Технічні науки

УДК 004.852

Мазурік Олексій Юрійович
студент
Національний технічний університет України «Київський політехнічний інститут»
Мазурик Алексей Юрьевич
студент
Национальный технический университет Украины «Киевский политехнический институт»
Mazurik O.
student
National Technical University of Ukraine “Kyiv Polytechnic Institute”

ПОКРАЩЕННЯ РЕЗУЛЬТАТІВ РОБОТИ РЕКОМЕНДАЦІЙНИХ СИСТЕМ ЗА ДОПОМОГОЮ АЛГОРИТМУ SVD

УЛУЧШЕНИЕ РЕЗУЛЬТАТОВ РАБОТЫ РЕКОМЕНДАЦИОННЫХ СИСТЕМ ПРИ ПОМОЩИ АЛГОРИТМА SVD

IMPROVEMENT OF RECOMMENDATION SYSTEMS WITH SVD ALGORITHM

Анотація: Досліджені базові проблеми рекомендаційних систем, їх типи та різниця між ними. Також наведено опис алгоритму SVD та досліджено результати його роботи.

Ключові слова: рекомендаційні системи, прогнозування, комп’ютерні науки, комп’ютерні технології.

Аннотация: Исследованы базовые проблемы рекомендационных систем, их типы и разница между ними. Также приведено описание алгоритма SVD и исследованы результаты его работы.

Ключевые слова: рекомендационные системы, прогнозирование, компьютерные технологии, компьютерные науки.

Summary: Basic problems of recommendation systems, basic types of it and difference between them were investigated. Also there is description of modern algorithm SVD and results of its work.

Key Words: recommendation systems, forecasting, computer science.

Рекомендаційні системи з'явилися на сучасному ринку IT як механізм для заміни статичному списку рекомендацій при пошуку або покупках на веб-сайтах. Ці системи формують рейтинговий перелік об'єктів (товарів, фільмів, музичних композицій) на основі різних критеріїв: релевантність, популярність, історія оцінок тощо.

При розробці рекомендаційних систем зазвичай розробники стикаються з рядом проблем прогнозування:

- *Розрідженість даних* (більшість користувачів не ставить оцінки товарам, отже дані з попередніми оцінками являють собою розріджену матрицю).
- *Холодний старт* (робота з новими користувачами або товарами).
- *Синонімія* (проблема розпізнавання схожих товарів з різними назвами). [2]
- *Шахрайство* (цілеспрямоване завищення рейтингів певних товарів їх власниками).
- *Розмаїття* (при великій вибірці нові або маловідомі товари мають низькі позиції в рейтинговому списку).
- *Білі ворони* (унікальні користувачі, смаки яких дуже важко обробити, оскільки вони не співпадають зі смаками відокремлених типів). [3]

Базові підходи розробки рекомендаційних систем

Колаборативна фільтрація

Це один з методів прогнозу в рекомендаційних системах, який використовує відомі переваги (оцінки) групи користувачів для прогнозування невідомих переваг (оцінок) іншого користувача. За

допомогою цього алгоритму будується певна таблиця користувачів, які групуються за схожістю, та прогнозуються результати для інших користувачів.

Наприклад, маємо декількох користувачів порталу з музикою. Всіх користувачів можна поділити на групи за їх смаками (одним подобається джаз, іншим - рок). За цією інформацію в середині кожної групи можна виділити найпопулярніші хіти, які користувачі слухають більше всього. Отже, кожному учаснику певної групи будуть рекомендовані популярні композиції, які ним не були ще прослухані. [5]

Переваги: швидка робота алгоритмів (K-based та ін.) - мала кількість ітерацій; прості в реалізації.

Недоліки: холодний старт; нема що рекомендувати новим або нетиповим користувачам; розріджені матриці оцінок (іноді неможливо зробити прогноз); шахрайство.

