

В. В. Міхав, Є. В. Мелешко

Центральноукраїнський національний технічний університет, Кропивницький, Україна

МЕТОД РОБОТИ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ У КОМП'ЮТЕРНІЙ МЕРЕЖІ ТИПУ PEER TO PEER

Анотація. Стаття присвячена розробці методу роботи рекомендаційних систем в однорангових децентралізованих комп'ютерних мережах. Існує велика кількість методів роботи рекомендаційних систем для веб-сайтів та застосунків, призначених для централізованих комп'ютерних мереж. В той же час дослідження децентралізованих однорангових мереж показало, що питання створення рекомендаційних систем для такої архітектури практично не розглядалося. Проведено дослідження методів роботи однорангових децентралізованих структурованих комп'ютерних мереж, наведено узагальнені принципи їх роботи. Запропоновано метод роботи рекомендаційної системи на основі колаборативної фільтрації для однорангової децентралізованої структурованої комп'ютерної мережі з адресацією на основі розподілених хеш-таблиць. Для представлення даних рекомендаційної системи було використано дві структури даних: розгорнуті зв'язні списки та хеш-таблиці з відкритою адресацією. Використовувалося два типи розгорнутих списків: асоційовані з користувачами – містили списки вподобаних ними об'єктів та асоційовані з об'єктами – містили списки користувачів, які вподобали відповідні об'єкти. Кожен комп'ютер децентралізованої комп'ютерної мережі містив список вподобань асоційований зі своїм користувачем та списки вподобань асоційовані з об'єктами, які на ньому розташовані. Доступ до відсутньої на поточному комп'ютері інформації здійснювався за допомогою запитів до інших комп'ютерів за відповідними ідентифікаторами, організованими розподіленою хеш-таблицею однорангової децентралізованої мережі. Проведено експериментальне дослідження запропонованого методу роботи рекомендаційної системи. Воно показало, що розроблений метод має досить високі показники точності (Precision) та низьку середньоквадратичну помилку. Недоліком розробленого методу є низькі показники повноти (Recall), що виникає через те, що метод часто робить помилкові негативні прогнози. Це жодним чином не впливає на точність рекомендацій, тому що об'єкти з негативним прогнозом вподобань не потрапляють до списку рекомендацій. Але це погіршує наповненість списків, адже багато елементів, які можна було додати до них, відсіюються.

Ключові слова: рекомендаційні системи, колаборативна фільтрація, комп'ютерні мережі, однорангові мережі, децентралізовані мережі, розподілені хеш-таблиці.

Вступ

Рекомендаційні системи стали звичним та важливим елементом сучасних контент-орієнтованих веб-сайтів та соціальних мереж [1, 2].

Існує велика кількість методів роботи рекомендаційних систем для веб-сайтів та застосунків, призначених для централізованих комп'ютерних мереж. В той же час дослідження децентралізованих однорангових мереж показало, що питання створення рекомендаційних систем для такої архітектури практично майже не розглядалося [3-11].

Хоча в peer to peer мережах дуже гостро стоїть питання інформаційного пошуку, а рекомендаційні системи з різними алгоритмами фільтрації даних якраз могли б значно полегшити користувачам пошук корисного для них контенту та збільшили б інтерес до однорангових мереж для розподіленого зберігання контенту.

В деяких видах peer to peer мереж класичні методи роботи рекомендаційних систем не зможуть працювати без певної адаптації та змін внаслідок особливостей архітектури цих мереж.

Методи організації P2P мереж можна розділити на наступні три види [3-13]:

– Централізовані однорангові мережі, наприклад, BitTorrent [12].

– Неструктуровані децентралізовані однорангові мережі, цей тип мереж використовує алгоритм пошуку Flooding, типовий представник – Gnutella [13].

– Структуровані децентралізовані однорангові мережі, найчастіше будуються на основі розподілених хеш-таблиць, наприклад, на основі алгоритму Kademia [10] або Chord [11].

