УДК 519.8

Т.В. Бесчастний, О.Д. Фірсов
Дніпропетровський національний університет імені Олеся Гончара

СИСТЕМА НЕПРЯМОЇ ОЦІНКИ ПАРАМЕТРІВ ВЗАЄМОДІЮЧИХ ОБ'ЄКТІВ ДЛЯ ПРОЕКТУ NETFLIX

Побудована ефективна система рекомендацій для проекту компанії NetFlix у рамках сучасних обчислювальних можливостей. Результатом є аналіз існуючих результатів по проблематиці, створення програмного фрэймворка для умовного алгоритму рекомендацій, розробка та реалізація удосконаленого алгоритму рекомендацій, обчислювальний експеримент на тестових даних пропонованих NetFlix. Також проведений кількісний і якісний аналіз роботи алгоритму.

Построена эффективная система рекомендаций для проекта компании NetFlix в рамках современных вычислительных возможностей. Результатом являются анализ существующих результатов по проблематике, создание программного фрэймворка для условного алгоритма рекомендаций, разработка и реализация усовершенствованного алгоритма рекомендаций, вычислительный эксперимент на тестовых данных предлагаемых NetFlix. Также проведен количественный и качественный анализ работы алгоритма.

The research suggests the optimal choice of recommendation system algorithm with taking into account scale of the NetFlix problem and restrictions of modern computers. The results of this work are: analysis of existing results, creation of the experimental program framework for the common recommendation system algorithm, implementation of the chosen recommendation system, calculation of the implemented algorithms on the test data supplied by NetFlix is performed. Qualitative and quantitative analysis of the algorithms work is done.

Ключові слова: система рекомендацій, колаборативна фільтрація, аналіз за змістом, метод k-найближчих сусідів.

Вступ. Останні часи спостерігається бурхливий ріст у теоретичних та практичних дослідженнях в галузі систем рекомендацій, оскільки актуальність цих систем для ринку IT важко переоцінити. Ця галузь зараз починає відігравати не менш важливу роль, ніж галузь пошуку інформації. Багато IT-гігантів, таких як Amazon, NetFlix, TiVo, iTunes, CDNOW, Musicmatch, Everyone's Critic, USENET, та інші використовують системи рекомендацій, як потужний інструмент для більш повного та індивідуального задоволення потреб користувачів, завдяки більш спрямованої реалізації своїх послуг.

Системи рекомендацій мають різне застосування, наприклад: Amazon – рекомендує покупки багатьох видів товарів, NetFlix займається прокатом DVD.

Задача, що запропонувала компанія NetFlix, була взята з предметної галузі її бізнесу – прокату DVD фільмів та їх рекомендацій. Вона призвела до значного підйому інтересу до галузі систем рекомендацій – розроблено багато різних алгоритмів і побудовано багато моделей для дослідження і використання. Специфіка поставленої NetFlix задачі також пов'язана з розмірністю даних: у системі є 17770 фільмів, 480 189 користувачів і більш ніж 100 000 000 оцінок.

Аналіз існуючих результатів. Насам перед слід зазначити відсутність строгої теорії побудови систем рекомендацій. У літературі можно знайти навіть різні визначення поняття системи рекомендацій. Наприклад, наступні.

Система рекомендацій – система, що надає рекомендацію щодо відповідності об'єкту вимогам користувача, за попередньою історією оцінювання та іншою інформацією [2].

Системи рекомендацій аналізують інтерес користувача в об'єкті, задля надання індивідуальної рекомендації [3].

Система рекомендацій повинна [9]:

– Видавати користувачу список об'єктів, які йому, ймовірно, сподобаються. При цьому користувач надає список об'єктів, які йому подобаються і після цього система аналізує свою внутрішню базу.
– Здійснювати збір і збереження даних про смак користувачів.
– У процесі роботи виконувати самонавчання (автоматично регулювати свої внутрішні параметри, за якими вона оцінює ймовірність того, що даний об'єкт сподобається даному користувачу).

