

Шарабан О.Ю.

студент,

Науковий керівник: Лєсна Н.С.

кандидат технічних наук, професор,

Харківський національний університет радіоелектроніки

РОЗРОБКА ГІБРИДНОГО МЕТОДУ ФОРМУВАННЯ МУЗИЧНИХ РЕКОМЕНДАЦІЙ ЗА ПРИНЦИПОМ ПОДВІЙНОЇ ОРГАНІЗАЦІЇ

У повсякденному житті люди покладаються на рекомендації інших людей, виходячи з розмов із ними, рекомендаційних листів, інформаційних повідомлень із засобів масової інформації, соціальних опитувань, путівників тощо. Рекомендаційні системи сприяють цьому природньому соціальному процесу та допомагають людям серед великої кількості доступних книг, статей, веб-сторінок, фільмів, музики, ресторанів, жартів, продовольчих товарів і тому подібного, знаходити найбільш цікаву і цінну для них інформацію. Серед методів, що використовуються для формування рекомендацій, найкраще себе зарекомендували методи колаборативної фільтрації (КФ) [1] та методи на основі контенту [2].

Методи КФ використовують базу даних вподобань користувачів до певних елементів, щоб передбачити додаткові теми або продукти, які можуть сподобатися новому користувачу. Системи, що реалізують рекомендаційний підхід на основі контенту, аналізують набір документів і/або опис об'єктів, що раніше були оцінені користувачем, і будують модель або профіль інтересів користувача на основі особливостей об'єктів, оцінених тим самим користувачем. Існують також гібридні методи [3], які певним чином комбінують зазначені методи для отримання більш точних результатів.

У даній роботі автор описує та аналізує реалізований ним новий вид гібридного рекомендаційного методу, який, завдяки своїм певним особливостям, призначений для генерації більш точних та якісних рекомендацій.

Розглянемо метод детальніше. Для того, щоб результати роботи методу були найбільш оптимальними, вводиться поняття принципу подвійної організації та розробляється алгоритм, який поєднуватиме у собі принципи паралельної та конвеєрної організації [3] і використовуватиме відомі методи наступним чином:

- фільтрація відбувається в двох незалежних робочих блоках;
- у одному з блоків першим працюватиме метод колаборативної фільтрації, а по закінченню його роботи отримані результати опрацює метод на основі контенту;
- у іншому блоці, навпаки, спочатку відпрацює фільтрація на основі контенту, а потім колаборативна фільтрація;
- після отримання результатів з двох блоків спільний перелік треків буде виведений у якості рекомендацій.

Кожен з окремих блоків підтримує у своїй роботі принцип конвеєрної організації. Тобто вихідні дані, що були отримані першим методом,

використовуються як вхідні дані для другого методу в межах одного блоку. Завдяки цьому підходу проводиться відбір максимально якісного набору рекомендацій.

Оскільки ми маємо справу з кінцевою кількістю рекомендацій, в першу чергу необхідно визначити розмір списку з треками, який буде представлений користувачу. Задамо мінімальну кількість треків, які будуть відображатися як рекомендації, у 30 треків. Кількість композицій, що будуть запропоновані різними методами, коливатиметься з огляду на особливості роботи кожного з методів.

Розглянемо роботу методів генерації рекомендацій, які є першою ланкою в ланцюгу кожного з блоків, тобто тих, що відпрацюють першими незалежно від виду методу. Спочатку буде сформовано вихідний перелік треків, для яких і будуть формуватися рекомендації. Для цього ми об'єднаємо усі треки, що знаходяться в історії користувача, у три категорії: візьмемо треки, які найбільш часто прослуховувалися користувачем, треки, які він прослуховував у найближчий час, та треки зі списку його улюблених треків. Далі кожній з категорій надамо однаковий пріоритет. Доля треків, які послугують основою для пошуку, буде однаковою для трьох названих категорій. Задамо їх сумарну початкову кількість, наприклад, у 30 треків. Таким чином вона становитиме 33.3% або 10 треків для кожної категорії у даному випадку. Якщо кількість треків у якійсь із категорій буде меншою за 10, буде використовуватись та їх кількість, яка є доступною. У разі якщо треки певної категорії взагалі будуть відсутні – категорія не буде враховуватись. Таким чином, кількість треків для категорій буде розраховуватись за формулою 1:

$$n = \frac{P}{N}, \quad (1)$$

де P – кількість треків для відображення, N – кількість категорій, у яких є треки, n – кількість треків для кожної категорії. Якщо жодна з категорій не міститиме треків – за відсутністю сенсу у роботі рекомендаційних методів з порожніми даними у якості рекомендацій виступатимуть популярні у світі треки.

Наступним кроком буде безпосереднє опрацювання сформованих даних одним із зазначених рекомендаційних методів. Метод колабораційної фільтрації, що працюватиме у межах кожного з блоків, шукатиме треки, які є популярними та мають найбільшу кількість прослуховувань у інших користувачів, які слухали треки даного користувача.

Пошук рекомендацій у рамках цього методу відбуватиметься у власній базі даних системи, де знаходиться уся інформація про користувачів. Для першого треку будуть знайдені користувачі, у яких цей трек також входить до одної з категорій. З отриманого списку користувачів буде знайдений трек або кілька треків, які відносяться до названих категорій у найбільшій кількості з цих користувачів. Якщо це кілька треків – вони упорядкуються за загальною кількістю прослуховувань. Потім трек або кілька треків будуть додані до результуючого списку. Процедура буде повторюватися для кожного треку зі списку нашого користувача. Таким чином, буде сформовано перелік з деякою кількістю записів, відсортованих по кількості прослуховувань у інших користувачів.

