

Таблиця 1- результати тестування роботи алгоритму із різними вхідними параметрами

Тестування проводилось на комп'ютері із наступними параметрами: процесор AMD Ryzen 4700U 7(8 ядер, 8 потоків, 2.00-4.10 ГГц, кеш 8 Мб), оперативна пам'ять DDR4(2400MHz) 16 гб, периферійна пам'ять M.2 PCIe SSD: 512 Гб. Отже, найточніший маршрут і найшвидша робота алгоритму була досягнута коли швидкість випаровування феромонів становить 0.5, а вплив феромонів 2.

При проведенні дослідження було встановлено, що різні вхідні параметри можуть погіршувати чи покращувати роботу алгоритму. Також, на основі проведеного дослідження, можна стверджувати що мурашиний алгоритм є оптимальним для вирішення задач маршрутизації.

#### Список використаної літератури

1. *Shortest Path Problem* – [Електронний ресурс]. Режим доступу: <https://www.sciencedirect.com/topics/computer-science/shortest-path-problem>
2. *Ant colony optimization theory: A survey* – [електронний ресурс]. Режим доступу: <https://www.sciencedirect.com/science/article/pii/S0304397505003798>
3. *Performance Comparison - C++ / Java / Python / Ruby/ Jython / JRuby / Groovy* – [Електронний ресурс]. Режим доступу: <https://www.cxyzjd.com/article/pj1258/17527257>

УДК 004.01

Бралант Р. А.

Студент 2-го курсу СО «Магістр»

Спеціальності 122 «Комп'ютерні науки»

Баркалов О.О.

професор кафедри інформаційних технологій

## МОДИФІКАЦІЯ РЕКОМЕНДАЦІЙНОГО АЛГОРИТМУ КОЛАБОРАТИВНОЇ ФІЛЬТРАЦІЇ З ВИКОРИСТАННЯМ АЛГОРИТМУ TOMPSON SAMPLING

Донецький національний університет імені Василя Стуса, Україна

Рекомендаційна система – це алгоритми, що пропонують різні товари та послуги на основі отриманих та проаналізованих даних про користувача. Найчастіше вони рекомендаційні алгоритми використовуються у різних комерційних проектах, інтернет магазинах, музикальних та відео сервісах і т. д.

В основі роботи item-item алгоритму колаборативної фільтрації лежить принцип використання user-item матриці. Ця матриця формується за таким принципом: у її рядках записуються користувачі системи, а у стовпцях продукти та категорії продуктів. Пересічення рядків та стовпців матриці містить оцінку,

яку продукт отримав в результаті певних дій користувача. Побудувати таку матрицю можна за допомогою двох методів збору інформації:

Явний.

- Неявний.

Явний спосіб полягає у тому, що користувачеві пропонується залишити відгук, оцінку або анкету про даний продукт. Не всі користувачі у захваті від такої перспективи, тому у даній роботі було прийняте рішення використовувати неявний збір інформації. Тобто система буде сама відслідковувати та аналізувати дії виконані покупцем на сайті. Таким чином буде зібрано більше інформації, що в свою чергу дасть більшу точність рекомендацій.

Після формування user-item матриці потрібно обрати одну з мір схожості, за якими будуть оцінюватися продукти та послуги. Найпоширенішими є наступні міри:

- Коефіцієнт кореляції Пірсона
- Коефіцієнт Тахімото
- Косинусна міра
- Евклідова відстань

Обирати один з запропонованих варіантів потрібно лише виходячи із формату вхідних даних, який ми будемо оцінювати, на роботу алгоритму рекомендації це ніяк не впливає.

Після того як буде проведена оцінка схожості продуктів та послуг буде обрахована вихідна оцінка. Вираховуватись вона буде за наступною формулою:

$$r_{u,i} = k \sum_{u' \in U} sim(u, u') r_{u',i}$$

У даній формулі:

- $r_{u,i}$  – прогнозована оцінка.
- $r_{u',i}$  – відома оцінка схожого продукту або послуги.
- $sim(u, u')$  – ступінь подібності продукту або послуги.

У результаті роботи item-item алгоритм [1] поверне перелік продуктів та послуг посорттованих за ступенем схожості. Для того щоб підвищити якість роботи алгоритму буде також враховуватись оцінка продуктів.

Для того, щоб уникнути повторів рекомендацій кожного разу, коли користувач перезавантажує сторінку, було прийнято рішення додати рандомізацію до уже сформованої матриці. Таким чином, кожного разу, item-item алгоритм буде розбивати масив отриманих посорттованих елементів на масиви що містять по три елементи, та обирати один випадковий продукт із кожного такого масиву, що буде відображено на рекомендаційній каруселі.

Ми розділимо user-item матрицю на три окремі матриці із різними проаналізованими даними. Таким чином пошук буде вестись між трьома масивами даних з їх переліком продуктів. Кожен продукт буде оцінюватися алгоритмом наступним чином:

- Успішний (продався завдяки рекомендації – 1)

- Не успішний (не був проданий завдяки рекомендації – 0).

Таким чином за допомогою розподілу Бернуллі [2] ми можемо описати задачу наступним чином: нам потрібно знайти вибірку даних з трьох запропонованих вище, продукти з якої будуть мати найбільший успіх продаж.

