **R:** зручна мова, створена для виконання статистичного аналізу, має багато пакетів, з уже реалізованими алгоритмами для машинного навчання. Але за рахунок того, що вона дуже повільна, і не може працювати з розподіленими даними, обробляти на ній більш-менш великі обсяги даних дуже довго.
- **Java:** універсальна мова, яка працює з розподіленими даними і Hadoop, але майже немає бібіліотек з реалізованими алгоритмами машинного навчання.
- **Python:** достатньо швидка скриптова мова, що має декілька потужних бібліотек для статистичного аналізу і вміє працювати з розподіленими даними і Hadoop.

Зважаючи на приведені вище аргументи, мною біло вибрано Python для реалізації експерементальних досліджень. Розглянемо детальніше можливості python для статистичної обробки даних:

- numpy — дозволяє створювати складні структури даних, що в свою чергу дозволяє ефективно їх використовувати для алгоритмів аналізу даних.
- Scikit-learn — має багато реалізованих алгоритмів для аналізу даних, що можна умовно розділити на алгоритми класифікації, регресії, кластеризації та зменшення розмірності даних, детвльніше можна побачити на рисунку 3.5.1.

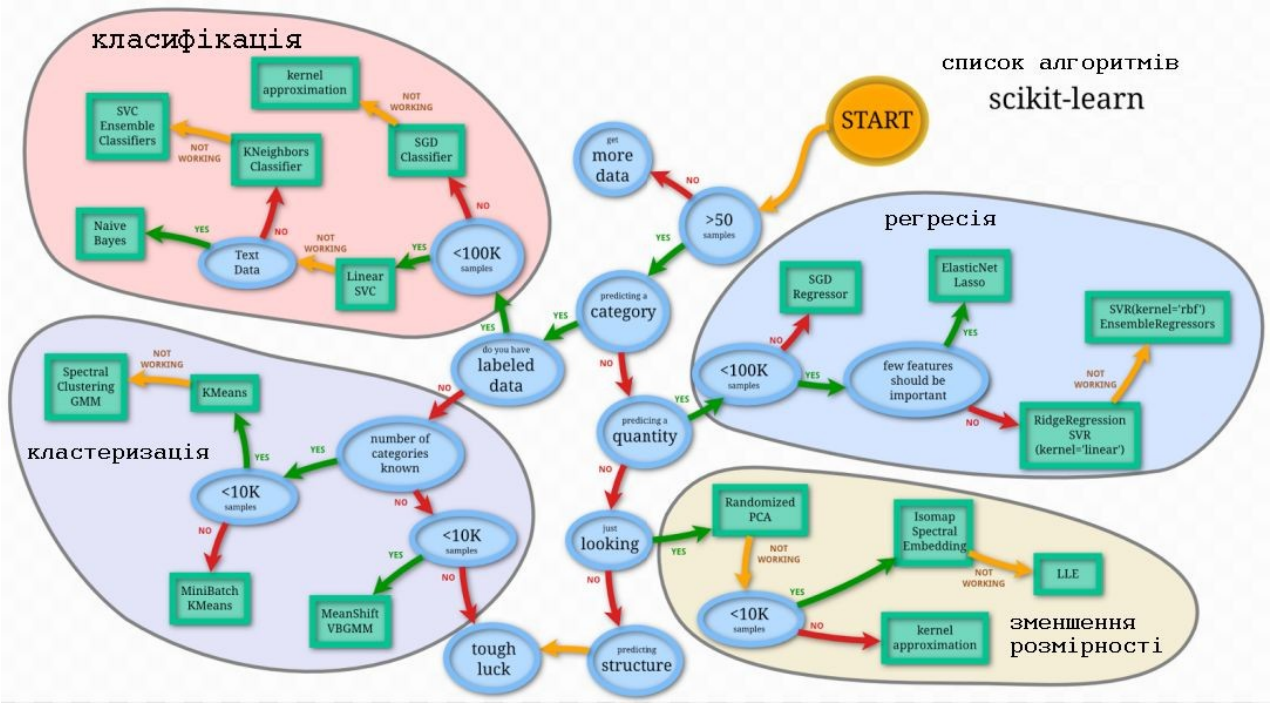

Рис. 3.5.1. - класифікація реалізованих алгоритмів бібліотеки scikit-learn

Отже, у зв'язку з поставленими задачами, при виконанні дипломної роботи, мною було вирішено, що доцільно розгорнути середовище керування проектом (Redmine) репозиторій (Bitbucket Git) для керування версіями, Hadoop, на кластері, що складався з 4х нод для керування даними і використовувати Python як мову для написання програмної частини.

4. ЕКСПЕРЕМЕНТАЛЬНІ НАБОРИ ДАНИХ

Для проведення експеременту по перевірці гіпотези використання соціальних мереж в РС, висунутої в розділі 3, необхідні тренувальні та тестові дані. Для цієї цілі мною було вибрано відкриті набори даних, що є в вільному доступі. Перший з цих наборів — анонімізований набір даних користувачів з соціальних мереж, набір різних подій, що могли зацікавити користувача, і відношення користувачів до подій (у вигляді зацікавлений/незацікавлений). В даному наборі також є тестовий набір користувачів, для яких треба передбачити їх зацікавленість подіями.

Другий набір даних — датасет Epinions. Це рейтинги, що зібрані з оцінок фільмів, автомобілів та інших речей, користувачами сайту Epinions.

Детальніше обидва набори даних описані нижче.

4.1. Набір даних Kaggle Event Recommendation Engine Challenge

Дані взяті для тестування з відкритого датасету, який був представлений на Event Recommendation Engine Challenge на порталі Kaggle.

Датасет складається з шести файлів: train.csv, test.csv, users.csv, user_friends.csv, events.csv і event_attendees.csv

train.csv має шість стовпчиків: *user, event, invited, timestamp, interested, і not_interested*.

test.csv містить ті ж стовпчики, як **train.csv**, крім *interested, і not_interested*. . Кожен рядок файла відповідає певній події.

- *event* — це id, що визначає подію.
- *user* — id, що визначає користувача.
- *invited* — бінарна змінна, що визначає, чи був запрошений користувач на подію, чи ні.
- *timestamp* - це час в форматі ISO-8601 UTC, який показує приблизний час

(+/- 2 години) , коли користувач побачив запрошення на подію.

- *interested* — бінарна змінна, що показує чи користувач зацікавився даною подією (натиснув що “приєднається” до події, або “можливо приєднається”); 1 — користувач зацікавлений подією, 0 — не зацікавлений.
- *not_interested* — аналогчно, бінарна змінна, яка показує, чи натискав користувач кнопку “не йти” відносно даної зустрічі; 1 — якщо натискав, і 0 — якщо ні.

Якщо користувач бачив подію — відповідні значення interested і not_interested: 0,0

users.csv містить демографічні дані про деяких з користувачів (у тому числі всіх користувачів, що входять в **train** і **test** файли), і має такі стовпці: user_id, locale, birthyear, gender, joinedAt, location, and timezone.

