Технічні науки

УДК 004.852

Яковець Михайло Вікторвич

студент

Національний технічний університет України «Київський політехнічний інститут»

Яковец Михаил Викторович

студент

Национальный технический университет Украины «Киевский политехнический институт»

Yakovets M.

student

National Technical University of Ukraine “Kyiv Polytechnic Institute”

ФОРМУВАННЯ РЕКОМЕНДАЦІЙ НА ОСНОВІ МОДЕЛІ ПРИХОВАНИХ ФАКТОРІВ

ФОРМИРОВАНИЕ РЕКОМЕНДАЦИЙ НА ОСНОВЕ МОДЕЛИ СКРЫТЫХ ФАКТОРОВ

FORMATION OF RECOMMENDATIONS BASED ON THE LATENT FACTOR MODEL

Анотація: В статті проведено огляд існуючих підходів до формування рекомендацій. Запропоновано алгоритм формування рекомендацій на основі моделі прихованих факторів. Проведено порівняння запропонованого алгоритму з існуючими алгоритмами.

Ключові слова: машинне навчання, рекомендаційний алгоритм, колаборативна фільтрація, приховані фактори.

Аннотация: В статье проведен обзор существующих подходов к формированию рекомендаций. Предложен алгоритм формирования рекомендаций на основе модели скрытых факторов. Проведено сравнение предложенного алгоритма с существующими алгоритмами.

Ключевые слова: машинное обучение, рекомендательный алгоритм, колаборативних фильтрация, скрытые факторы.

Summary: In the article a review of existing approaches to the formation of recommendations was conducted. Formation of recommendations algorithm based on the model of latent factors was developed. The comparison of the proposed algorithm with existing algorithms was conducted.

Key words: machine learning, recommendation algorithm, collaborative filtering, latent factors.

Вступ. Основне завдання рекомендаційної системи – це надання персоналізованих рекомендацій користувачу, які враховують його уподобання при виборі предметів. Задача підвищення якості рекомендацій важлива тим, що з ростом обсягів даних, які зберігаються у мережі і пропонуються користувачу, зростає необхідність полегшення пошуку потенційно корисної інформації. Крім цього, компаніям, які займаються Інтернет-комерцією, підвищення якості рекомендацій дозволить збільшити продажі. Існує два основних типи алгоритмів формування рекомендацій [5]. Рекомендаційні алгоритми на основі вмісту. Даний підхід заснований на використанні даних з профілів користувачів і даних про об’єкти. Вміст профіля користувача може складатися не лише з історії його покупок чи оцінок об’єктів, але й з великої кількості інших показників: вік, стать і т.д. Аналогічна ситуація з профілями об’єктів. Також існують колаборативні рекомендаційні алгоритми. Є множина користувачів $u \in U$, множина об'єктів $i \in I$ (фільми, треки, товари і т.п.) і множина дій $(r_{ui}, u, i, ...) \in \mathcal{D}$

(дії, які користувачі здійснюють з об'єктами). Кожна дія задається користувачем u, об'єктом i, своїм результатом r_{ui}. Результати формуються у вигляді матриці, наприклад, матриці рейтингів, які користувачі присвоїли об’єктам. Рекомендація формується на основі цієї матриці. Серед колаборативних методів фільтрації виділяють алгоритми на основі пошуку прихованих факторів користувачів і об’єктів. Суть алгоритму полягає в факторизації матриці рейтингів, тобто розбиття її на дві матриці, перемножуючи стопці і рядки яких, можна передбачити значення рейтингів в початковій матриці. [3]

Мета роботи. Провести огляд існуючих підходів до формування списку рекомендацій. Запропонувати алгоритм колаборативної фільтрації на основі моделі прихованих факторів. Проаналізувати роботу розробленого алгоритму і провести його порівняння з іншими алгоритмами колаборативної фільтрації.

Основна частина.