Отже, в результаті роботи семплювання Томпсона [3] ми отримуємо модель ймовірностей винагороди. Коли доступність винагороди є бінарною (як у цьому випадку так чи ні), ідеальною моделлю для такої ймовірності є бета-розподіл. Тому ми введемо два лічильники успіхів та невдач,  $\alpha$  та  $\beta$ . Це будуть змінні, за допомогою яких буде вестись рахунок успіхів та невдач. У бета-розподілу є усереднене значення, яке обчислюється наступним чином:

$$\frac{\alpha}{\alpha + \beta} = \frac{\text{кількість успіхів}}{\text{кількість спроб}}$$

Нам потрібно знайти dataset з найбільшою ймовірністю успіху продажу.

Спочатку ми не знаємо, яка ймовірність успіху продажу у конкретної вибірки продуктів, тому можна присвоїти  $\alpha$  і  $\beta$  значення 1, що дасть нам рівномірний розподіл. Це початкове припущення називається апіорною ймовірністю: ймовірність виникнення якоїсь події до того, як ми отримали будь-які відомості про неї.

Перевіривши dataset на успішність, ми можемо скоригувати наші очікування на те, що він працює. Така нова ймовірність після отримання якихось відомостей називається апостеріорною ймовірністю. Вона теж представлена бета-розподілом, але тепер ми оновили значення  $\alpha$  і  $\beta$  на основі отриманої винагороди.

Якщо dataset успішний, то винагорода дорівнює 1, і альфа-лічильник успіхів – збільшується на 1. Лічильник невдач – не росте. Якщо ж ми не отримаємо винагороду, то  $\alpha$  не зміниться, а  $\beta$  збільшиться на 1. Чим більше ми збираємо даних, тим сильніше бета-розподіл починає відрізнятися від прямої лінії і стає все більш точною моделлю ймовірності усередненої винагороди. Підтримуючи значення  $\alpha$  і  $\beta$ , алгоритм семплювання Томпсона може описати очікувану середню винагороду та рівень його достовірності.

На відміну від жадібного алгоритму, у якому на кожному тимчасовому кроці обирають дію з найвищою очікуваною винагородою, навіть якщо достовірність очікування невисока, семплювання Томпсона бере зразки з бета-розподілу кожної дії, і вибирає дію з найвищим повернутим значенням. Оскільки дії, що рідко аналізуються, мають широкі розподіли, то у них більше діапазон можливих значень. Тому dataset, який має низьку очікувану середню винагороду, але який тестувався рідше за dataset з більш високим середнім очікуванням, може повернути більше вибіркового значення і буде обраний на наступному часовому кроці.

Кількість використання dataset збільшує достовірність його очікуваного середнього. Це відображено у звуженні розподілу ймовірності, і вибіркоче значення буде братися з діапазону значень, які ближчі до реального середнього. В результаті дослідження (explore) зменшується, а використання (exploit)

зростає, тому що алгоритм починає частіше вибирати dataset з вищою ймовірністю отримання винагороди.

Потрібно уникнути повного ігнорування одного з запропонованих масивів даних, для вирішення цієї задачі було прийняте рішення поставити нижню межу для коефіцієнту успішності кожної матриці, що буде рівна 0,3. Таким чином, ми залишаємо можливість залишатися продуктам з невдалої матриці на рекомендаційній каруселі, у меншій кількості, але з надією, що вона у перспективі зможе повернути свою успішність.

Таким чином було розроблено нову рекомендаційну систему, яка вирішує ряд проблем, що присутні у її аналогів, за допомогою додавання нових операцій до уже існуючих алгоритмів та розширення уже існуючого функціоналу.

#### Список літератури:

1. Authors: Peng Hu , Rong Du , Yao Hu and Nan Li - Hybrid Item-Item Recommendation via Semi-Parametric Embedding - [Електронний ресурс] – режим доступу до ресурсу: <https://www.ijcai.org/proceedings/2019/0350.pdf>
2. Розподіл Бернуллі - [Електронний ресурс] – режим доступу до ресурсу: <http://people.umass.edu/~biep540w/pdf/bernoulli.pdf>
3. Authors: Daniel J. Russo , Benjamin Van Roy , Abbas Kazerouni , Ian Osband and Zheng Wen - Thompson Sampling algorithm - [Електронний ресурс] – режим доступу до ресурсу: [https://web.stanford.edu/~bvr/pubs/TS\\_Tutorial.pdf](https://web.stanford.edu/~bvr/pubs/TS_Tutorial.pdf)

**УДК 004.01**

Дужак А. О.,  
студент СО «Бакалавр»,  
Нескородева Т.В д.т.н, доцент  
кафедри інформаційних технологій

## ОСОБЛИВОСТІ РОБОТИ РЕКОМЕНДАЦІЙНИХ АЛГОРИТМІВ YOUTUBE

*Донецький національний університет імені Василя Стуса, м. Вінниця*

**Вступ.** YouTube – популярний відеохостинг, заснований у 2005 році. Станом на березень 2020 року за даними Alexa Company сайт Youtube.com став другим за відвідуваністю у світі та Україні, поступаючись лише Google.com [1,2]. Сайт дозволяє завантажувати, редагувати, дивитися та коментувати відеоролики кожному користувачу. Протягом років свого існування YouTube розширював свій функціонал і ставав все більш універсальною та багатофункціональною інформаційною системою. За даними Hootsuite, станом на 2020 рік платформою користується більше 2 мільярдів чоловік щомісяця, а щохвилини на сайті з'являється 500 годин відеоконтенту.