- *user_id* - це ідентифікатор користувача в системі.
- *locale* - рядок, що представляє мову користувача
- *birthyear* - 4-значне ціле число, що представляє рік, коли користувач народився.
- *gender* - чоловік або жінка, в залежності від статі користувача.
- *joinedAt* - ISO-8601 час по UTC - рядок, що представляє коли користувач вперше побачивн запрошення на подію додаток.
- *location* - рядок, що представляє розташування користувача (якщо відомо).
- *timezone* - ціле число, що представляє зсув часу користувача по UTC (у хвилинах).
- *likes* — список сторінок, які сподобались користувачу за останній час

user_friends.csv містить соціальні дані про користувача, в двох колонках: користувач та друзі користувача. Друзі — це список друзів користувача, розділених комою.

events.csv містить дані про події, і має 9 стовпців: event_id, user_id, start_time, city, state, zip, country, lat and lng.

- **event_id** це ідентифікатор події
- **user_id** - ідентифікатор користувача, який створив подію.
- **city, state, zip, country** представляють більш детальну інформацію про розташування місця (якщо відомий).
- **lat and lng** - представляють координати широти і довготи на місці з округленням до трьох знаків після коми.
- **start_time** - ISO-8601 час по UTC, коли рядок, що показує, коли подія почнеться.
- **event_type** — категорія події, що проводиться

event_attendees.csv містить інформацію про те, які користувачі взяли участь у яких подіях, і має такі стовпці: event_id, yes, maybe, invited, і no.

- *event_id* - ідентифікує подію.
- *yes, maybe, invited*, і *no* — списки користувачів, розділені комою, які прийняли своє рішення що до події.

4.2. Датасет Epinions

Epinions [18] - це сайт, де збираються думки споживачів, тут користувачі можуть рецензувати і оцінювати різні речі, в тому числі фільми, автомобілі, книги, програмне забезпечення тощо. Набір даних складається з 49290 користувачів, 664824 / Відгуки рейтингів, і 139 738 різних предметів , для яких складено рейтинги. Рейтинги визначені за п'ятизіркового шкалою, де одина і дві зірки представляють негативні рейтинги, три зірки представляють рейтинги амбівалентності і чотири, п'ять зірок представляють позитивні оцінки. Користувачі також висловлюють довіру до набору користувачів, чиї огляди / рейтинги виявилися цінним. Загальна кількість оцынок довіри 487181.

Користувач отримує рекомендацію, тільки якщо вона пов'язана з іншими користувачами через довірчі відносини. Після фільтрації ізольованих користувачів, набір даних має 73960 користувачів і 589 714 рейтингів.
Кожен користувач має такі поля, що його характеризують: *user_id*, *location*, *gender, birthyear, likes*, що ідентичні за заначенням полям датасету Kaggle.

4.3. Моделювання профілю користувача

Ці набори даних, що описані вище, були вибрані, так як вони практично ідентичні з даними, які можна отримати про користувача через API соціальних мереж. Далі розглянуто API Facebook, щоб показати, що доступні через нього дані практично такі самі, як і в датасетах Kaggle та Epinions. Тобто, можна вважати що вибрані набори даних моделють справжніх користувачів, які могли б користуватися такою РС.
За допомогою API Facebook, про користувача можливо зібрати такі дані: ім'я, прізвище, мову що користувач використовує в Facebook, стать, електронну пошту, релігію, день народження, місто, де користувач знаходиться, його освіту та вподобання, які заповнив користувач.
До вподобань відносяться такі категорії як музика, фільми, книжки, відомі особи, тощо. Ці поля можуть бути як заповнені, так і ні. Саме тому при моделюванні профілю користувача доводиться структуризувати дані з його профілю, бо одні поля можуть бути як заповненими, так і пустими, тобто відсутня цілісність даних.
Також, за допомогою API Facebook, можна дізнатися яким сторінкам користувач ставив “like”. Ці сторінки мають не тільки назву, а і тип, що також дозволяє визначати вподобання користувача.
Повні можливості API Facebook показані на рисунку 4.3.2

Рис. 4.3.1 – Профіль користувача Facebook та його останні “лайки”, отримані через API

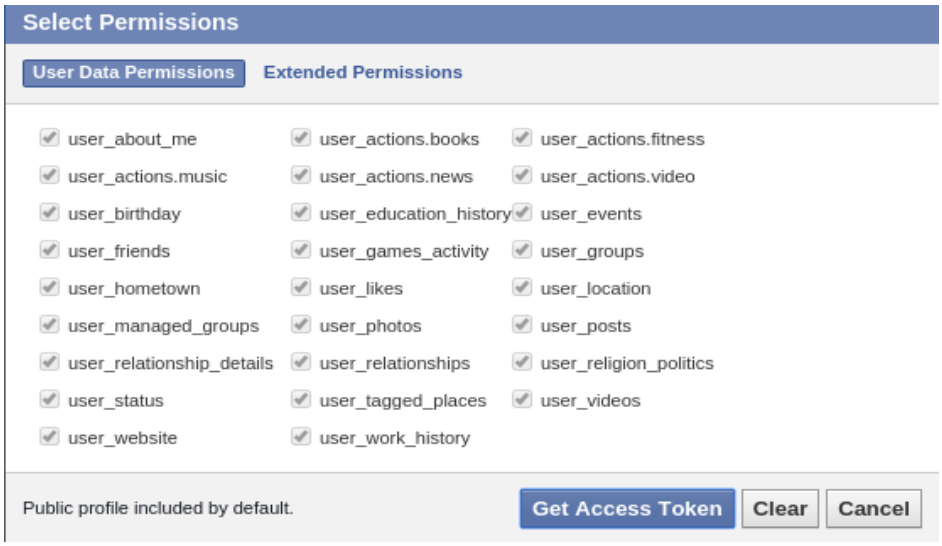

Рис. 4.3.2 – Дані, які можна отримати за допомогою Facebook API

Таким чином, в ході дослідження гіпотеза перевіряється на 2х різних наборах даних, що дозволить більш повно дослідити, як дані з профілю користувача впливають на точність рекомендацій для даних з рейтенговим оцінюванням та без нього.

5. ЕКСПЕРЕМЕНТАЛЬНА ПЕРЕВІРКА ГІПОТЕЗИ

Для доведення гіпотези, висунутої в розділі 3, на зазначеній інфраструктурі було побудовано РС для обробки описаних вище наборів даних. За основу бралася модель SocialMF[39].

5.1. Тренування моделі

Тренування з оцінками з кожної категорії

Використовуючи нормалізовану матрицю довіри $S^{(c)}$ модель тренується, створивши іншу матрицю, розкладанням даної, по категоріям с. Для кожного профілю користувача, який входить в якусь категорію, отримується окремий профіль $Q^{(c)}$ і окремий профіль $P^{(c)}$ для кожної речі, яка відноситься в категорію с. Це модифікована модель SocialMF, в якій використовується інша функція для врахування соціальних категорій друзів:

$$\begin{aligned}
&\mathcal{L}^{(c)}(R^{(c)}, Q^{(c)}, P^{(c)}, S^{(c)*}) = \\
&\frac{1}{2} \sum_{(u,i)\text{obs.}} \left(R^{(c)}_{u,i} - \hat{R}^{(c)}_{u,i}\right)^2 \\
+ \quad &\frac{\beta}{2} \sum_{\text{all } u} \left((Q^{(c)}_u - \sum_v S^{(c)*}_{u,v} Q^{(c)}_v)(Q^{(c)}_u - \sum_v S^{(c)*}_{u,v} Q^{(c)}_v)^\top \right) \\
+ \quad &\frac{\lambda}{2} \left(||P^{(c)}||^2_F + ||Q^{(c)}||^2_F \right),
\end{aligned} \tag{5.1.1}$$