Теоретична база систем рекомендацій торкається таких дисциплін, як алгебра, нейронні мережі, теорія оптимізації, теорія прийняття рішень, психологія та інші.

З огляду на різноманітність галузей використання, різні моделі систем рекомендацій достатньо сильно відрізняються. Необхідність дослідження різних методів і моделей диктується ще й тим, що ефективність багатьох методів розв'язання значно залежить від індивідуальних особливостей задачі.

Загальна постановка задачі, приведена в [3] виглядає наступним чином:

Нехай $U=\{u_j,\ j=1,...,m\}$ – це множина користувачів, а $I=\{i_j,\ j=1,...,n\}$– множина об'єктів, з якими працюють користувачі. Кожен користувач, здатен давати оцінку (рейтинг) r_{ui} об'єкта, що він переглянув.

Кожен користувач u_j надає список об'єктів з оцінками. Система рекомендації у відповідь повинна надавати список об'єктів, які мають потенційно сподобатися іншому користувачу (з відповідною потенціальною оцінкою).

Введемо матрицю

$$R=\{r_{ui},\ u\in U,\ i\in I\}, \quad (1)$$

де елемент r_{ui} – оцінка користувача u об'єкта i. Взагалі кажучи, частина елементів матриці – ненульова (відомі оцінки), а частина – нульова (невідомі оцінки).

Введемо множину

$$V=\{(u,i),\ r_{ui}\neq 0,\ u\in U,\ i\in I\} \quad (2)$$

– індексів відомих оцінок, та множину відомих оцінок

$$R_v=\{r_{ui},\ (u,i)\in V\}. \quad (3)$$

Усі елементи r_{ui} множини R_v відомі з матриці R.

Нехай

$$N=\{(u,i),\ r_{ui}=0,\ u\in U,\ i\in I\} \quad (4)$$

– множина індексів невідомих оцінок. Множину невідомих оцінок будемо позначати

$$R_N=\{r_{ui},\ (u,i)\in N\}. \quad (5)$$

Усі елементи R_N – невідомі. Елемент $r_{ui}\in R_N$ – точна оцінка користувача u, яку він ще не поставив об'єкту *i*.

Позначимо через

$$\bar{R}_N=\{\bar{r}_{ui},\ (u,i)\in N\} \quad (6)$$

– множину потенційних (наближених) оцінок користувачів, що вони поставили би б ще не переглянутим об'єктам.

Отже задача системи рекомендацій – це за відомою множиною R_v оцінити значення елементів R_N – побудувати множину $\bar{R}_N$. У [2] надана загальна класифікація методів і підходів у галузі систем рекомендації. Автори виділяють три основних підходи (моделі), щодо побудови алгоритмів.

Аналіз за змістом (content-base analysis). У літературі цей підхід також називається частковою фільтрацією. До цього підходу відносять алгоритми, засновані на аналізі змісту (опису) об'єкта голосування зі змістом профілю користувача (частковій обробці). Вони дають рекомендації на основі порівняння інформації про предмет рекомендації з уявленнями про те, що цікаво користувачам. Даний підхід в основному ґрунтується на кластеризації профілів користувачів та опису об'єктів з подальшим знаходженням відповідності та кількісної міри відповідності.

Оскільки має місце робота зі змістом (в основному в текстовому вигляді), то даний клас алгоритмів є спеціалізованим для конкретної предметної галузі. Зазвичай основна проблема побудови такої системи полягає в тому, що потрібне об'єктивне, точне і обов'язкове заповнення змісту профілів, що достатньо важко досягти у промисловій системі.

До того ж практика показала неефективність використання даного класу методів і віддає перевагу наступному класу.

Колаборативна фільтрація (collaborative filtering). Даний підхід ще називається загальною фільтрацією. Він є найпоширенішим у галузі систем рекомендацій і поєднує собою декілька моделей і багато алгоритмів.