Для централізованих однорангових мереж можливе застосування стандартних методів побудови рекомендаційних систем, оскільки у таких мережах є сервер, що зберігає посилання на файли та статистику. Складнощі виникають при намаганні створити рекомендаційні системи для децентралізованих однорангових мереж, тому що треба враховувати їх архітектуру і адаптувати відомі методи та підходи до неї. Зокрема, в децентралізованих мережах уся інформація розподілена по усім комп'ютерам мережі (це і файли, і таблиці маршрутизації, і статистика тощо), тому виникає проблема збору інформації для рекомендаційної системи та розгляд можливості зібрати не всю інформацію, а мінімально потрібну, щоб не опитувати всі комп'ютери у мережі.

У цій роботі було запропоновано метод роботи рекомендаційної системи для однорангової децентралізованої структурованої комп'ютерної мережі, побудованої з застосуванням розподілених хеш-таблиць. Було вирішено розглядати саме структуровані децентралізовані мережі, так як на відміну від неструктурованих вони гарантують знаходження файлу, наявного у даний момент часу у мережі. А неструктуровані децентралізовані мережі знаходять наявний файл тільки з деякою ймовірністю, тому менше підходять для якісного розподіленого зберігання та обміну файлами.

1. Дослідження загальних принципів роботи однорангових децентралізованих структурованих комп'ютерних мереж

Розглянемо принципи роботи однорангових децентралізованих структурованих комп'ютерних мереж. Найчастіше для індексації та маршрутизації вони використовують розподілені хеш-таблиці (Distributed hash table, DHT). Для простоти будемо називати далі такі комп'ютерні мережі – P2P DHT мережі.

Узагальнені принципи роботи P2P DHT мереж наступні [3-5, 10, 11]:

1. *Ідентифікатори комп'ютерів.* Комп'ютери мають ідентифікатори $h \in H$, що змінюються у діапазоні $[0, 2^m-1]$. Вони визначається деякою хеш-функцією або залежить від черговості підключення комп'ютерів до мережі. Наприклад, 1-й комп'ютер у мережі отримує ідентифікатор $h_0=0$. i -тий комп'ютер одержує ідентифікатор $h_i=i$ або один з вільних ідентифікаторів, що звільнилися внаслідок виходу з мережі деяких комп'ютерів, що раніше були приєднані. Оскільки в децентралізованих мережах комп'ютери можуть приєднуватися та від'єднуватися від мережі, тобто, учасники постійно змінюються, задача виявлення уже невикористовуваних індексів і надання їх новим комп'ютерам є важливою та по-різному вирішується у різних алгоритмах. Як правило, не тільки при приєднанні нового комп'ютера до мережі, а й з деякою періодичністю відбувається перевірка доступності вузлів та оновлення таблиць маршрутизації.

2. *Ідентифікатори файлів.* Файли також мають

ідентифікатори $h \in H$, що змінюються у такому ж діапазоні як i у комп'ютерів $[0, 2^m-1]$. Вони визначаються деякою хеш-функцією. Використовується так зване узгоджене хешування, що на відміну від лінійного передбачає зміну в середньому тільки K/n ключів, де K – число ключів, а n – число слотів. Популярним є застосування алгоритму хешування SHA-1 в P2P DHT мережах, в такому разі $m=160$, що дозволяє створити досить велику множину ідентифікаторів.

3. *Зберігання файлів.* Файли чи їх частини (при розподіленому зберіганні) або шляхи до них (при розподіленому індексуванні) зберігаються на комп'ютерах таким чином, щоб ідентифікатор комп'ютера та ідентифікатор файлу (його частини) співпадали. Або, при відсутності у мережі комп'ютера з потрібним ідентифікатором, вони повинні бути максимально близькі з наявних варіантів за деякою метрикою відстані. Надалі для простоти будемо розглядати розподілене зберігання файлів. Для забезпечення надійності зберігання інформації кожен файл дублюють на декілька комп'ютерів, для цього обирається q наявних у мережі комп'ютерів, ідентифікатори яких найближчі за обраною метрикою відстані до ідентифікатора файлу. При зміні складу учасників мережі, за необхідності, відбувається перерозподіл частини файлів між комп'ютерами.

4. *Таблиця маршрутизації.* Зберігання таблиці маршрутизації також є розподіленим. Кожен комп'ютер містить інформацію потрібну для доступу (мережеві адреси) найближчих N сусідів $X = \{x_{i-n}, \dots, x_{i+n}\}$ рис. 1.