де $R^{(c)}_{u,i}$ в категорії c; $\hat{R}^{(c)}_{u,i}$ це передбачена оцінка речі *i* в категорії с:

$$\hat{R}^{(c)}_{u,i} = r^{(c)}_m + Q^{(c)}_u {P^{(c)}_i}^\top \tag{5.1.2}$$

де $r^{(c)}_m$ - середнє значення відомих оцінок в категорії с. Сума містить всі відомі пари оцінок «користувач — річ» в певній категорії с. Для спрощення обчислень,

рівняння може бути спрощено за рахунок градієнта, аналогічно до моделі SocialMF:

$$\begin{aligned}
&\frac{\partial \mathcal{L}^{(c)}}{\partial Q_u^{(c)}} = \\
&\sum_{i:cat(i)=c} I_{u,i}^{R^{(c)}} \left(r_m^{(c)} + Q_u^{(c)} P_i^{(c)^T} - R_{u,i}^{(c)} \right) P_i^{(c)} + \lambda Q_u^{(c)} \\
+ \quad & \beta \left(Q_u^{(c)} - \sum_{v \in \mathcal{C}_u^{(c)}} S_{u,v}^{(c)*} Q_v^{(c)} \right) \\
- \quad & \beta \sum_{v:u \in \mathcal{C}_v^{(c)}} S_{v,u}^{(c)*} \left(Q_v^{(c)} - \sum_{w \in \mathcal{C}_v^{(c)}} S_{v,w}^{(c)*} Q_w^{(c)} \right),
\end{aligned} \tag{5.1.3}$$

де $I_{u,i}^{R^{(c)}}$ - це індикатор того, що користувач *u* оцінив річ *i* в категорії c. Тобто якщо користувач здійснював оцінку, то значення буде 1, в іншому випадку — 0.

Отже, ця модель може бути застосована для рекомендації в межах однієї категорії c.

Навчання моделі, використовуючи оцінки з усіх категорій

Цю ж модель можна використовувати і для декількох категорій. Єдина відмінність — необхідно сумувати пари «користувача-оцінка» по всіх категоріях, спочатку окремо тренуючи для кожної. Тоді модель буде мати вигляд:

$$\frac{1}{2} \sum_{(u,i)\text{obs.}} \left(R_{u,i} - \hat{R}_{u,i} \right)^2, \tag{5.1.4}$$

Далі, для врахування даних, що беруться з профілю користувача, таких як: мову яку використовує користувач (мова що він нею користується в Facebook), стать, релігію, день народження, місто, де користувач знаходиться, його освіту та вподобання, які заповнив користувач, в формулу (5.1.3) добавляється сума

вагових коефіцієнтів W, що розраховуються наступним чином:

$$W = \sum_{N}^{i=1} (a_i * b_i) = \sum_{N}^{i=1} (a_i * d_i * f_i) \quad (5.1.5)$$

де $a_i = R_{u,i}^{(c)}$, тобто це передбачена оцінка речі(події) *i* в категорії c, а b_i — коефіцієнт цієї події, що складається з вагового коефіцієнта d_i, який має значення від 0 до 1, і індикатора присутності f_i, що показує чи присутня дана категорія c в інтересах користувача, i = 1, якщо присутня і 0, якщо відсутня

Ваговий коефіцієнт d_i виставляється розробником PC (тобто, фактично розробник PC вирішує, наскільки важливий той чи інший фактор).

W сумується по всім категоріям і нормалізується до 1, тоді формула (5.1.4) модифікується таким чином:

$$W \frac{1}{2} \sum_{(u,i)obs.} (R_{ui} R'_{ui})^2 \quad (5.1.6)$$

Для кожної моделі окремо визначаються коефіцієнти, що в ній враховуються. Наприклад, при роботі з датасетом Epinions, місце розташування користувача не бралося до уваги, а при роботі з датасетом Kaggle, навпаки, воно мало велике значення.

Для датасету Epinions я використовував такі вагові коефіцієнти: останні “лайки” користувача, його вподобання (з профілю), та стать.

Для датасету Kaggle я використовував вагові коефіцієнти: місцезнаходжкння користувача, його “лайки”, вподобання, та стать.

5.2. Оцінка ефективності

Для експеременту був використаний 10-разовий клас валідації (10-fold cross validation). Кожен раз 80% даних використовувалось для тренування моделі, а 20% що залишилися, використовувалися як тестові дані. Для оцінки точності

рекомендацій використовувалась середньоквадратична похибка (Root Mean Square Error, RMSE) і медіана абсолютної похибки (Mean Absolute Error (MAE), так як це найбільш популярні метрики для оцінювання якості статистичних моделей.

$$RMSE = \sqrt{\frac{\sum_{(u,i)\in\mathcal{R}_{test}} (R_{u,i} - \hat{R}_{u,i})^2}{|\mathcal{R}_{test}|}}, \quad (5.2.1)$$

$$MAE = \frac{\sum_{(u,i)\in\mathcal{R}_{test}} |R_{u,i} - \hat{R}_{u,i}|}{|\mathcal{R}_{test}|}. \quad (5.2.2)$$

де R*test* - дані з тестового набору.

5.3. Результати

В таблиці 5.3.1 приведено результати для дтасету Kaggle. На малюнку 5.3.1 візуалізовано результати з таблиці 5.3.1.

Category	SocialMF	+категоризац	+дані з профі	+категоризац
Media/News/P	**0.863**	0.874	**0.862**	**0.76**
Community	0.812	0.809	**0.797**	**0.769**
Music	0.794	0.798	**0.788**	**0.748**
Author	0.786	**0.781**	**0.781**	**0.75**
Sports	0.777	0.782	**0.773**	**0.75**
Local/Travel	0.955	**0.947**	0.953	**0.937**
Software	0.841	0.842	**0.831**	**0.825**
Politician	0.766	0.766	0.769	**0.739**
Cars	0.832	**0.828**	**0.829**	**0.785**
Business	0.813	0.812	**0.802**	0.668

