Розробка системи для виявлення та доповнення вподобань користувача

Виконав: студент IV курсу НТУУ «КПІ» ННК «ІПСА»
Групи КА-25
Телещук Ігор
Науковий керівник: ас. Кухарєв С.О.

Актуальність роботи

Рекомендаційні системи порівнюють однотипні дані від різних людей і розраховують список рекомендацій для конкретного користувача. При цьому існуючим рекомендаційним системам притаманні деякі проблеми, а саме проблема неповноти даних , “холодного старту” , та необхідність зберігати величезну матрицю вподобань у пам'яті в результаті чого значно погіршується якість цих рекомендацій .

Постановка задачі

- провести аналіз предметної області;
- зібрати та обробити дані, необхідні для прогнозування ;
- спрогнозувати та заповнити пропуски у матриці вподобань ;
- проаналізувати результати порівнявши з реальнми даними;
- розглянути проблеми рекомендаційних систем та запропонувати варіанти їх вирішення;
- скорегувати алгоритм та провести прогнозування на ньому;
- порівняти поліпшені результати з реальними даними.

- **Мета** дослідження полягає у визначенні найкращих алгоритмів для указання найбільш точних рекомендацій для користувача.
- **Об’єктом** дослідження є дані, оброблені натуральним шляхом та отримані з відкритих ресурсів мережі Інтернет.
- **Предметом** дослідження є рекомендаційні алгоритми.
- Для виконання мети та завдань дипломної роботи були використані наступні **алгортми** роботи : user-based модель, item-based модель , метод k найближчих сусідів.

Колаборативна фільтрація

Кореляційні моделі(Memory-Based Collaborative Filtering)

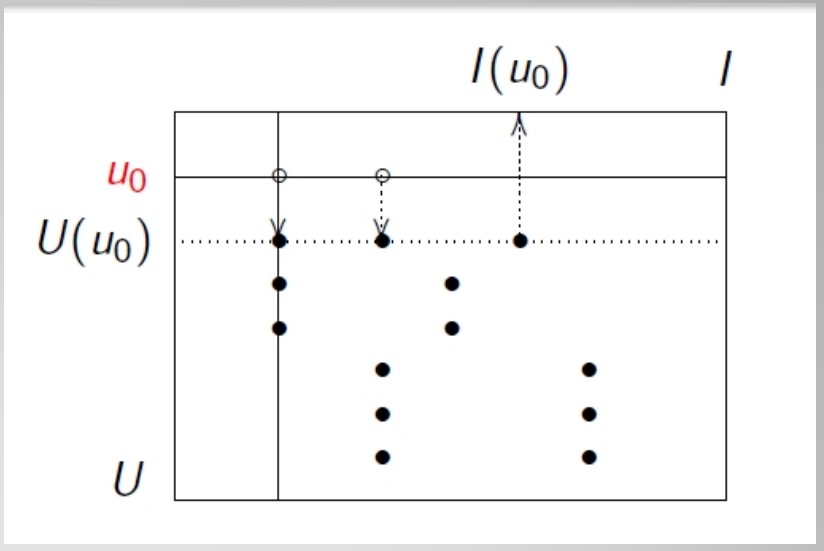
- Зберігання всієї матриці даних
- Подібність клієнтів - кореляція рядків матриці
- Подібність обєктів - кореляція стовпців матриці

Латентні моделі(Latent Models for Collaborative Filtering)

- Оцінювання профілей клієнтів та обєктів
- Зберігання профілей замість зберігання всієї матриці
- Подібність клієнтів та обєктів – подібність їх профілей

USER-BASED MODEL

Ідея базується на тому , що клієнти , які схожі на u_0, також купляли $I(u_0)$.