Колаборативна фільтрація – підхід, де рекомендації надаються з урахуванням тільки попередньої історії оцінювання. Оскільки в даному класі моделей використовується лише інформація про попередню історію оцінювання і виключається будь-яка додаткова інформація про користувачів і об'єкти оцінювання, то одну модель (а також алгоритм) можна використовувати в різних предметних галузях без змін.

Найпоширеніша модель у даному підході і набір алгоритмів, що базуються на даній моделі є модель найближчих сусідів.

Метод k найближчих сусідів (k Nearest Neighbor Method). Нехай маємо множину користувачів U, множину об'єктів I і матрицю оцінок R. Позначимо терміном користувач вектор, розмірності n, усіх оцінок, що належать користувачу u_j, позначимо терміном об'єкт вектор, розмірністю m, усіх оцінок, що належать даному об'єкту i_j .

Уведемо поняття схожості між користувачами чи об'єктами. Будемо казати, що користувач u_j є схожим на користувача u_k, якщо вони здатні надавати одним і тим самим об'єктам схожі оцінки. Кількісну міру схожості визначає конкретна реалізація методу. Як правило, її вибирають із двох схожих між собою мір – косинуса між векторами чи коефіцієнта кореляції Пірсона.

Значення косинуса схожості між користувачами обчислюється

$$Sim_{u,v}=\frac{\sum_{i=1}^{n} r_{u,i} r_{v,i}}{\sqrt{\sum_{i=1}^{n} r_{u,i}^{\ 2} \sum_{i=1}^{n} r_{v,i}^{\ 2}}} \quad (7)$$

Між об'єктами

(7)

$$Sim_{i,j} = \frac{\sum_{u=1}^{m} r_{u,i} r_{u,j}}{\sqrt{\sum_{u=1}^{m} r_{u,i}^{\,2} \sum_{u=1}^{m} r_{u,j}^{\,2}}} \qquad (8)$$

Коефіцієнт кореляції між користувачами обчислюється

$$Sim_{u,v} = \frac{\sum_{i=1}^{n} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i=1}^{n} (r_{u,i} - \bar{r}_u)^2 \sum_{u=1}^{m} (r_{u,i} - \bar{r}_v)^2}} \qquad (9)$$

(9)

Між об'єктами

.

$$Sim_{i,j} = \frac{\sum_{u=1}^{m} (r_{u,i} - \bar{r}_i)(r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{u=1}^{m} (r_{u,i} - \bar{r}_i)^2 \sum_{u=1}^{m} (r_{u,j} - \bar{r}_j)^2}} \qquad (10)$$

Розглянемо два варіанти реалізації методу k найближчих сусідів.

Орієнтований на користувачів (user-oriented). У цьому підході алгоритм працює з користувачами системи (векторами оцінок користувачів). Ідея методу – знаючи вектори оцінок користувачів у системі встановити міру схожості між ними і намагатися представити невідомий рейтинг для даного користувача через лінійну комбінацію (зважену суму) k його сусідів.

Нехай маємо задачу системи рекомендацій у термінах (1)–(5). Наша задача – побудувати множину наближених оцінок $\bar{R}_N$. Для визначеності будемо розуміти обчислення на кожному кроці методу лише однієї наближеної оцінки $\bar{r}_{ui}$.

На першому кроці будуємо матрицю S, розміром (m×m), де m – кількість користувачів у системі. Кожен елемент матриці $s_{u,v}$ – кількісна міра схожості між користувачами u, v. Зазвичай коефіцієнт схожості визначається формулами (7)-(9). Оскільки користувачів у системі, як правило, велика кількість, то коефіцієнт схожості повинен обчислюватися швидко, бажано за лінійний час (лінійно від кількості об'єктів у системі).

На другому кроці будуємо список коефіцієнтів схожості для даного користувача системи u та сортуємо його у порядку спадання.