Модель прихованих факторів

Далі розглядатимемо модель і алгоритм на основі даних з предметної області фільмів. Тобто, є множина користувачів $u \in U$, множина фільмів $i \in I$, оцінка, яку користувач поставив фільму r_{ui}. Усі оцінки зручно представляти як матрицю рейтингів.

Нижче описана спрощена модель представлення оцінок, на основі моделі – переможця конкурсу Netflix Prize. Для початкової моделі необхідна більша кількість структурованих даних, як наприклад, час виставлення рейтингу. Всі необхідні дані були надані учасникам в рамках конкурсу Netflix Prize. [4]

Проте, для роботи виключно з рейтингами фільмів, пропонується наведена нижче модель:

$$r_{ui} = \mu + b_i + b_u + \mathbf{q}_i * \mathbf{p}_u ,$$

де r_{ui} – оцінка з матриці рейтингів *R*;

μ – середній рейтинг по вибірці;

b_i – базовий рейтинг користувача;

b_u – базовий рейтинг фільму;

$\mathbf{q}_i$ – вектор факторів фільму;

$\mathbf{p}_u$ – вектор факторів користувача.

b_u – характеризує базовий фактор користувача, як його оцінки відрізняються від середньої по вибірці. Цей фактор впливає на рейтинг незалежно від факторів фільму. Наприклад, якщо користувач ставить усім фільмам хороші оцінки, то базовий фактор буде високим, і – навпаки, якщо користувач жадібний на оцінки.

b_i – характеризує базовий фактор фільму, його відносну якість. Цей фактор впливає на рейтинг незалежно від факторів користувача. Наприклад, якщо фільм краще знятий або більше розкручений, то й значення факторі буде вище, ніж у інших фільмів.

Алгоритм формування рекомендацій на основі моделі прихованих факторів

Якщо передбачити оцінки, які користувач поставить фільмам, то для вдалого формування рекомендацій необхідно у список рекомендацій додавати фільми з найвищими передбаченими оцінками. Щоб передбачити

оцінки, треба знайти чисельні значення прихованих факторі, які формують оцінку.

Для цього по відомим значенням оцінок, які користувачі поставили фільмам (беруться з матриці рейтингів), знайдемо такі значення μ, b_i, b_u, $\mathbf{q}_i$, $\mathbf{p}_u$ $\forall u,i$, які найкраще наближають значення $\mu + b_i + b_u + \mathbf{q}_i * \mathbf{p}_u$ до реальної оцінки, з точки зору квадрату похибки:

$$b_*, q_*, p_* = \arg\min_{b,q,p} \sum_{(i,u)} (r_{i,u} - \mu - b_i - b_u - q_u^T p_i)^2 + \lambda * (b_i^2 + b_u^2 + \|q_u\|^2 + \|p_i\|^2),$$

де λ – регуляризатор.

Регуляризація в машинному навчанні – метод додавання деякої додаткової інформації до умови з метою вирішити некоректно поставлене завдання або запобігти перенавчанню. Ця інформація часто має вигляд штрафу за складність моделі.

Для пошуку невідомих параметрів використано метод градієнтного спуску [2]. На кожному елементі вибірки невідомі параметри змінюються за даними формулами:

$$b_i = b_i + \gamma * (e_{iu} - \lambda b_i),$$

$$b_u = b_u + \gamma * (e_{iu} - \lambda b_u),$$

$$q_{uj} = q_{uj} + \gamma * (e_{iu}\, p_{ij} - \lambda q_{uj}),$$

$$q_{ij} = q_{ij} + \gamma * (e_{iu}\, p_{ij} - \lambda q_{uj})$$

Під час проходження усієї вибірки з даними, рахується помилка:

$$err = \mu - b_i - b_u - q_u^T p_i,$$

яка потім підноситься в квадрат і додається. Тобто після проходження вибірки підраховується RMSE. Якщо новее значення RMSE