Таблиця 5.3.1 – Результати тестування на Kaggle датасеті

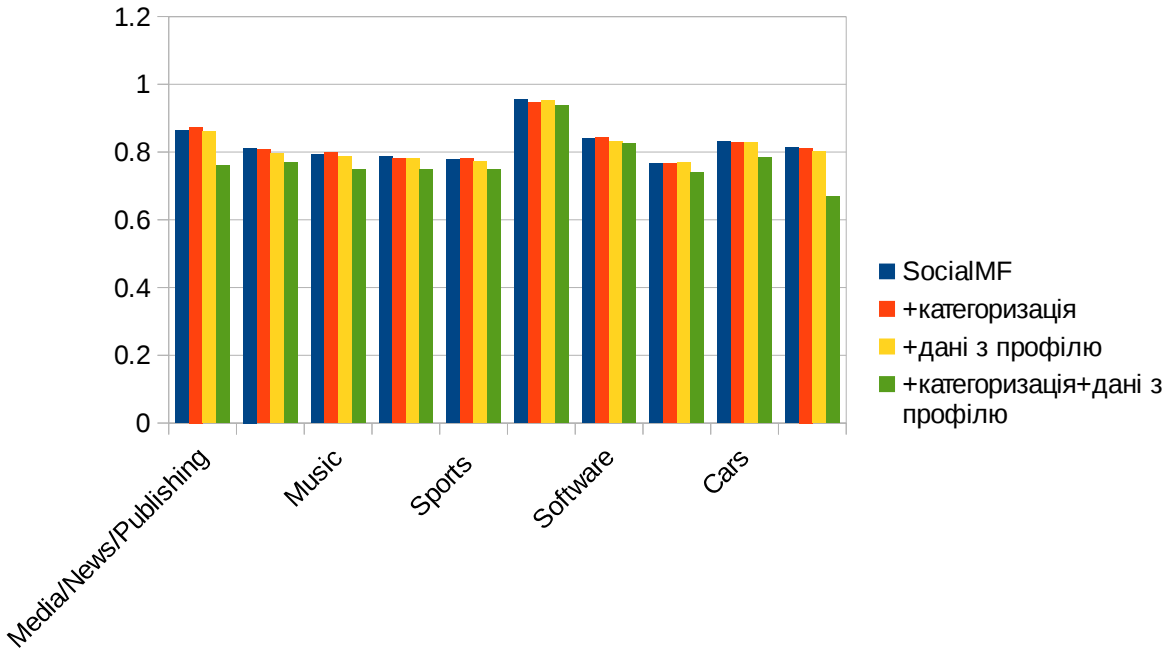

Рис. 5.3.1 – Результати тестування на Kaggle датасеті

Category	SocialMF	+категоризація	+дані з профілю	+категоризація+дані з профілю
Media/News/P	0.98	1	**1.01**	**0.96**
Films	1.102	1.05	**1.06**	**1.045**
Music	0.779	0.982	**0.973**	**0.953**
Books	0.855	**0.774**	**0.767**	**0.735**
Sports	1	0.995	**0.988**	**0.964**
Electronics	1.101	1.075	**1.061**	**1.056**
Pets	0.781	**0.759**	0.763	**0.75**
Cars	1.076	1.08	**1.073**	**1.03**
Games	1.052	1.051	**1.042**	**1.036**

Таблиця 5.3.2 – Результати тестування на датасеті Epinions

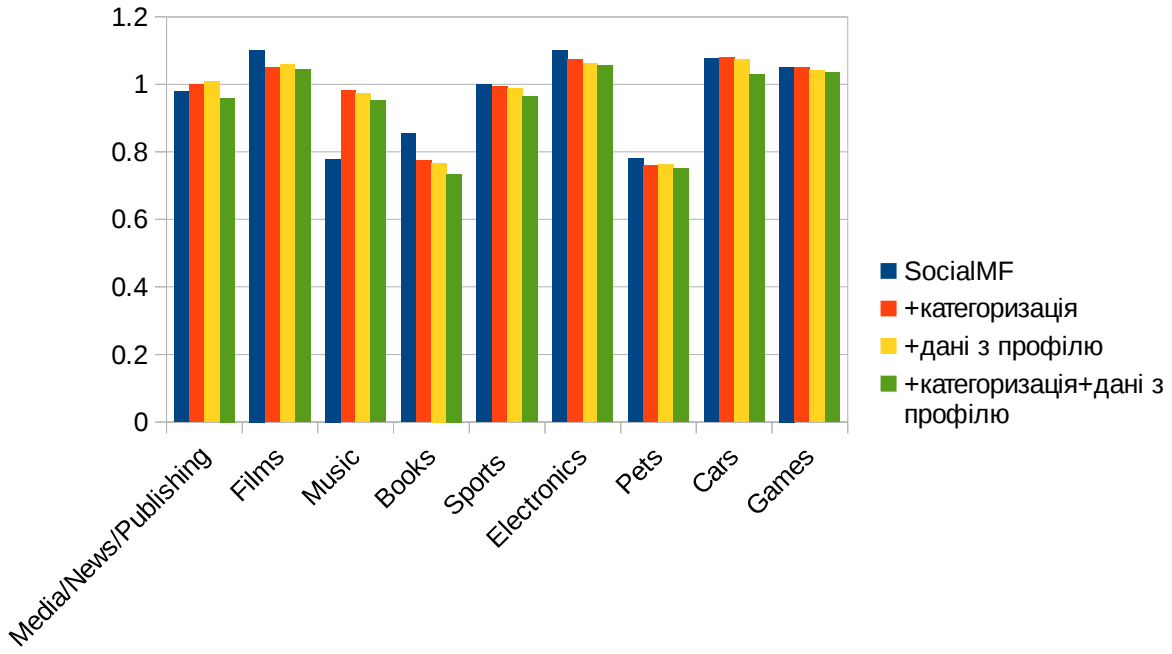

Рис. 5.3.2 – Результати тестування на датасеті Epinions

В таблиці 5.3.2 приведено результати для дтасету Epinions. На малюнку 5.3.2 візуалізовано результати з таблиці 5.3.2.

З приведених результатів видно, що запропонований метод використання даних з профілю користувача та категоризації друзів очікувано зменьшив похибку рекомендації (RMSE). Також з результатів видно, що рекомендації більш точні на датасеті Kaggle, що очікувано, враховуючи велике значення вагового коефіцієнта місця знаходження користувача.

6. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

Даний розділ у дипломній роботі обумовлений вимогами законодавства України щодо правильних та комфортних умов праці кожного громадянина. Рекомендаційна система, яку ми досліджуємо та розробляємо у дипломній роботі, передбачає також і роботу людей з нею, а отже необхідно перейматись їхньою безпекою. Ніяких специфічних особливостей, шкідливих для людини, у роботі з даною системою немає.

6.1. Аналіз умов праці на робочому місці

Аналізувати наші вихідні дані будемо за такими факторами:

- освітлення;
- шум;
- мікроклімат;
- електромагнітні випромінювання;
- електробезпека;
- пожежна безпека.

Характеристика приміщення і робочого місця:

Нижче схематично зображено кімнату, що необхідна для роботи розробників:

- 3,6м завширшки;
- 5,4м довжиною;
- 2,8м висотою.