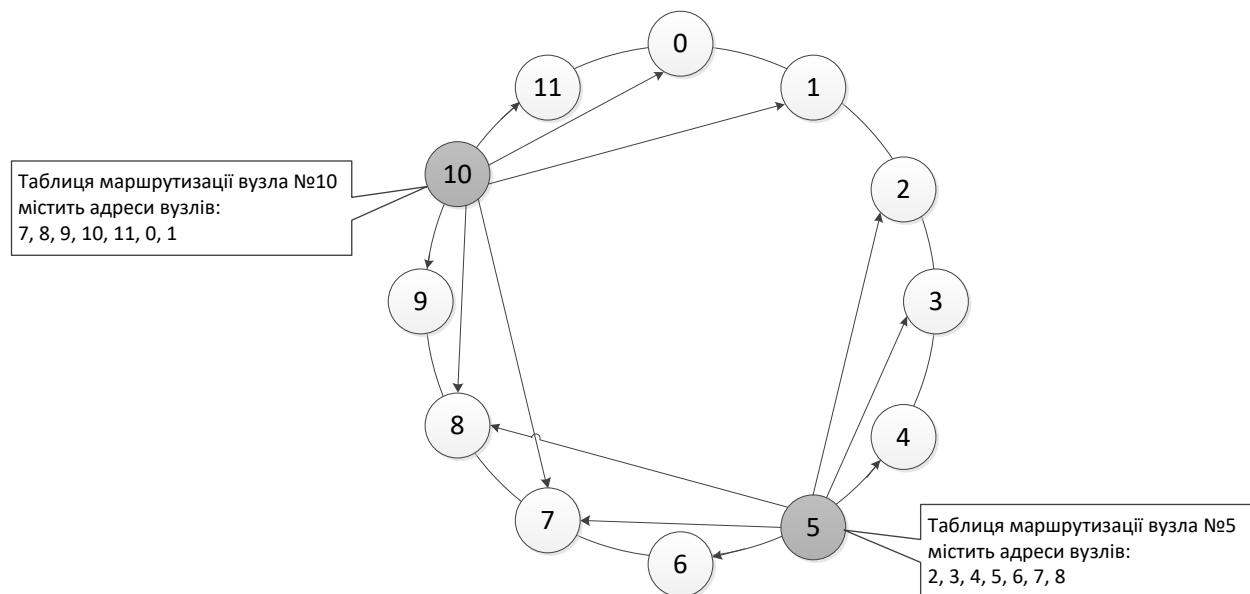


Рис. 1. Схематичне зображення принципу наповнення таблиць маршрутизації в однорангових децентралізованих структурованих комп'ютерних мережах: таблиці маршрутизації показані для комп'ютерів №5 та №10 при зберіганні 6 найближчих сусідів у мережі з 12 вузлів за принципами алгоритму Chord

Визначення множини X , а також розмір N цієї множини, у кожному конкретному методі здійснюється по-різному.

Наприклад, в алгоритмі Chord [11] таблиця маршрутизації комп'ютера містить $2r+1$ записи, тобто

інформацію про сам комп'ютер, а також про r комп'ютерів, що мають найближчі менші номери серед наявних у мережі комп'ютерів, та про r комп'ютерів, що мають найближчі більші номери серед наявних комп'ютерів.

Комп'ютери об'єднуються в кільце, тобто, останній комп'ютер посилається на перший. Що викликає можливі проблеми з неправильним замиканням кільця, виникненням декількох кілець та розривом кільця при помилках в оновленні таблиць маршрутизації. В алгоритмі Kademia [10] таблиця маршрутизації зберігається у вигляді так званих K-bucket-ів. У кожному K-bucket-ті вузла зберігається інформація про K вузлів мережі, чия відстань до нього знаходиться в межах інтервалу $[2^i, 2^{(i+1)})$, у якості метрики відстані використовується операція XOR. Якщо, наприклад, для створення ідентифікаторів застосовується хеш-функція SHA-1, то кількість K-bucket-ів на кожному вузлі буде 160. K – загально-системне число, наприклад 20. Кожен K-bucket – це список, що містить не більше K -записів; тобто для мережі з $K=20$ кожен вузол матиме списки, що містять до 20 вузлів для певної відстані від себе. На практиці виходить, що кожен вузол зберігає інформацію про вузли, з якими будь-коли взаємодіяв, протягом певного часу (наприклад, час життя запиту 24 години). Таким чином розмір таблиці маршрутизації змінюється динамічно, а деякі K-bucket-ти будуть порожніми в деякий момент часу.