На третьому кроці обираємо k перших коефіцієнтів, що відповідають k найбільш схожим сусідам (користувачам системи). Коефіцієнт k визначається реалізацією алгоритму. Позначимо через множину N(k,u) – множину, що складається із k найближчих сусідів користувача u.

На четвертому кроці обчислюємо невідому шукану оцінку $\bar{r}_{ui}$ як зважену суму оцінок найближчих сусідів за формулою

$$\bar{r}_{ui} = b_{ui} + \frac{\sum_{v \in N(k,u)} s_{uv} (r_{vi} - b_{vi})}{\sum_{v \in N(k,u)} s_{uv}}, \qquad (11)$$

де b_{ui} – деяким чином нормоване значення оцінки користувача u чи об'єкта i у системі. Найбільш часто b_{ui} визначається як середня оцінка даного користувача чи об'єкта.

Даний алгоритм обчислює всі значення елементів з множини $\bar{R}_N$.

Як показано в [1; 3; 7] даний підхід показує гарні практичні результати в наступних випадках:

– кількість об'єктів значно перевищує кількість користувачів у системі;
– кожен з користувачів системи має достатньо багато оцінених об'єктів;
– нові об'єкти та користувачі у системі додаються не досить часто;
– система орієнтована не тільки на надання наближеної невідомої оцінки, але й важливе надання списку користувачів, найбільш схожих за смаком з даним.

Орієнтований на об'єкти (item-oriented). Даний підхід є схожим з попереднім (орієнтованим на користувачів), проте замість користувачів алгоритм працює з об'єктами системи [4].

На першому кроці, як і у попередньому підході, будуємо матрицю S, але вже розміром (n×n), де n – кількість об'єктів у системі. Кожен елемент матриці $S_{i,j}$ – кількісна міра схожості між об'єктами i, j. Для знаходження коефіцієнта схожості використовуються ті ж самі підходи, що і у попередньому методі за формулами (7), (9).

На другому кроці будуємо список коефіцієнтів схожості для даного об'єкта системи та сортуємо його у порядку спадання.

На третьому кроці обираємо k перших коефіцієнтів, що відповідають k найбільш схожим сусідам (об'єктам системи).

Позначимо через множину N(k,u) – множину, що складається із k найближчих сусідів об’єкта u.

На четвертому кроці обчислюємо невідому шукану оцінку $\bar{r}_{ui}$ як зважену суму оцінок найближчих сусідів за формулою

$$\bar{r}_{ui} = b_{ui} + \frac{\sum_{j \in N(k,u)} s_{ij}(r_{uj} - b_{uj})}{\sum_{j \in N(k,u)} s_{ij}}. \qquad (12)$$

Даний підхід має переваги у випадку, коли:

– кількість користувачів значно переважає кількість об’єктів системи;

– система орієнтована не тільки на надання наближеної невідомої оцінки, але й важливе надання списку об’єктів, що найбільш схожі між собою [4].

Оскільки обидві реалізації подібні між собою і результати модифікацій одного з підходів легко перенести на інший, то будемо надалі розглядати лише модифікації підходу, орієнтованого на об’єкти.

Розглянемо деякі з багатьох модифікації методів k найближчих сусідів, що мають переваги над класичною (вищеописаною) реалізацією.

Метод k найближчих сусідів з оцінками за замовчуванням. Оскільки, як правило, кількість користувачів і об’єктів у системі достатньо велика, то навіть при достатньо великій кількості оцінок для кожного користувача (об’єкта) її відношення до загальної кількості об’єктів (користувачів) зостається малим. Це призводить до того, що при обчисленні коефіцієнтів схожості та подальшому виборі найбільш схожих сусідів втрачається значна кількість корисної інформації.

Наприклад, об’єкт має лише кілька оцінок (кількох користувачів) і ці оцінки є схожими з оцінками іншого об’єкта, проте в іншого об’єкта оцінок значно більше й інші оцінки значно відрізняються. При цьому система буде вважати даний об’єкт дуже схожим з другим, хоча в даній ситуації він просто має завищені оцінки від кількох користувачів.