Фільтрація вмісту (контенту)

Цей тип алгоритмів прогнозування базується на моделі об’єкту, оцінки якого будуть прогнозуватися. Для кожного об’єкту буде побудовано математичну модель з використанням конкретних характеристик товару (параметри моделі). Рекомендації будуть базуватися на порівнянні характеристик поточного товару та власне характеристик користувача (інформація, яка міститься в профілі користувача).

Для прикладу візьмемо сайт з онлайн-кінотеатром. Нехай маємо користувача, який передивлявся наступні фільми: “Міцний горішок” (бойовик), “Скайфолл” (бойовик, триллер). Висувається гіпотеза, що цьому користувачу подобаються фільми з жанром бойовик, тому логічно будуть створені рекомендації фільмів жанру бойовик. [5]

Переваги: більш точний результат; немає проблеми холодного старту, оскільки рекомендації базуються на моделі об’єкта, а не на попередніх оцінках користувачів.

Недоліки: “затратне” створення моделі (її побудова досить складна), невисока швидкодія алгоритмів (багато обчислень); втрата точності при скороченні параметрів моделі.

Гібридні системи

Даний тип алгоритмів поєднує в собі підходи колаборативної та content-based фільтрації. Цей підхід найбільш популярний при розробці рекомендаційних систем для комерційних сайтів, так як його було створено щоб подолати проблеми колаборативної фільтрації, а також покращити якість прогнозування оцінок конкретної моделі.

Переваги: велика швидкодія; кращі результати.

Недоліки: дуже дорога розробка рекомендаційної системи, оскільки реалізація цього типу алгоритмів дуже складна; важко підтримувати, оскільки навіть незначні зміни в роботі призводять до змін роботи алгоритму.

В наш час найпопулярнішими алгоритмами для рекомендаційних систем виявилися підходи, засновані на колаборативній фільтрації, та content-based системи. Майже всі вони мають такі недоліки, як холодний старт, тривіальність результатів рекомендацій тощо. Одним з нових алгоритмів, який майже не має звичайних проблем для заснованих на сусідстві підходів, виявився гібридний SVD алгоритм, який було створено саме для покращення результатів звичайних алгоритмів.

Постановка завдання

Маємо множину користувачів $u \in U$, множину об’єктів (фільми, треки, товари тощо) $i \in I$, та множину подій (дії, які користувачі виконують над об’єктами) $(r_{ui}, u, i, \dots) \in D$. Кожна подія задається користувачем u, об’єктом i, своїм результатом r_{ui}, а також, можливо, ще іншими характеристиками. Наша ціль:

- передбачити перевагу:

$$\hat{r}_{ui} = Predict(u, i, \dots) \approx r_{ui}.$$

- персональні рекомендації:

$$u \to (i_1, \dots, i_K) = Recommend_K(u, \dots).$$

- схожі об’єкти:

$$u \to (i_1, \dots, i_M) = Similar_m(i, \dots).$$

Таблиця 1. Оцінки користувачів

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
User 1	5	4	5			
User 2	4		5			
User 3		3	5		4	
User 4				3	4	
User 5			4	2	4	
User 6	3					5

Таблиця 2. Завдання для прогнозування

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
User 1	5	4	5			
User 2	4	?	5			
User 3		3	5	?	4	
User 4			?	3	4	
User 5	?		4	2	4	?
User 6	3					5

Основна ідея колаборативної фільтрації: схожим користувачам зазвичай подаються схожі об’єкти. Найпростіший метод вирішення даної задачі:

- Оберемо деяку умовну міру схожості користувачів за їх історією оцінок $sim(u, v)$.
- Поділимо користувачів на групи (кластери) таким чином, щоб схожі користувачі знаходилися в одному кластері: $u \to F(u)$.
- Оцінку користувача об’єкту будемо передбачувати як середню оцінку кластера для цього об’єкта:

$$\widehat{r_{ui}} = \frac{1}{|F(u)|} \cdot \sum_{v \in F(u)} r_{vi}$$

Проблеми алгоритму:

- Рекомендація новим користувачам. Для таких користувачів не знайдеться відповідного кластера зі схожими на них користувачами.
- Рекомендація для нетипового користувача. Ми ділимо усіх користувачів на якісь класи (шаблони).
- Якщо в кластері ніхто не оцінив об'єкт, то зробити передбачення не вийде.