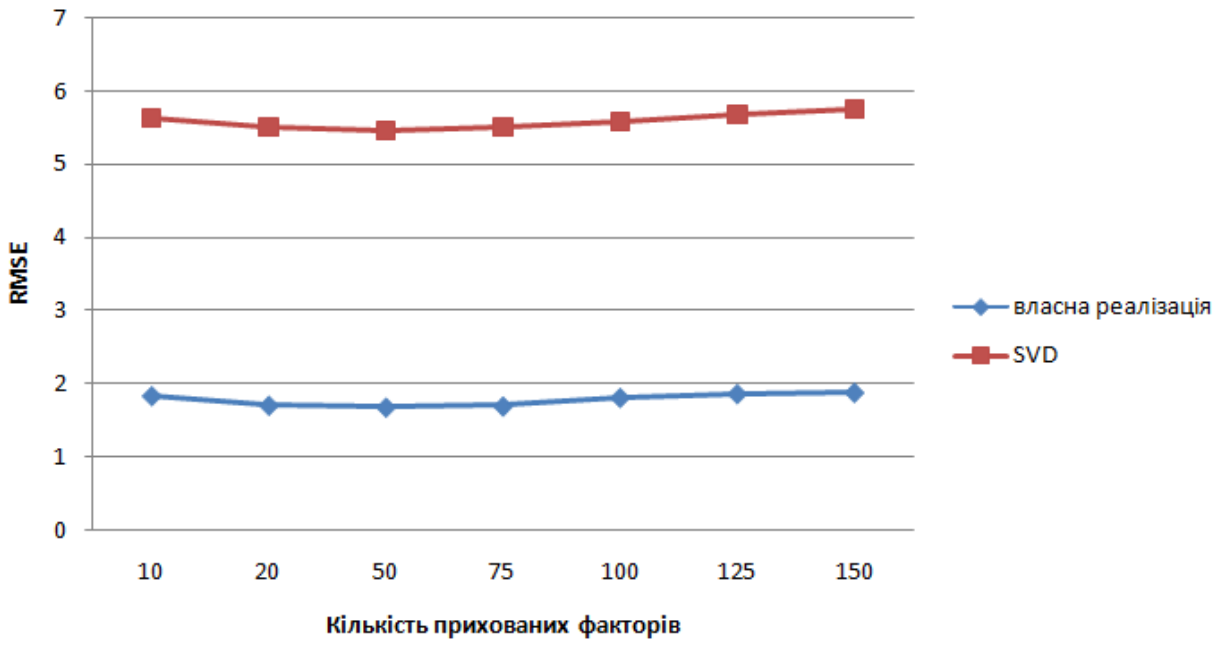
зменшується мало, то темп навчання збільшується вдвічі. Критерієм завершення навчання є зміна RMSE за цикл навчання менше, ніж на epsilon.

Порівняння запропонованого алгоритму з іншим алгоритмом колаборативної фільтрації

В ході експерименту, алгоритм навчався на начальній вибірці , в якій знаходилось 1.6 мільйонів оцінок користувачів, а перевірявся на тестовій вибірці розміром в 400 тисяч оцінок. Параметри алгоритму: регуляризатор – 2.5, epsilon – 0.000001.