5. *Пошук файлів.* Файли у мережі шукаються за їх хеш-значеннями. Комп'ютер, який здійснює пошук файлу, вибирає в своїй таблиці маршрутизації j комп'ютерів, ідентифікатори яких найближчі за обраною метрикою відстані до ідентифікатора шуканого файлу. Кожен комп'ютер з множини вибраних $S = \{s_0, s_1, \dots, s_j\}$ перевіряє чи міститься безпосередньо на ньому шуканий файл, і якщо ні, то пересилає пошукове повідомлення уже j найпідходящим комп'ютерам зі своєї таблиці маршрутизації, і так поки файл не буде знайдений. Чим більше значення s , тим менша ймовірність відмови та вищий рівень інформаційної безпеки, але й вище навантаження на мережу. Можна було б пересилати запит тільки на один комп'ютер з найближчим до шуканого ідентифікатором, але може виявитися, що він на даний час відсутній у мережі. Або ж цим комп'ютером володіє зловмисник, що перенаправить запит не на пошук та завантаження потрібного файлу, а на завантаження вірусу, в той же час при направленні запиту на декілька комп'ютерів можна буде обрати ту відповідь, яка прийде від більшості, що значно зменшить ймовірність інформаційної атаки.

6. *Основні запити до мережі.* Методи роботи P2P DHT мереж повинні містити також наступні функції:

- Перевірка наявності сусідніх вузлів у мережі для оновлення таблиць маршрутизації;
- Пошук вузлів за ідентифікатором;
- Пошук файлів за ідентифікатором.

Зокрема, в протоколі алгоритму Kademia для їх реалізації наявні наступні 4 типи запитів:

- 1) PING – необхідний для перевірки існування конкретного вузла у мережі;
- 2) STORE дає змогу розмістити інформацію на заданому вузлі;
- 3) FIND_VALUE – дозволяє знайти значення за ключем;

4) FIND_NODE – використовується для пошуку найближчих K вузлів до заданого ідентифікатора (схожий на FIND_VALUE, тільки ніколи не повертає значення, завжди вузли).

2. Запропонований метод роботи рекомендаційної системи у однорангових децентралізованих структурованих мережах

З врахуванням особливостей архітектури P2P DHT мереж було запропоновано метод роботи рекомендаційних систем для них.

Для реалізації рекомендаційної системи у P2P DHT мережі найкраще підійдуть підходи засновані на колаборативній фільтрації, адже для неї достатньо знати тільки реакцію користувача на контент, що може бути представлена оцінками, лайками або самим фактом завантажень файлів.

Вміст файлів, їх назви та інформацію про користувачів знати не обов'язково.

Таким чином можна працювати лише з ідентифікаторами файлів, ідентифікаторами користувачів та статистикою завантаження файлів, це все можна отримати при роботі з децентралізованою структурованою одноранговою комп'ютерною мережею.

Для представлення даних рекомендаційної системи у пам'яті було використано дві структури даних: розгорнуті зв'язні списки та хеш-таблиці із відкритою адресацією.

Розгорнуті зв'язні списки були обрані після проведення експерименту з використанням програмної імітаційної моделі, де тестувалися різні методи зберігання даних рекомендаційної системи [14]. Хеш-таблиці обрані для проміжних обчислень.

Розгорнутий зв'язний список – це зв'язний список, який складається із блоків з наперед визначеною кількістю елементів.

Така структура дає можливість поєднати ефективно розширення зв'язних списків та переваги послідовного доступу до елементів блоку. Було використано цю структуру даних для збереження інформації про вподобання користувачів.

