

ПЕРЕВАГИ ТА НЕДОЛІКИ SMMC АЛГОРИТМУ ПРИ СТВОРЕННІ РЕКОМЕНДАЦІЙ

Вінницький національний технічний університет;

Анотація

Запропоновано метод створення рекомендацій SMMC (Social Matrix Factorization for Collaborative Filtering), що базується на використанні Марковських ланцюгів для музичного інтерактивного бота та оцінити його переваги та недоліки при використанні.

Ключові слова: Python, Telebot, SMMC, музичні рекомендації, алгоритм.

Abstract

The method for creating recommendations SMMC (Social Matrix Factorization for Collaborative Filtering) based on the use of Markov chains for a musical interactive bot and to evaluate its advantages and disadvantages when using.

Keywords: Python, Telebot, SMMC, music recommendations, algorithm.

Вступ

Рекомендаційні системи відіграють ключову роль у персоналізації контенту, зокрема музики. У цій статті розглядається алгоритм музичних рекомендацій SMMC (Social Matrix Factorization for Collaborative Filtering), що базується на Марковських ланцюгах. Такий підхід дозволяє прогнозувати наступний музичний трек, використовуючи ймовірнісні переходи між композиціями з урахуванням попереднього контексту. Ланцюг Маркова [1] в математиці це випадковий процес, що задовольняє властивість Маркова і який приймає скінченну чи зліченну кількість значень або станів. Існують ланцюги Маркова як з дискретним так і з неперервним часом. В даній статті розглядається дискретний випадок.

SMMC алгоритм

1. Марковський ланцюг — це стохастичний процес, який задовольняє властивість Маркова зображений на рисунку 1.

$$P(X_{n+1} = x | X_n = x_n, X_{n-1} = x_{n-1}, \dots, X_1 = x_1) = P(X_{n+1} = x | X_n = x_n)$$

Рисунок 1 – Властивість Маркова

Це означає, що для передбачення наступного стану достатньо знати лише поточний стан.

2. Формування множини станів

Множина станів S формується зі списку доступних музичних треків, які завантажуються з файлу tracks.txt. Кожен трек є окремим станом. На рисунку 2 зображено множину станів.

$$S = \{T_1, T_2, \dots, T_N\}$$

Рисунок 2 – Множина станів

3. Ініціалізація матриці переходів

Якщо історія прослуховування відсутня, алгоритм використовує рівномірний розподіл ймовірностей заданих на рисунку 3.

$$p_{ij} = \frac{1}{N}, \quad \forall i, j$$

Рисунок 3 – Рівномірний розподіл ймовірностей

4. Оновлення матриці переходів

Після кожної рекомендації ймовірності переходів змінюються. Якщо після треку T_i був обраний трек T_j , то ймовірність переходу між ними збільшується. На рисунку 4 вказано формулу збільшення ймовірності переходів.

$$p_{ij} = p_{ij} + 1$$

Рисунок 4 – Збільшення ймовірності переходів

Потім на рисунку 5 відбувається нормалізація всіх ймовірностей, щоб їх сума в кожному рядку дорівнювала 1:

$$p_{ij} = \frac{p_{ij}}{\sum_k p_{ik}}$$

Рисунок 5 – Нормалізація усіх ймовірностей

5. Генерація рекомендації

Наступний трек вибирається випадково, але з урахуванням ймовірностей переходу з поточного стану. Для цього використовується вибірка за ймовірнісним розподілом зображена на рисунку 6.

$$T_{n+1} \sim P(T_{n+1}|T_n)$$

Рисунок 6 – Вибірка треків за ймовірнісним розподілом

По-перше з переваг застосованого алгоритму є адаптивність. Алгоритм змінює ймовірності переходів на основі історії прослуховувань користувача. З кожним новим переходом система оновлює свої оцінки та підлаштовується під індивідуальні вподобання, що дозволяє з часом підвищити релевантність рекомендацій.

По-друге алгоритм простий та ефективний, адже методика базується на обчисленні ймовірностей переходів між треками, що не потребує значних обчислювальних ресурсів. Це дозволяє використовувати алгоритм навіть на пристроях з обмеженими потужностями або в умовах, де важлива швидкість обробки.

По-третє, на відміну від багатьох сучасних моделей машинного навчання, алгоритм SMMC [2] дозволяє безпосередньо аналізувати матрицю переходів. Це дає змогу розуміти логіку вибору рекомендацій, що є важливим для розробників і дослідників. Також модель можна представити у вигляді графа, що дозволяє краще зрозуміти зв'язки між треками. Графічний аналіз допомагає оцінити, які треки найчастіше рекомендуються разом і які мають найбільший вплив на подальший вибір користувачів.

Серед недоліків виділяють те, що алгоритм не враховує довгострокові вподобання, адже алгоритм аналізує лише останній трек при генерації наступної рекомендації. Це означає, що загальні музичні вподобання користувача можуть не враховуватися, якщо в короткостроковій перспективі він слухав різноманітні жанри або випадкові треки. Також значним недоліком є низька персоналізація тому, що алгоритм не аналізує індивідуальні характеристики користувача. Наприклад, він не враховує жанрові вподобання, час прослуховування, улюблених виконавців. Він однаково працює для всіх користувачів, тому може не відповідати специфічним музичним смакам окремої особи.

Висновки

Встановлено, що запропонований підхід дозволяє покращити формування музичних рекомендацій інтерактивним ботом на основі заданих даних і має значну кількість переваг ніж недоліків при використанні.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Теоретичні відомості про ланцюг Маркова та його визначення [Електронний ресурс]. — Режим доступу: https://uk.wikipedia.org/wiki/%D0%9B%D0%B0%D0%BD%D1%86%D1%8E%D0%B3_%D0%9C%D0%B0%D1%80%D0%BA%D0%BE%D0%B2%D0%B0 (дата звернення: 22.03.2025). — Назва з екрана.

2. Purushotham S., Liu Y. Collaborative Topic Regression with Social Matrix Factorization for Recommendation Systems [Електронний ресурс]. — Режим доступу: <https://icml.cc/2012/papers/407.pdf> (дата звернення: 22.03.2025). — Назва з екрана.

Тіслін Олексій Юрійович — студент групи 4ПІ-21б, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: oleksiytislin13@gmail.com

Науковий керівник: Катєльніков Денис Іванович - к.т.н., доцент кафедри програмного забезпечення, Вінницький національний технічний університет, Вінниця, fuzzy2dik@gmail.com.

Tislin Oleksiy Y. — student of group 4PI-21b, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: oleksiytislin13@gmail.com

Supervisor: Katielnikov Denys Ivanovich - Ph.D., Associate Professor of the Department of Software Engineering, Vinnytsia National Technical University, Vinnytsia, fuzzy2dik@gmail.com.