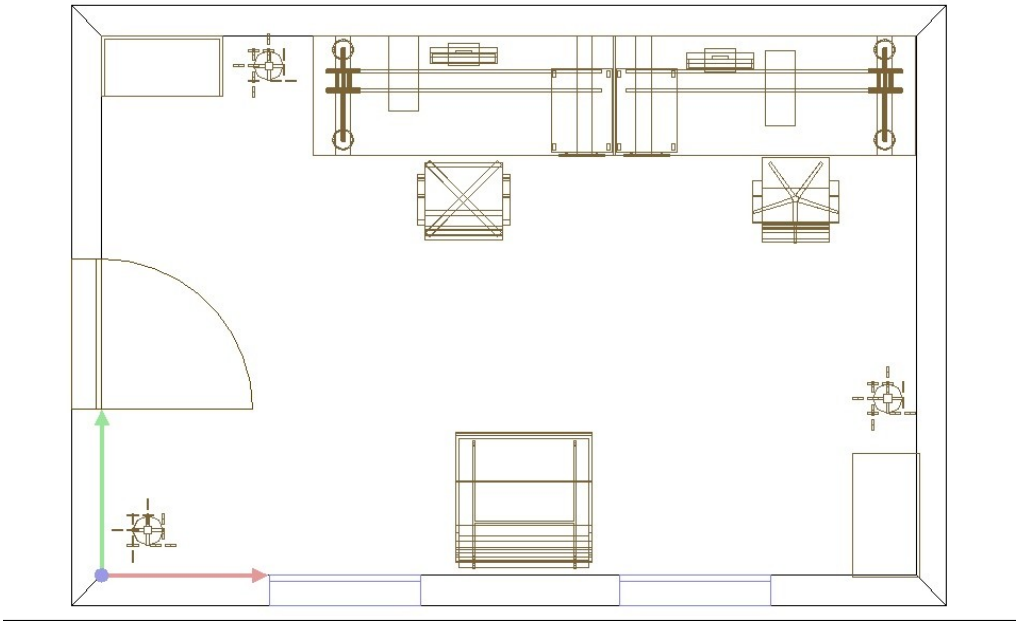

Рисунок 6.1.1 – План приміщення

Загальна площа кімнати складає: $S_{заг} = 3,6 \times 5,4 \approx 20 м^2$.

Загальний об’єм кімнати: $V_{заг} = 3,6 \times 5,4 \times 2,8 \approx 56 м^3$.

Враховуючи, що кімната розрахована на 2 чоловік, то на 1 працівника припадає площі в $10 м^2$ та об’єму в $25 м^3$. За нормативними даними, що наведено в «Правилах охорони праці під час експлуатації ЕОМ» та ДСанПіН 3.3.2-007-98[5] «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами ЕОМ», обсяг простору для однієї людини в лабораторії з ЕОМ повинен бути не менше $V_1=20 м^3$, а площа не менше $S_1=6 м^2$. Оскільки фактичні параметри більші ніж нормативні, то приміщення задовольняє перерахованим вимогам.

№	Параметри	Норма	Отримані результати
1	Площа	не менше $6 м^2$	$10 м^2$
2	Об’єм	не менше $20 м^3$	$28 м^3$

Таблиця 6.1.1 – Фактичні дані та нормативні вимоги

№	Найменування	Характеристика	Кількість, шт.
1	Стіл комп’ютерний		2
2	Крісло функціональне		2
3	Комп’ютерний монітор	19" TFT- монітор Asus PW201F Wide Screen, 5ms	2
4	Системний блок	Celeron D347 4.0GHz i945PL DDR II 2048 Mb PC2-5600 667MHz HD2400 PROSATA 200Gb FDD/DVD-RW	2
5	Комп’ютерна мишка	X5-60MD A4Tech 5V 100mA	2
6	Комп’ютерна клавіатура	240R Logitech OEM W4511-8F	2

Таблиця 6.1.2 – Перелік та опис устаткування

Робочі столи мають розміри 700x1400x800, що задовольняє рекомендованим вимогам. Комп’ютерні екрани розташовані на відстані приблизно в 60см від очей працівників (при нормативних даних 48см для 19″ екрану). Висота рухомого крісла та рівень нахилу осанки регулюються відповідно до побажань працюючого.

У приміщенні знаходяться два столи та два стільця, розміри яких приведені в таблиці 6.1.3.

Параметр	Норматив, мм	Місце 1, мм	Місце 2, мм
Довжина столу	600-1400	1400	1400
Ширина столу	800-1000	800	800
Висота столу	680-800	700	700
Ширина простору для ніг	Не менш 500	700	700
Висота простору для ніг	Не менш 600	600	600
Висота стільця	400-500	400-500	400-500
Відстань від очей до монітору	600-800	600	600

Таблиця 6.1.3 – Параметри робочого місця

Кімнату обладнано сучасною системою вентиляції, окрім цього майже щодня відкриваються вікна на провітрювання. Проводиться кожного дня вологе прибирання, під час якого вимикаються всі електричні прилади, окрім сервера (сервер повинен працювати постійно, планова чистка проводиться раз на

місяць).

У приміщенні виконуються роботи переважно сидячи та не потребуючі систематичної фізичної напруги при роботі з комп’ютером . Згідно з цим, дані роботи можна віднести до категорії 1б у відповідності до ДСН 3.3.6-042-99[6], тобто робіт, які виконуються сидячи, стоячи чи пов’язані з ходьбою, і вимагають деякого фізичного навантаження. Енерговитрати організму людини при такому виді робіт складають до 150 ккал/год.

6.2. Мікроклімат

Повітря у робочому приміщені характеризується наступними параметрами:

- температура повітря, C^o;
- відносна вологість повітря ,%;
- швидкість руху повітря, м/с.

Визначимо норми та фактичні значення мікроклімату досліджуваного приміщення згідно ДСанПіН 3.3.2-007-98[5].

№	Пора року	Категорія робіт	Температура повітря, C^o		Відносна вологість повітря, %		Швидкість руху повітря, м/с	
			Норма	Факт	Норма	Факт	Норма	Факт
1	Холодна	1а легка	20-24	18-23	40-60	40-50	0,1	0,1
2	Тепла	1а легка	23-25	20-28*	40-60	50-60	0,1	0,1

Таблиця 6.2.1 – Нормативні та фактичні значення мікроклімату

28* – без кондиціонера

Вентиляція в приміщенні штучна. Розрахуємо коефіцієнт холодовиробництва:

$$W = A*0.035 + E1*0.2 + E2*0.15 + F*A*0.319 \quad (6.2.1)$$

де А – об’єм приміщення, $м^2$

E1 – кількість комп’ютерів

E2 – кількість людей

F – якщо приміщення знаходиться на останньому поверсі будинку під горизонтальною покрівлею.

Так як у приміщенні відсутня природна вентиляція, то повинен бути встановлений канальний кондиціонер з швидкістю подачі повітря:

$$V = L * n \qquad (6.2..2)$$

де L = 20 м^3/год,

n – кількість робочих місць.

V = 40 м^3/год

В холодний період приміщення обігрівається внутрішньою системою опалення за допомогою батарей (водний носій тепла). Вони нагріваються до температури близько 60 ^{o}C.

Всі значення відповідають вимогам ДСН 3.3.6.042-99[6].

Мова вже йшла про вологе прибирання в кімнаті майже щодня, таким чином надлишкової іонізації повітря вдається уникати. Запиленість повітря спричинюється наступними факторами: працівники відділу, папір, зовнішнє середовище. Допустима норма має бути до 0,15 мг/м^3. Паперової роботи в приміщені дуже мало, до того ж дві людини в приміщені суттєвої ролі в запиленні не відіграють.

6.3. Природне та штучне освітлення

В кімнаті є два вікна через які в світлу частину доби потрапляє достатня кількість світла для роботи.

Зорові умови праці при штучному освітленні характеризуються значенням освітленості, показником дискомфорту, коефіцієнтом пульсації освітленості.

В приміщенні, що аналізується, використовується система загального рівномірного освітлення. В якості джерела світла використовуються

люмінесцентні лампи GE LT18 марки 1xQT12 50W у кількості 6 штук і розміщені у шести світильниках, що розташовані на стелі в два ряди (рис. 6.3.1).