Використовувалося два типи списків:

- 1) асоційовані з користувачем – містять список об'єктів, які він вподобав;
- 2) асоційовані з об'єктом – містять список користувачів, які вподобали цей об'єкт.

Таким чином, використовуються дві копії інформації про вподобання, що збільшує витрати пам'яті, але дає можливість уникати перегляду усіх вподобань для генерації рекомендації та дає можливість врахувати архітектуру P2P DHT мереж. Зокрема, кожен комп'ютер може містити список вподобань асоційований зі своїм користувачем та списки вподобань асоційовані з об'єктами, які на ньому розташовані. Ця інформація доступна для збору кожному комп'ютеру, для цього треба тільки зберігати статистику дій відповідного користувача та статистику завантажень наявних файлів.

В такому разі під вподобанням буде розумітися факт завантаження файлу. Якщо впровадити у P2P DHT мережу можливість оцінювання файлів, схема буде трохи складнішою.

Доступ до потрібної інформації, яка не розміщена на комп'ютері користувача, якому треба сформулювати рекомендації, може бути отримана запитами до інших комп'ютерів за ідентифікаторами, організованими розподіленою хеш-таблицею P2P DHT мережі.

Також було використано хеш-таблиці з відкритою адресацією для збереження проміжних значень, необхідних для генерації рекомендацій, в тому числі для підрахунку кількості входжень елементів. Для цього було реалізовано операцію інкременту, яка працює наступним чином:

1. Здійснюється пошук елемента за ключем.
2. Якщо елемент не існує, створюється новий елемент зі значенням 1.
3. Якщо елемент існує і має значення -1, то його значення не змінюється, інакше – збільшується на 1.

Такий підхід дає можливість заборонити інкрементувати певні елементи.

Запропонований метод роботи рекомендаційної системи складається з наступних етапів:

Етап 1. Створити хеш-таблицю з ідентифікаторами користувачів H_u та хеш-таблицю з ідентифікаторами об'єктів H_f . Одержати доступ до розгорнутих списків, асоційованих з користувачами L_u та асоційованих з об'єктами L_f , що містять історію дій (завантаження/вподобання).

Етап 2. Переглянути список вподобань L_{u0} користувача U_0 , для якого треба створити рекомендації. Для кожного вподобаного користувачем U_0 об'єкту створити у хеш-таблиці об'єктів H_f елемент зі значенням -1.

Етап 3. Для кожного користувача, який вподобав один з об'єктів, що в H_f мають значення -1, інкрементувати відповідний елемент у хеш-таблиці H_u .

Етап 4. Для кожного користувача U_i з хеш-таблиці користувачів H_u , елемент якого не містить -1, переглянути відповідний список вподобань L_{ui} та інкрементувати елементи у хеш-таблиці об'єктів H_f , якщо деякий користувач вподобав відповідний об'єкт.

Етап 5. Створити масив M_{rec} для збереження списку рекомендацій. Для кожного елемента з хеш-таблиці об'єктів H_f , значення якого не -1, здійснити спробу вставити цей елемент до масиву M_{rec} у порядку спадання значень. Обрати $TopN$ рекомендацій з масиву M_{rec} для показу користувачу U_0 .

На початку роботи запропонований метод позначає цільового користувача U_0 (якому треба сформулювати рекомендації) як недоступного для участі у проміжних обчисленнях, шляхом занесення у хеш-таблицю користувачів елемента, ключ якого – це ідентифікатор цього користувача, а значення дорівнює -1.

Після цього виконується обхід списку вподобань цільового користувача. Кожен вподобаний об'єкт заноситься до хеш-таблиці об'єктів зі значенням -1, щоб уникнути повторної рекомендації уже вподобаних об'єктів.

Якщо система допускає повторні рекомендації, то цей етап може бути змінений.

Далі для кожного вподобаного об'єкту вибирається список користувачів, які його вподобали.

Для кожного користувача із цього списку

виконується інкрементація відповідного значення у хеш-таблиці користувачів. Це значення буде використано для визначення ступеня подібності користувачів. Після завершення обходу хеш-таблиця користувачів фільтрується, щоб виключити користувачів, вподобання яких не будуть використовуватися при формуванні рекомендації.