Метод фільтрації на основі контенту шукатиме треки, які є схожими на обрані треки користувача (найбільш часто прослуховувалися, прослуховувалися у найближчий час, були додані до списку улюблених) за певними «музичними» характеристиками, а саме: тональністю, ритмом, гармонією, інтенсивністю, структурою та іншими. Пошук відбуватиметься у базі даних сервісу Spotify, де знаходиться велика кількість уже заздалегідь зібраних даних про характеристики величезної кількості музичних треків. Ця база даних була створена та наповнена великою кількістю професійних музикантів, тому немає сумнівів у достовірності та якості рекомендацій, які будуть надані користувачам на основі даних, отриманих з даної бази. Результати будуть отримуватися за допомогою надсилання запитів через відкритий Spotify Web API. Для кожного з треків будуть знайдені та обрані записи з такими ж значеннями характеристик або зі значеннями, що є максимально наближеними до даних. Ці записи будуть включені до результуючого списку рекомендацій.

Після роботи першого методу з блоку одразу починає працювати другий метод. Вхідними даними для нього, як було зазначено раніше, будуть треки, що були запропоновані попереднім методом. Їх кількість заздалегідь невідома та залежить тільки від популярності початково обраних треків серед користувачів системи. Остаточо сформований перелік композицій, який з'являється після роботи другого методу, буде представлений користувачу як кінцевий результат. Таким чином, може виникнути одна з двох наступних ситуацій: коли кількість рекомендованих треків більша, ніж необхідна та коли їх кількість менша. У першому випадку ситуацію зможуть виправити коефіцієнти схожості [4], які призначаються кожному треку-рекомендації у процесі роботи методів. Ці коефіцієнти показують ступінь схожості вихідного треку на вхідний та визначаються на етапі пошуку схожих треків до одного спільного треку за критеріями, властивими поточному методу (популярність треку у інших користувачів для КФ та відмінність від характеристик вхідного треку для контент-орієнтованого методу). Саме 30 треків з найбільшими коефіцієнтами і будуть відображені для користувача. У другому випадку, коли кількість рекомендованих треків менша за необхідну, як компенсація за недостачу юзеру будуть запропоновані популярні у світі треки. Якщо користувач захоче побачити більшу кількість рекомендацій, кількість треків для відображення може бути збільшена.

Таким чином, розроблений автором гібридний метод, що працює за принципом подвійної організації, за рахунок подвійного опрацювання даних та відсіву може вирішити наступні проблеми існуючих рекомендаційних методів, такі як надмірна схожість рекомендацій, потенційно мала кількість або відсутність рекомендацій взагалі, за рахунок компенсації даних з одного блоку даними з іншого. У такого методу існують також певні недоліки, такі як високе споживання обчислювальних ресурсів та зниження швидкості роботи через трудоємність процесів, що дає гарні приводи для його подальшого розвитку та оптимізації. Але описані недоліки можуть бути легко скомпенсовані

підвищенням потужності серверу, на якому працює система або ж встановленням даної програмної системи на іншому, більш потужному сервері.

Список використаних джерел:

1. K. Miyahara and M. J. Pazzani, «Collaborative filtering with the simple Bayesian classifier,» in Proceedings of the 6th Pacific Rim International Conference on Artificial Intelligence, pp. 679-689, 2000.
2. P. Melville, R. J. Mooney, and R. Nagarajan, «Content-boosted collaborative filtering for improved recommendations,» in Proceedings of the 18th National Conference on AI (AAAI '02), pp. 187–192, Edmonton, Canada, 2002.
3. Burke, R.: Hybrid Recommender Systems: Survey and Experiments. UMUAI 12 (4), 331-370. (2002).
4. Van Setten, M.: Supporting People in Finding Information: Hybrid Recommender Systems and Goal-Based Structuring. Report No. 016 (TI/FRS/016). Enschede, the Netherlands: Telematica Instituut. (2005).

Шевченко С.О.

студент,

Науковий керівник: Лєсна Н.С.

кандидат технічних наук, професор,

Харківський національний університет радіоелектроніки

УПРАВЛІННЯ ЕНЕРГОСПОЖИВАННЯМ ОБЧИСЛЮВАЛЬНИХ СЕРВЕРІВ

Перші засоби керування енергоспоживанням електронно-обчислювальних пристроїв/машин були запропоновані ще в 1991-92 роках, а саме технологія BatteryMAX [1] та Advanced Power Management (APM) [2]. Технологія BatteryMAX відслідковувала простої у роботі комп'ютера та виконувала спеціальний код для переходу складових системи в енергозберігаючий режим. Ця технологія була розроблена для портативних комп'ютерів того часу. Пізніше, у 1992 році, Intel та Microsoft запропонували набір функцій для керування енергоспоживанням – APM. Реалізація цього функціоналу була закладена безпосередньо у BIOS (Basic Input/Output System). При збільшенні різноманітності периферійних пристроїв та плат-розширень все складніше було підтримувати керування енергоспоживанням одним лише кодом прошивки BIOS, тому вже в 1996 році був запропонований стандарт ACPI (Advanced Configuration and Power Interface) [3], котрий включив у себе можливості APM.

Відкритий стандарт ACPI описує стани системи в цілому та стани для обладнання. Відповідно до цього стандарту існують: глобальні стани системи, стани для пристроїв, стани центрального процесору та стани швидкодії пристрою.

Глобальні стани позначаються як Gx. Їх лише чотири, а саме:

– G0 – стан нормальної роботи системи;