Два підходи, щодо покращення роботи алгоритму:

User-based

Підхід, в якому ми опираємося на припущення, що вподобання користувача схожі на вподобання схожих користувачів. Замінемо жорстку кластеризацію на формулу:

$$\widehat{r_{ui}} = \underline{r_u} + \frac{\sum_{v \in U_i} sim(u,v)\left(r_{vi} - \underline{r_v}\right)}{\sum_{v \in U_i} sim(u,v)}$$

Проблеми підходу:

- Нові/нетипові користувачі.
- Проблема розмаїття [1]

Item-based

Підхід, в якому ми вважаємо, що користувачу сподобаються схожі товари з тим, що він вже обрав. Жорстка кластеризація буде замінена наступним чином:

$$\widehat{r_{ui}} = \underline{r_i} + \frac{\sum_{j \in I_u} sim(i,j)(r_{uj} - \underline{r_j})}{\sum_{j \in I_u} sim(i,j)}$$

Недоліки:

- Холодний старт.
- Рекомендації часто тривіальні. [1]

Алгоритм SVD

SVD (Singular Value Decomposition) - сингулярне розкладання матриці. Теорема про сингулярний розклад стверджує, що у будь-якої матриці A розміру n на m існує розкладання в добуток трьох матриць: U, Σ та V^t:

$$\underset{n \times m}{A} = \underset{n \times n}{U} \times \underset{n \times m}{\Sigma} \times \underset{m \times m}{V^T}$$

Матриці U і V - ортогональні, а Σ - діагональна (хоч і не квадратна).

$$UU^T = I_n, VV^T = I_m$$

$$\Sigma = diag(\lambda_1, \dots, \lambda_{\min(n,m)}), \lambda_1 \geq \cdots \geq \lambda_{\min(n,m)} \geq 0$$

Лямбди в формулі розташовані за спаданням. Доведення теореми буде опущене.

Окрім звичайного розкладання існує ще усічене, коли ми з усіх лямбд залишаємо лише перші $\boldsymbol{d}$ чисел, а інші вважаємо рівними нулю.

$$\lambda_1, \dots, \lambda_{\min(n,m):=0}$$

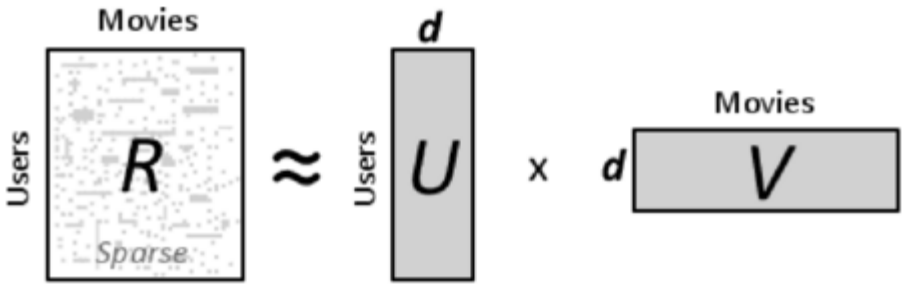
Це рівносильно тому, що в матрицях $\boldsymbol{U}$ і $\boldsymbol{V}$ ми залишаємо лише **d** стовпців, а матрицю Σ обрізаємо до квадратної розмірністю $\boldsymbol{d \times d}$.

$$\underset{n \times m}{A'} = \underset{n \times d}{U'} \times \underset{d \times d}{\Sigma'} \times \underset{d \times m}{V'^T}$$

Виявляється, що на практиці нова матриця A' дуже добре наближає вихідну матрицю $\boldsymbol{A}$ та, тим більше, є найкращим наближенням. [4]

SVD для рекомендацій

Спростимо трохи отриману формулу, вважаючи добуток перших двох матриць за одну:



$$\widehat{r_{ui}} = f(p_u, q_i)$$

Тобто отримано наступний алгоритм: щоб передбачити оцінку користувача *U* для фільма *V*, ми беремо деякий вектор ***p*** (набір параметрів користувача) та вектор ***q*** (набір параметрів фільму). Їх скалярний добуток і буде необхідним передбаченням.