Під час наступного етапу виконується обхід усіх елементів хеш-таблиці користувачів. Для кожного з цих користувачів вибирається відповідний список вподобань.

Для кожного елемента з цього списку виконується інкрементація відповідного значення у хеш-таблиці об'єктів. На основі цього значення визначається, чи потрапить об'єкт до рекомендації.

На останньому етапі з хеш-таблиці об'єктів відбираються елементи із найбільшими значеннями та передаються як рекомендації для цільового користувача.

3. Результати експериментів для визначення точності запропонованого методу роботи рекомендаційної системи

Було проведено експериментальне дослідження запропонованого методу роботи рекомендаційної системи в P2P DHT мережі. Результати проведених експериментів наведені у табл. 1.

Для проведення експериментів використовувалася програмна імітаційна модель P2P DHT мережі та дані з датасету MovieLens [15], що переводилися до потрібного формату, зокрема, оцінки для простоти були перетворені на лайки в одній серії експериментів та лайки і дизлайки в іншій серії експериментів.

Були проведені наступні серії експериментів, що відрізнялися способом оцінювання об'єктів у програмній моделі та обробкою результатів:

– факт вподобання користувачем файлу визначався наявністю лайку (або ж інша інтерпретація – фактом завантаження файлу) – експерименти №1-3.

– відношення користувача до файлу визначалося лайками та дизлайками – експерименти №4-6.

– відношення користувача до файлу визначалося лайками та дизлайками, кожен з них множився на коефіцієнт $k=2$ перед запуском розрахунків для формування рекомендацій – експерименти №7-9.

– відношення користувача до файлу визначалося лайками та дизлайками, кожен з них множився на коефіцієнт $k=5$ перед запуском розрахунків для формування рекомендацій – експерименти №10-12.

В межах кожної серії експериментів використовувалися різні розміри списків рекомендацій, а саме, 128, 512 та 1024 об'єкти у списку.

Для оцінки якості роботи рекомендаційної системи використовувалися наступні метрики [16]: Precision (точність), Recall (повнота) та RMSE (середньоквадратична помилка):

$$\text{Precision} = \frac{tp}{tp + fp}, \quad (1)$$

$$\text{Recall (True Positive Rate)} = \frac{tp}{tp + fn}, \quad (2)$$

де tp – результати, в яких позитивний прогноз виявився вірним; fp – результати, в яких позитивний прогноз виявився помилковим; fn – результати, в яких негативний прогноз виявився помилковим.

Таблиця 1 – Результати експериментів для перевірки якості роботи запропонованого методу

№ експ.	Спосіб оцінювання контенту у системі	Кількість об'єктів у списку рекомендацій	Precision (Точність)	Recall (Повнота)	RMSE
1	Тільки лайки (або факт завантаження)	128	0,7605	0,2033	0,5258
2	Тільки лайки (або факт завантаження)	512	0,7160	0,4443	0,4471
3	Тільки лайки (або факт завантаження)	1024	0,6917	0,6074	0,4052
Середнє значення			0,7227	0,4183	0,4593
4	Лайки та дизлайки	128	0,7786	0,2082	0,5199
5	Лайки та дизлайки	512	0,7530	0,4346	0,4324
6	Лайки та дизлайки	1024	0,7382	0,5825	0,3812
Середнє значення			0,7566	0,4084	0,4445
7	Лайки та дизлайки, помножені на $k=2$	128	0,7936	0,1969	0,5219
8	Лайки та дизлайки, помножені на $k=2$	512	0,7820	0,3880	0,4399
9	Лайки та дизлайки, помножені на $k=2$	1024	0,7765	0,4983	0,3941
Середнє значення			0,7840	0,3610	0,4519
10	Лайки та дизлайки, помножені на $k=5$	128	0,8338	0,1507	0,5372
11	Лайки та дизлайки, помножені на $k=5$	512	0,8391	0,2293	0,4976
12	Лайки та дизлайки, помножені на $k=5$	1024	0,8401	0,2529	0,4858
Середнє значення			0,8376	0,2109	0,5068

Також для оцінки якості роботи рекомендаційної системи використовувалася метрика [16] RMSE (середньоквадратична помилка):

$$RMSE = \sqrt{\frac{1}{|\tau|} \sum_{(u,i) \in \tau} (\hat{r}_{ui} - r_{ui})^2}, \quad (3)$$

де \hat{r}_{ui} – прогнозовані рейтинги (вподобання) для тестового набору даних τ пар користувач-об'єкт (u, i) ; r_{ui} – справжні рейтинги.