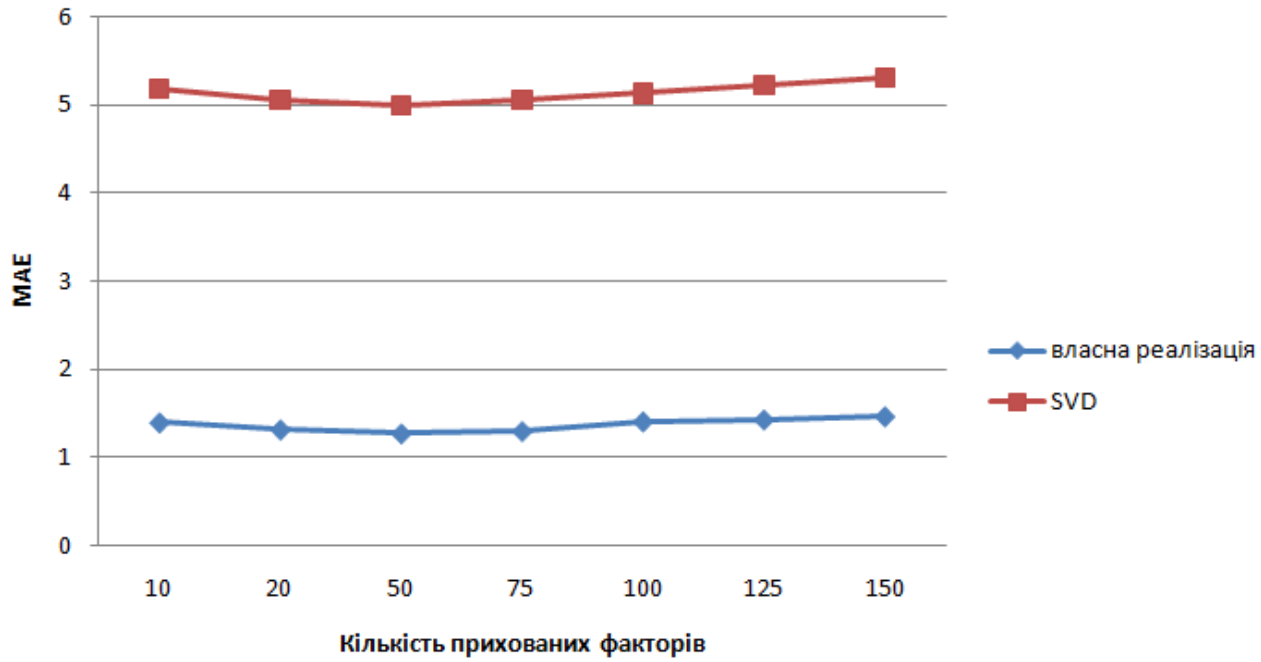
Запропонований алгоритм було порівняно із алгоритмом колаборативної фільтрації з бібліотеки python-recsys [1], в основі якого лежить метод SVD. Якість роботи обох алгоритмів було оцінено за метриками RMSE І MAE.

На графіку 1 зображено зміну оцінки похибки RMSE в залежності від кількості прихованих факторів для алгоритму SVD і власної реалізації:



Графік 1 – Порівняння похибки за RMSE

На графіку 2 зображено зміну оцінки похибки MAE в залежності від кількості прихованих факторів для алгоритму SVD і власної реалізації:



Графік 2 – Порівняння похибки за MAE

Висновки. Стрімке зростання кількості даних різної природи, доступної користувачам, породжує проблему пошуку релевантної інформації. В даній роботі розв'язувалася задача побудови і аналізу алгоритму формування рекомендацій, який збільшує якість рекомендацій за обраними критеріями.

В роботі отримані наступні результати: проведено порівняння існуючих підходів до формування списку рекомендацій; запропоновано алгоритм колаборативної фільтрації на основі даних про рейтинги фільмів; проаналізовано роботу розробленого алгоритму. Власна реалізація алгоритму пошуку прихованих факторів показала кращі результати точності передбачення оцінок користувачів, ніж алгоритм SVD (з бібліотеки python-recsys) за критеріями RMSE і MAE.

Література:

1. Python-recsys on Github [Електронний ресурс]. – Режим доступу: https://github.com/ocelma/python-recsys

2. Вікіпедія – Метод стохастичного градієнта [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Метод_стохастичного_градієнта

3. Y. Koren, R. Bell, C. Volinsky, Matrix Factorization Techniques for Recommender Systems. IEE Computer Society, 2009.

4. R. Bell, Y. Koren and C. Volinsky. The BellKor Solution to the Netflix Prize. 2007.

5. Рекомендательные системы: Часть 1. Введение в подходы и алгоритмы [Електронний ресурс]. – Режим доступу:https://www.ibm.com/developerworks/ru/library/os-recommender1/