Наприклад, у векторі користувача на першому місці буде стояти параметр, який відповідає за стать користувача (хлопчик чи дівчинка), а на другому - його вік. У фільмів на аналогічному першому місці буде стояти параметр, який вказує на те, чи цей фільм подобається більше хлопчикам/дівчатам, а інший - для якої вікової категорії цей фільм більше підходить. Тому ми бачимо, що цей алгоритм дозволяє не тільки передбачувати оцінки. Також ми можемо передбачувати скриті інтереси користувачів та параметри об’єктів.

Але все виявляється не так просто. По-перше, нам не відома повністю матриця оцінок ***R***, а, по-друге, розклад матриці на добуток не єдиний, тому не факт, що на першому місці вектора ***p*** буде стояти параметр, що відповідає за стать.

Подолати ці проблеми допоможуть сучасні методи оптимізації та машинне навчання.

Ми не знаємо матрицю оцінок, тому однозначний SVD розклад знайти неможливо. Але створити рекомендаційну систему, яка буде працювати схожим чином - можна. Отже, ми маємо знайти деякий вектор користувача та вектор фільму з різними параметрами. Оскільки ми маємо попередні оцінки користувачів їх можна використати як навчальну вибірку. А для знаходження результатів, близьких до існуючих, ми маємо обрати оптимальні параметри моделей фільмів та користувачів. Для покращення машинного навчання результатів часто використовують регуляризатори.

Пошук оптимальних параметрів можна прискорити використавши вже відомі алгоритми градієнтного спуску та метод найменших квадратів.

[4]

Додаткові властивості рекомендацій

Виявляється, що на сприйняття рекомендацій досить часто впливає не лише якість ранжування результатів. Також дуже важливими є інші фактори: різноманітність, несподіваність, новизна та багато інших. Тож при розробці рекомендаційної системи необхідно обов'язково мати на увазі ці фактори.

Висновки

Рекомендаційні системи широко використовуються в мережі Інтернет для різних цілей: збільшення часу перебуття на сайті користувачами, різноманітність вибору тощо. Все це веде до збільшення прибутків підприємства, яке використовує рекомендаційну систему. Зараз існує величезна купа алгоритмів, які дозволяють покращити якість рекомендацій. Але це проблема прогнозування, тому однозначної відповіді який алгоритм краще працює немає. Тому є сенс експерементувати з різними параметрами рекомендацій, типами алгоритмів, тестувати їх роботу на користувачах. Саме таким шляхом можна досягти найкращих результатів.

В результаті аналізу алгоритмів для рекомендації непогані результати показав алгоритм SVD, оскільки це гібридний алгоритм, який базується на основних підходах до розробки рекомендаційних систем: колаборативній фільтрації та content-based алгоритмах. Тому він має менше недоліків, ніж ці алгоритми взяті окремо. Основною проблемою, звичайно, є проблема холодного старту, оскільки не можна навчити нейронну мережу коректно передбачувати оцінки не маючи тестової вибірки. Але в цілому це непоганий алгоритм.

Література:

1. Melville P.,Mooney R., Nagarajan R. (2002). Content-Boosted Collaborative Filtering for Improved Recommendations. Austin, TX, USA: University of Texas, USA. p1-6.

2. Linden G., Smith B., and York J. (2003). Item-to-Item Collaborative Filtering. Los Alamitos, CA, USA: IEEE Internet Computing. p76-80.

3. Sarwar B., Karypis G., Konstan J., and Riedl J. (2001). WWW10 Conference Materials / Item-Based Collaborative Filtering Recommendation Algorithms. Hong Kong: WWW10. p285-289.

4. William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery (1992). Numerical Recipes in C / Singular Value Decomposition. 2nd ed. New York: Cambridge University Press. p59-71.

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