Усі дані з датасету були поділені на робочі та тестові. На основі робочих даних здійснювалися прогнози вподобань користувачів та формувалися списки рекомендацій. На основі тестових даних перевірялася точність рекомендацій за формулами (1)-(3).

Як показали результати експериментів, наведені у табл. 1, точність (Precision) запропонованого методу виявилася досить високою і варіювалася у діапазоні $0,6917 \div 0,8401$ в залежності від обраних параметрів системи. Найкращі результати були отримані, коли була можливість використовувати і лайки, і дизлайки користувачів, а їх значення були помножені на коефіцієнт $k=5$.

Повнота (Recall) розробленого методу, на жаль, виявилася низькою через те, що метод часто робить помилкові негативні прогнози. Це жодним чином не впливає на точність рекомендацій, тому що об'єкти з негативним прогнозом вподобань не потрапляють до списку рекомендацій. Але це погіршує наповненість списків, адже багато елементів, які можна було додати до списків, відсіюються.

Середньоквадратична помилка досить низька, її значення варіювалося у діапазоні $0,3812 \div 0,5372$ і дещо зменшувалася при збільшенні довжини списку рекомендацій.

Висновки

Було запропоновано метод роботи рекомендаційної системи на основі колаборативної фільтрації для однорангової децентралізованої структурованої комп'ютерної мережі з адресацією на основі розподілених хеш-таблиць.

Також було проведено експериментальне дослідження запропонованого методу. Воно показало, що розроблений метод має досить високі показники точності (Precision) від $0,6917$ до $0,8401$ в залежності від

обраних параметрів системи та низьку середньоквадратичну помилку від 0,3812 до 0,5372.

Найкращі результати отримані, коли була можливість використовувати і лайки, і дизлайки користувачів, а їх значення множилися на коефіцієнт $k=5$.

Недоліком розробленого методу є низькі показники повноти (Recall), що виникає через те, що метод

часто робить помилкові негативні прогнози. Це не впливає на точність рекомендацій, але погіршує наповненість списків рекомендацій, адже багато елементів, які можна було додати до них, відсіюються.

Повнотою рекомендацій було пожертвовано для адаптації колаборативної фільтрації до архітектури P2P мереж.