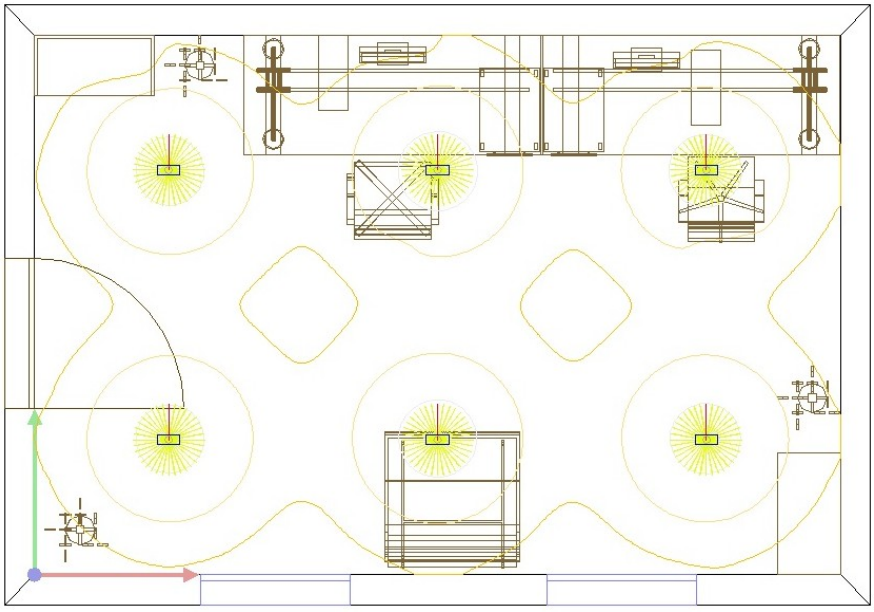

Рисунок 6.3.1 – Розміщення ламп у приміщенні.

Нормою для даного виду діяльності є освітленість робочого місця не менше Ен = 300 лк.

Для розрахунку освітленості використовувалась програма DiaLux 4.3, що являє собою набір інструментів для планування освітлення приміщень.

Мінімальна освітленість на робочому місці - 23 люкс. Максимальна освітленість – 546 люкс. Середня освітленість – 227 люкс.

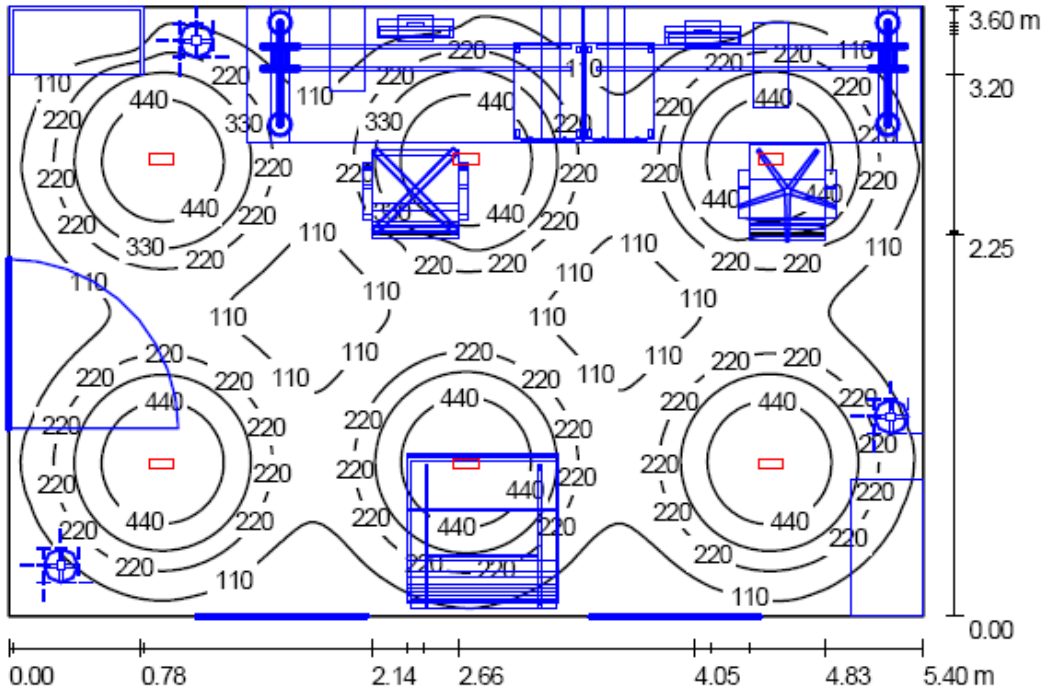

Рисунок 6.3.2 – Діаграма освітленості

Высота помещения: 2.800 m, Монтажная высота: 2.800 m, Показатель техсодержания: 0.80 — Значения в Lux, Масштаб 1:47

Поверхность	ρ [%]	E_{cp} [lx]	E_{min} [lx]	E_{max} [lx]	E_{min} / E_{cp}
Рабочая плоскость	/	227	23	546	0.10
Полы	20	150	4.69	277	0.03
Потолок	70	35	24	45	0.68
Стенки (4)	50	47	4.06	93	/

Рабочая плоскость:

Высота:	0.850 m
Растр:	128 x 128 Точки
Краевая зона:	0.000 m

Ведомость светильников

№	Шт.	Обозначение (Поправочный коэффициент)	Φ [lm]	P [W]
1	6	DIAL 18 HALOSPOT® EL-F 50 W/KLR (1.000)	1000	50.0
		Всего:	6000	300.0

Удельная подсоединенная мощность: 15.43 W/m² = 6.80 W/m²/100 lx (Поверхность основания: 19.44 m²)

Рисунок 6.3.4 – Освітленість задовольняє зазначеним нормам

6.4. Шуми

Поряд з розглядуваним приміщенням немає підприємств, заводів та інших джерел шуму. Автомобільна дорога знаходиться з протилежної сторони будинку.

Джерелом шуму є робочі станції – персональні комп’ютери. А саме – вентилятори блоків живлення, охолоджуючі прилади процесорів і різних швидко-нагрівних апаратних компонентів системи. Також деякий шум створює робота жорстких дискових накопичувачів.

Рівень звукового тиску в приміщенні розрахуємо по формулі:

$$Lекв = 10\lg\left(\frac{1}{T}\sum t_i * 10^{0.1L_i}\right) \quad (6.4.1)$$

T – Загальний час дії системи. В нашому випадку дорівнює 8 годинам.

t_i – Час дії рівня L_i

L_i – рівень звуку і-го елементу

Обладнання, що необхідно враховувати при розрахунку Lекв вказано в таблиці 7.5.

Джерело шуму	Рівень звука L_i, дБА	Час дії звука t_i, г	Кількість n, шт
Блок живлення комп’ютера	42	8	2
Вентилятор процесора	26	8	2
Жорсткий диск HDD SATA II Samsung 200Gb 8Mb	27	8	2
Кондиціонер Mitsubishi SRK28HG/SRC28HG	35	8	1

Таблиця 6.4.1 – Обладнання

Разом: Lекв ≈ 45,6 дБА

У ДСН 3.3.6.037-99[7] встановлюється норматив не більше 50 дБА. Наші

показники є допустимі і задовольняють встановленим нормам.