Необхідно разрахувати міру схожості між двома користувачами

$$w_{i,j} = \frac{\sum_a (r_{i,a} - \bar{r}_i)(r_{j,a} - \bar{r}_j)}{\sqrt{\sum_a (r_{i,a} - \bar{r}_i)^2} \sqrt{\sum_a (r_{j,a} - \bar{r}_j)^2}}$$

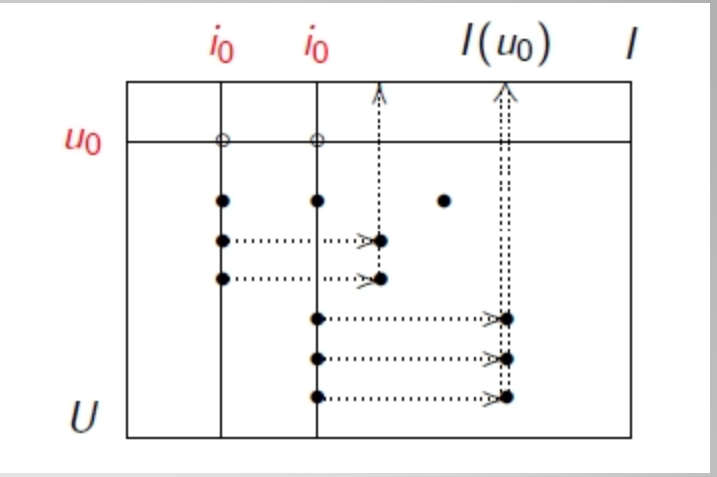
USER-BASED MODEL

Недоліки :

- Тривіальні рекомендації
- Не враховуються интереси конкретного користувача
- Проблема холодного старту
- Необхідність зберігати всю матрицю

ITEM-BASED MODEL

Ідея базується на тому , що разом з обєктами, які купляв u_0, часто купляють $I(u_0)$.



Недоліки ті ж самі , що і у user-based моделі.

Прогнозування рейтингу

$$\hat{r}_{i,a} = \bar{r}_i + \frac{\sum_j (r_{j,a} - \bar{r}_j)\, w_{i,j}}{\sum_j |w_{i,j}|}$$

, де сам прогноз складається з суми середнього рейтингу та зваженого рейтингу реальних користувачів.

Але виникає проблема того , що якщо даних надзвичайно багато , то кожного разу ми повинні рахувати суму по всіх користувачах.

Прогнозування рейтингу

На допомогу приходить метод к найближчих сусідів , що значно зменшує кількість доданків , адже ми не повинні рахувати суму по всіх користувачах , а лише тих , що знаходяться “близько” до користувача.

$$\hat{r}_{i,a} = \bar{r}_i + \frac{\sum_{j \in \mathrm{kNN}(i)} (r_{j,a} - \bar{r}_j)\, \mathrm{w}_{i,j}}{\sum_{j \in \mathrm{kNN}(i)} |\, \mathrm{w}_{i,j}\,|}$$

Метрики

- Коефіц.кореляції Пірсона

$$r_{xy} = \frac{\sum_{i=1}^{m} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{m} (x_i - \bar{x})^2 \sum_{i=1}^{m} (y_i - \bar{y})^2}} = \frac{cov(x,y)}{\sqrt{s_x^2 s_y^2}},$$

де $\bar{x}, \bar{y}$ — вибіркові середні x^m і y^m, s_x^2, s_y^2 — вибіркові дисперсії, $r_{xy} \in [-1, 1]$.