СПИСОК ЛІТЕРАТУРИ

1. "Recommender Systems Handbook" (2010) Editors F. Ricci, L. Rokach, B. Shapira, P. B. Kantor, New York, NY, USA: Springer-Verlag New York, Inc., 842 p.
2. Jones M. (2013) "Recommender systems, Part 1. Introduction to approaches and algorithms. Learn about the concepts that underlie web recommendation engines", URL: https://www.ibm.com/developerworks/opensource/library/os-recommender1/index.html?s_tact=105agx99&s_cmp=cp
3. Riposo Ju. Diffusion on the Peer-to-Peer Network // LAP LAMBERT Academic Publishing. – 2022. – 100 p.
4. Koo S.G.M. Multimedia Content Distribution Using Peer-to-Peer Overlay Networks: The Design and Analysis of the Next Generation Peer-to-Peer Networks // VDM Verlag Dr. Müller. – 2008. – 88 p.
5. Milojicic D.S., Kalogeraki V., Lukose R., Nagaraja K., Pruney J., Richard B., Rollins S., Xu Z. Peer-to-peer computing // Technical Report HPL-2002-57, HP Labs. – 2002. – 51 p. – URL: <https://www.cs.kau.se/cs/education/courses/dvad02/p2/seminar4/Papers/HPL-2002-57R1.pdf>
6. Zeinalipour-Yazti D., Kalogeraki V., Gunopulos D. Information retrieval techniques for peer-to-peer networks // Computing in Science & Engineering, Vol. 6, No. 4, pp. 20-26. – 2004. – DOI: 10.1109/MCSE.2004.12
7. Lua E.K., Crowcroft J., Pias M., Sharma R., Lim S. A Survey and Comparison of Peer-to-Peer Overlay Network Schemes // IEEE Communications survey and tutorial. – 2004. – URL: <https://snap.stanford.edu/class/cs224w-readings/lu04p2p.pdf>
8. Kalogeraki V., Gunopulos D., Zeinalipour-Yazti D. A Local Search Mechanism for Peer-to-Peer Networks // Proc. of CIKM'02, McLean VA, USA, 2002. – URL: <http://alumni.cs.ucr.edu/~csyiazti/downloads/papers/cikm02/cikm02.pdf>
9. Zeinalipour-Yazti D. Information Retrieval in Peer-to-Peer Systems // M.Sc Thesis, Dept. of Computer Science, University of California Riverside. – 2003. – URL: <http://alumni.cs.ucr.edu/~csyiazti/papers/msc/html/>
10. Kademia: A Design Specification – 2010. – URL: <https://xlattice.sourceforge.net/components/protocol/kademia/specs.html>
11. Stoica I., Morris R., Karger D.R., Kaashoek M.F., Balakrishnan H. Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications // ACM SIGCOMM Computer Communication Review, Vol. 31(4). – 2001. DOI: 10.1145/964723.383071
12. The BitTorrent Protocol Specification – 2017. – URL: http://www.bittorrent.org/beps/bep_0003.html
13. Gnutella Protocol Development – 2003. – URL: <https://rfc-gnutella.sourceforge.net>
14. Міхав В.В., Мелешко Є.В., Якименко М.С., Башенко Д.В. Методи зберігання даних рекомендаційної системи на основі зв'язних списків // Системи управління, навігації та зв'язку – Полтава: ПНТУ, 2021. – Т. 4(66). – С. 59-62. – DOI: <https://doi.org/10.26906/SUNZ.2021.4.059>
15. Harper F.M., Konstan J.A. The MovieLens Datasets: History and Context // ACM Transactions on Interactive Intelligent Systems (TiiS). – 2015. – 19 p. – URL: <https://doi.org/10.1145/2827872>
16. Мелешко Є.В. Методи оцінки якості роботи рекомендаційних систем // Системи управління, навігації та зв'язку. – Полтава: ПНТУ, 2018. – Вип. 5 (51). – С. 92-97. – DOI: <https://doi.org/10.26906/SUNZ.2018.5.09>

Received (Надійшла) 11.01.2023

Accepted for publication (Прийнята до друку) 08.03.2023

Method of working of a recommender system in a peer-to-peer computer network

V. Mikhav, Ye. Meleshko

Abstract. The article is devoted to the development of a method of work of recommendation systems in decentralized peer-to-peer computer networks. There are a large number of methods of recommendation systems for websites and applications designed for centralized computer networks. At the same time, the research of decentralized peer-to-peer networks has shown that the issue of creating recommendation systems for such architecture was practically not considered. The research of methods of work of structured decentralized peer-to-peer computer networks has been conducted, and generalized principles of their work are presented. The method of work of a recommendation system based on collaborative filtering for a structured decentralized peer-to-peer computer network with addressing based on distributed hash tables is proposed. To present the data of a recommendation system two data structures were used: the unrolled linked lists and the hash tables with open addressing. Also two types of unrolled linked lists were used: associated with users – contained lists of objects they like and associated with objects – contained lists of users who liked the appropriate objects. Each computer in a decentralized network has a preference list of associated with its user and preference lists of associated with objects located on it. The information missing on the current computer was accessed by requests to other computers using the corresponding IDs organized by a distributed hash table of a peer-to-peer decentralized network. An experimental research of proposed method of work of a recommendation system was conducted. It showed that the developed method has quite high Precision and a low RMSE. The disadvantage of the developed method is the low Recall, that is due to the fact that the method often makes false negative predictions. This does not in any way affect the accuracy of the recommendations, because objects with a negative prediction of preferences do not fall on the lists of recommendations. But this impairs the fullness of recommendations because many elements that could be added to them are rejected.

Keywords: recommendation systems, collaborative filtering, computer networks, peer-to-peer networks, decentralized networks, distributed hash tables.