6.5. Електробезпека

Електробезпека – це система організаційних та технічних заходів і засобів, що забезпечують захист людей від шкідливого та небезпечного впливу електричного струму, електричної дуги, електромагнітного поля і статичної електрики.

Споживачі електроенергії в приміщенні:

- освітлювальні прилади;
- обчислювальна техніка (комп’ютер).

В приміщенні використовується однофазна мережа електропостачання з напругою 220 В та частотою 50 Гц. Проводка схована. Вимикачі і розетки захищені корпусами, що відповідає правилам безпеки. .

Електрична щитова неподалік кімнати, що полегшує вимикання струму в разі крайньої потреби. Джерела світла знаходяться на висоті 3,1 метрів над землею, при мінімальний 2,5м. Робочі столи не є металічними. Отже, в приміщенні відсутні фактори підвищеної небезпеки, тому кабінет можна віднести до категорії приміщень без підвищеної електронебезпеки.

Провід марки ВВП-1 3×1,5 з мідними струмонепровідними жилами ізоляції з ПВХ пластика. Плоский, з роздільною основою.

Додаткових заходів щодо забезпечення електробезпеки кабінету не потрібно. Всі працівники приміщення мають бути ознайомлені з правилами техніки безпеки.

6.6. Безпека в надзвичайних ситуаціях

6.6.1. Технічні рішення системи запобігання пожежі

Протипожежні заходи базуються на вимогах ДБН В.1.1.– 7– 2002[8] щодо

виключення джерела загоряння. Якщо це джерело не може бути ізольованим за умовами технологічного процесу, то об'єкт (приміщення, устаткування) необхідно забезпечити надійною системою протипожежного захисту.

До заходів зниження наслідків пожежі належать:

- установлення вогнеперешкоджувачів;
- обмеження маси небезпечних речовин при зберіганні та в технологічних апаратах;
- водяне зрошення технологічних апаратів;
- винесення пожежонебезпечного обладнання до ізольованих приміщень;
- установлення в технологічному обладнанні швидкодіючих відмикаючих пристроїв;
- обмеження розповсюдження пожежі за допомогою протипожежних відстаней і перешкод;
- застосування вогнезахисних фарб та покриттів;
- захист технологічних процесів, обладнання та окремих приміщень установками пожежогасіння;
- застосування пожежної сигналізації;
- навчання персоналу способам ліквідації аварій та діям у разі пожежі;
- створення умов для найшвидшого введення в дію підрозділів пожежної охорони шляхом улаштування під'їзних шляхів, пожежних водоймищ та зовнішнього протипожежного водогону.

Технологічне обладнання за нормальних режимів роботи повинно бути пожежобезпечним, а на випадок несправностей та аварій передбачаються захисні заходи, які обмежують масштаб та наслідки пожежі.

6.6.2. Профілактика пожежі

Пожежна профілактика являє собою комплекс організаційних і технічних заходів, спрямованих на забезпечення безпеки людей, на запобіганні

пожежі, обмеження його поширення, а також створення умов для успішного гасіння пожежі. Для профілактики пожежі надзвичайно важлива правильна оцінка пожежонебезпеки будинку, визначення небезпечних факторів і обґрунтування способів і засобів пожежопопередження і захисту.

Одне з умов забезпечення пожежобезпеки - ліквідація можливих джерел запалення.

У приміщенні з електроприборами джерелами запалення можуть бути:

– несправне електроустаткування, несправності в електропроводці, електричних розетках і вимикачах. Для виключення виникнення пожежі з цих причин необхідно вчасно виявляти й усувати несправності, проводити плановий огляд і вчасно усувати всі несправності;

– несправні електроприлади. Необхідні міри для виключення пожежі містять у собі своєчасний ремонт електроприладів, якісне виправлення поломок, не використання несправних електроприладів;

– обігрівання приміщення електронагрівальними приладами з відкритими нагрівальними елементами. Відкриті нагрівальні поверхні можуть спричинити пожежу, тому що в приміщенні знаходяться паперові документи і довідкова література у виді книг, посібників, а папір – легкозаймистий предмет. З метою профілактики пожежі пропоную не використовувати відкриті обігрівальні прилади в приміщенні лабораторії;

– коротке замикання в електропроводці. З метою зменшення імовірності виникнення пожежі внаслідок короткого замикання необхідно, щоб електропроводка була схованою.

– влучення в будинок блискавки. У літній період під час грози можливе влучення блискавки внаслідок чого можливий пожежа. Щоб уникнути цього я рекомендую установити на даху будинку блискавковідвід;

– недотримання мір пожежної безпеки і паління в приміщенні також може спричинити пожежу. Для усунення загоряння в результаті паління в приміщенні

лабораторії пропоную категорично заборонити паління, а дозволити тільки в строго відведеному для цього місці.

З метою запобігання пожежі пропонується проводити з інженерами, що працюють у приміщенні, протипожежний інструктаж, на якому ознайомити працівників із правилами протипожежної безпеки, а також навчити використанню первинних засобів пожежогасіння.

У кімнаті знаходиться вогнегасник марки ОУ-2 (вуглекислотний), а згідно ППБУ-95[9] на кожні 20м^2 повинно бути 2 вогнегасники. Для приміщення таких розмірів необхідно ще один вогнегасник.

В приміщенні встановлена пожежна сигналізація Алай-П-8-1. Оповісники СПД-3-10Б-2, БРИЗ-3 реагуючі на задимленість та температуру відповідно. Сигналізація приміщення ввімкнена до централізованого інформаційного пункту, від якого в випадку появи пожежі сигнал передається в пожежну частину.

Параметри при евакуації розглянуті в таблиці 5.6.

Параметр	Норма	Фактичні
1	2	3
Висота дверного отвору	не менш 2,0 м	2,0 м
Ширина дверного отвору	не менш 0,8 м	1 м
Ширина проходу при евакуації	не менш 1 м	1,1 м

Таблиця 6.6.1 – Основні параметри виходів при евакуації

Двері відкриваються назовні.

Отже, загалом майже всі показники з безпеки праці на робочому місці задовольняють встановленим нормам.

Додаткових заходів щодо забезпечення електробезпеки кабінету не потрібно. Всі працівники приміщення мають бути ознайомлені з правилами техніки безпеки.

Освітленість не задовольняє зазначеним нормам. Потрібно прийняти необхідних заходів для вирішення цієї проблеми.

Для виправлення проблем протипожежної ситуації рекомендується додатково розмістити в приміщенні ще один вогнегасник.

Згідно з ДСН 3.3.6.037-99[7] показники шуму є допустимі і задовольняють встановленим нормам.

7. ВИСНОВКИ

Таким чином, інформаційне перевантаження остаточно стало основною проблемою Веб 2.0. Соціальні мережі не тільки надали можливість користувачам ділитися своїми думками, але й стали фактично, новою платформою для розробки алгоритмів інтелектуального аналізу даних.

В даній роботі розглянуто, як соціальні мережі можуть слугувати додатковими джерелами даних для підвищення структурованості даних і покращення точності їх аналізу.

Зокрема, розглянуто задачу – передбачення рейтингу в процесі рекомендації. Для цієї задачі було запропоновано покращення до існуючих підходів, враховуючи дані, що були отримані з соціальних мереж.