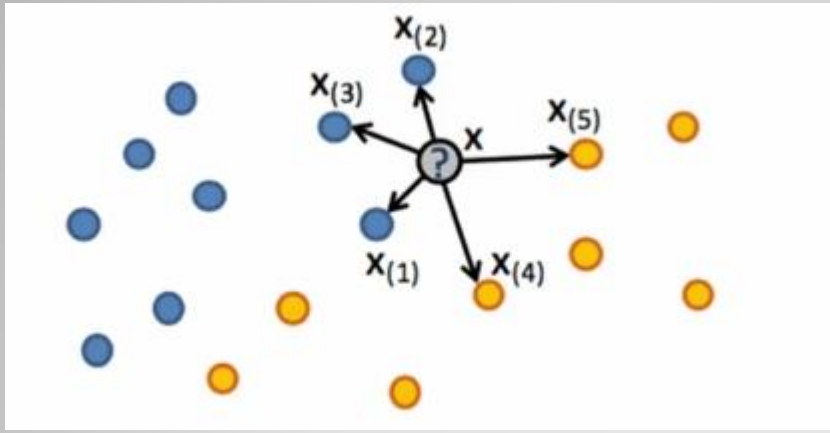
- Евклідова відстань

$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \ldots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}.$$

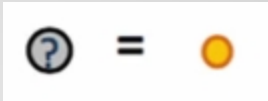
- Манхеттенська відстань

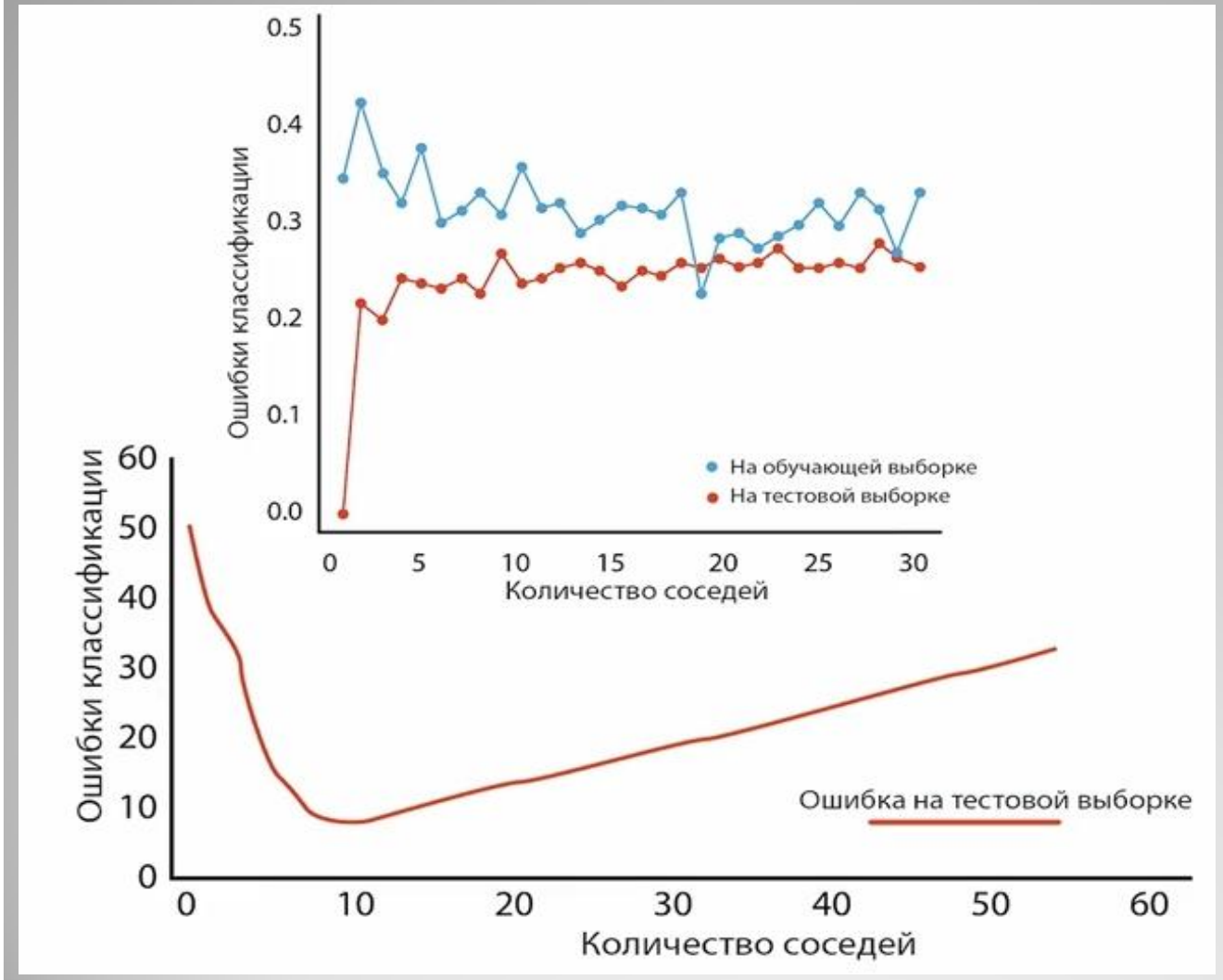
$$d_1(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|_1 = \sum_{i=1}^{n} |p_i - q_i|$$

Метод k найближчих сусідів



Отримуємо :





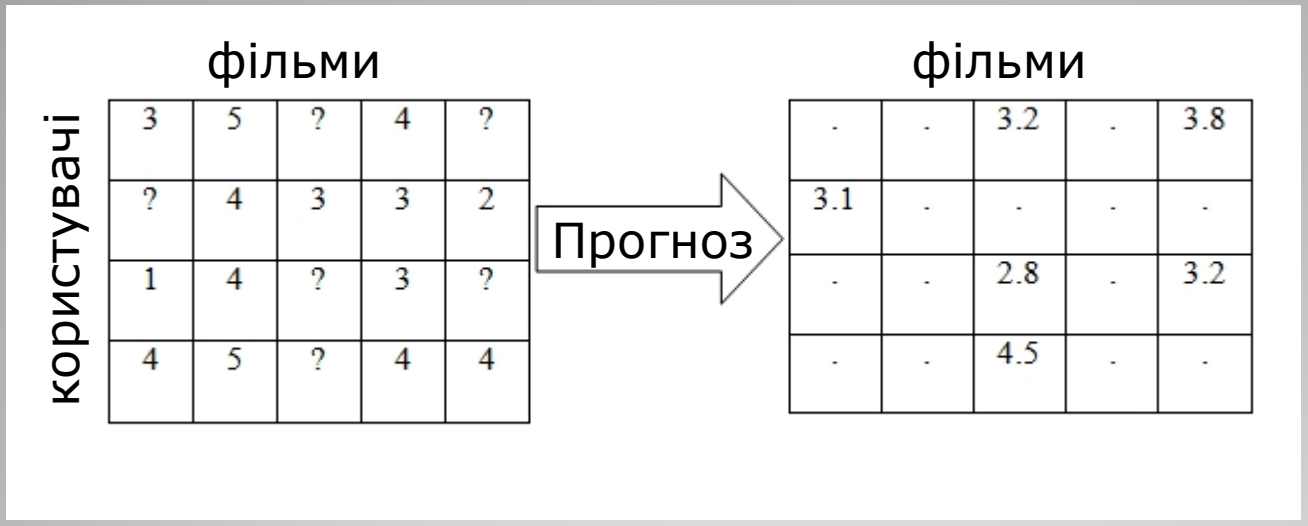
У якості помилки класифікації я використав показник RMSE.

Вхідні дані :

Ми маємо оцінки , які користувачі виставили фільмам , котрі вже подивились.

Задача :

- Необхідно спрогнозувати оцінки , які користувачі виставили б іншим фільмам.
- Порукомендувати те , що їм сподобається.



Дані представлено в у вигляді звязки з 4х параметрів :
Movie#
Customer#,Rating,Date of Rating
Customer#,Rating,Date of Rating
Customer#,Rating,Date of Rating
Example
4:
1065039,3,2005-09-06
1544320,1,2004-06-28
410199,5,2004-10-16

Дякую за увагу.