Розглянуто можливість використання рівня довіри між користувачами, категорізацію друзів, та використання інформації з профілю користувача. Це дозволило підвищити ефективність надання рекомендацій (зменшити RMSE) при побудові рекомендаційної системи.

Цей підхід також дозволив частково вирішити проблему «холодного старту» - додаткові дані, які збираються системою про користувача з його профілю в соціальних мережах дозволяють зробити припущення, що йому буде цікаво, а не рекомендувати речі навмання, поки не назбирається достатньо даних.

8. СПИСОК ЛІТЕРАТУРИ

1. NetflixPrize, http://www.netflixprize.com/
2. G. Adomavicius and A. Tuzhilin. “Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions”, IEEE Transactions on Knowledge and Data Engineering
3. ДСанПіН 3.3.2-007-98 Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ЕОМ – Режим доступу: http://document.ua/derzhavni-sanitarni-pravila-i-normi-roboti-z-vizualnimi-disp-nor4881.html
4. ДСН 3.3.6-042-99 Санітарні норми мікроклімату виробничих приміщень – Режим доступу: http://www.document.ua/sanitarni-normi-mikroklimatu-virobnichih-primishen-nor4880.html
5. ДСН 3.3.6.037-99 Санітарні норми виробничого шуму, ультразвуку та інфразвуку – Режим доступу: http://document.ua/sanitarni-normi-virobnichogo-shumu-ultrazvuku-ta-infrazvuku-nor4878.html
6. ДБН В.1.1.– 7– 2002 ПОЖЕЖНА БЕЗПЕКА ОБ’ЄКТІВ БУДІВНИЦТВА – Режим доступу: http://www.poliplast.ua/doc/dbn_v.1.1-7-2002..pdf
7. ППБУ-95 ППБУ-95. Правила пожежної безпеки України.
8. F. Ricci, L. Rokach, B. Shapira and P. B. (Eds.) Kantor “Recommender Systems Handbook”, 1st Edition., 2011, XXX, 842 p. 20 illus.
9. Y. Koren, R. Bell and C. Volinsky. “Matrix Factorization Techniques for Recommendation Systems”, IEEE computer, 2009.
10. J. Bennet and S. Lanning. “The Netflix Prize”, KDD Cup and Workshop, 2007. www.netflixprize.com.
11. Y. Koren. Collaborative filtering with temporal dynamics. In Proc. of KDD ’09, pages 447-456, Paris, France, 2009.
12. G.-R. Xue, C. Lin, Q. Yang, W. Xi, H.-J. Zeng, Y. Yu,and Z. Chen. Scalable collaborative filtering using cluster-based smoothing. In Proc. of SIGIR ’05, pages 114-121, Salvador, Brazil, 2005.
13. W. Deming and F. Stephan. On least square adjustment of sampled frequency

tables when the expected marginal totals are known, *Ann. Math. Statist.*, **6**, pp. 427-44, 1940.

14. V. Sindhwani, S. S. Bucak, J. Hu and A. Mojsilovic. One-Class Matrix Completion with Low-Density Factorizations. In IEEE ICDM 2010.
15. A. Paterek. Improving regularized singular value decomposition for collaborative filtering. In *KDDCup*, 2007.
16. S. Deerwester, S. Dumais, G. Furnas, R. Harshman, T. Landauer, K. Lochbaum, L. Streeter, et al. Latent semantic analysis / indexing. homepage: http://lsa.colorado.edu/.
17. YouTube, available at http://www.youtube.com.
18. Epinions, available at http://www.epinions.com/
19. Twitter, available at http://twitter.com
20. Facebook, http://www.facebook.com
21. Google Plus, http://plus.google.com/
22. Flixster, http://www.flixster.com
23. IMDB, http://www.imdb.com
24. Last.fm, http://www.last.fm
25. Sina corporation Q3 report 2012, http://tech.sina.com.cn/i/2012-1116/05307803196.shtml
26. Sina vote, available at http://vote.weibo.com/.
27. GroupLens. “MovieLens Data Sets”, available at http://www.grouplens.org/node/73#attachments, 2010.
28. S. Funk. Netflix update: Try this at home, 2006. http://sifter.org/simon/journal/20061211.html
29. J. A. Golbeck. Computing and applying trust in web-based social networks, Doctoral Dissertation, 2005.
30. C.-N. Ziegler and G. Lausen. “Analyzing Correlation Between Trust and User Similarity In Online Communities”, Proceedings of Second International Conference on Trust Management, pp. 251-265, 2004.

31. F. Walter, S. Battiston and F. Schweitzer. “A model of a trust-based recommendation system on a social network”, Autonomous Agents and Multi-Agent Systems, vol. 16, no. 1, pp. 1573-7454, Oct. 2007.

32. P. Bedi, H. Kaur, and S. Marwaha. Trust based recommender system for semantic web. In Proc. of IJCAI ’07, pages 2677-2682, 2007.

33. J. S. Breese and D. Heckerman and C. Kadie, ”Empirical Analysis of Predictive Algorithms for Collaborative Filtering”, Technical Report, Morgan Kaufmann, pp. 43-52, 1998.

34. Q. Yuan, S. Zhao, L. Chen, Y. Liu , S. Ding, X. Zhang and W. Zheng. Augmenting collaborative recommender by fusing explicit social relationships. Recommender Systems & the Social Web workshop, RecSys, 2009.

35. Toby Segaran. Programming Collective Intelligence. O'REILLY, 2007

36. Кислий Р.В. Моделювання профілю користувача для уникнення проблеми холодного старту в рекомендаційних системах / Кислий Р. В. // Системний аналіз та інформаційні технології: материалы 17-й Міжнародної науково-технічної конференції SAIT 2015, Київ, 22-25 червня 2015 р. – 2015. – с.244

37. Кислий Р. В. Колаборативна фільтрація в рекомендаційних системах на основі даних з сщціальних мереж / Кислий Р. В. // Матеріали III-ої міжнародної науково-практичної конференції Обчислювальний інтелект (результати, проблеми, перспективи), Київ-Черкаси, 12-15 травня 2015р. - 2015. - с.208

38. Kaggle, http://www.kaggle.com

39. M. Jamali and M. Ester. “A Matrix Factorization Technique with Trust Propagation for Recommendation in Social Networks” Proceedings of the 2010 ACM conference on Recommender systems(RecSys), 2010. http://dl.acm.org/citation.cfm?id=1864736

40. H. Ma, H. Yang, M. R. Lyu, and I. King. Sorec: Social recommendation using probabilistic matrix factorization. In International Conference on Information and Knowledge Management (CIKM), 2008. http://dl.acm.org/citation.cfm?id=1458205

41. H. Ma, I. King, and M. R. Lyu. Learning to recommend with social trust ensemble. In ACM conference on Research and development in information retrieval (SIGIR), 2009. http://dl.acm.org/citation.cfm?id=1571978

42. H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King. Recommender Systems with Social Regularization. In ACM International Conference on Web Search and Data Mining (WSDM), 2011. http://research.microsoft.com/en-

us/um/people/denzho/papers/rsr.pdf

43. L. Yu, R. Pan, and Z Li. Adaptive social similarities for recommender systems. In RecSys ’11 Proceedings of the fifth ACM conference on Recommender systems , 2011 http://dl.acm.org/citation.cfm?id=2043978