Другий приклад – при підрахунку зваженої суми, об’єкти, в яких немає оцінки від даного користувача відкидаються, хоча ці об’єкти є схожими з даним і попадають у список із k сусідів. У цьому випадку ми також втрачаємо значну кількість корисної інформації.

За даною модифікацією пропонується при обчисленні коефіцієнтів схожості і зваженої суми замість пустих рейтингів для деякого об’єкта брати деякі усереднені оцінки даного об’єкта й використовувати їх при обчисленнях. При цьому ми не втрачаємо корисну інформацію від схожих об’єктів і позбавляємося від вищезазначених проблем.

Метод k найближчих сусідів з узагальненими вагами. У [2; 3] наводиться значна модифікація методу k найближчих сусідів, орієнтованого на об’єкти, з метою усунення вищезазначених недоліків.

Для обчислення невідомої оцінки пропонується узагальнена формула зваженої суми

$$\bar{r}_{ui} = \sum_{j \in N(k,i)} w_{ij} r_{uj}. \qquad (13)$$

Для знаходження вагових коефіцієнтів w_{ij} виконуємо мінімізацію функції відхилу відомих оцінок від наближених за всіма оцінками даного користувача для всіх сусідів даного об’єкта

$$\min_{w} = \sum_{u \neq v} (r_{vi} - \sum_{j \in N(k,i)} w_{ij} r_{uj})^2. \qquad (14)$$

У загальному випадку автори пропонують вирішувати дану задачу через розв’язання СЛАР, що отримуємо, вирішуючи (14) методами класичної оптимізації. Після диференціювання, та переносу відомих значень у праву частину рівнянь отримуємо:

$$Aw = b;$$

$$A_{jk} = \sum_{v \neq u} r_{vj} r_{vk};$$

$$b_j = \sum_{v \neq u} r_{vj} r_{vi}.$$

Множина N(k,i) у (13), (14) – множина k найближчих сусідів об’єкта i, що визначається таким самим чином, як і в (12) (класичному підході, орієнтованому на об’єкти).

Авторами даної модифікації було помічено, що оскільки оцінок у системі порівняно мало з загальною кількістю всіх можливих, то добутки $r_{vj}r_{vk}$ і $r_{vj}r_{vi}$ з великою ймовірністю становляться нульовими, що призводить до втрати інформації про відношення між елементами пари об’єктів у подальших обчисленнях, навіть при ненульовому детермінанті матриці A.

У зв’язку з цим авторами запропоновано більш гнучкий узагальнений підхід для отримання коефіцієнтів w_{ij}:

$$\hat{A}w = \hat{b},$$

$$\bar{A}_{jk} = \frac{\sum_{v \in U(j,k)} r_{vj} r_{vk}}{|U(j,k)|},$$

$$\bar{b}_j = \frac{\sum_{v \in U(j,k)} r_{vj} r_{vi}}{|U(j,k)|},$$

$$\hat{A}_{jk} = \frac{|U(j,k)|\bar{A}_{jk} + \beta \cdot avg}{|U(j,k)| + \beta},$$

$$\hat{b}_j = \frac{|U(j,k)|\bar{b}_j + \beta \cdot avg}{|U(j,k)| + \beta},$$

де коефіцієнт β – емпірично підібраний, що несе у собі сенс регулювання тяжіння до елементу матриці A та загальної середньої оцінки. На практиці, при розв’язанні задачі розмірності поставленої компанією NetFlix пропонується вибір коефіцієнту β=500; avg – загальна середня оцінка у системі (за всіма користувачами та об’єктами); U(j,k) – множина користувачів, що дали оцінку об’єктам i, k.

У даному підході розв’язуємо узагальнену задачу. Матриця $\hat{A}$, взагалі кажучи, не має нульових елементів і є не виродженою, тому, в даному випадку, отримуємо значно інформативніший набір вагових коефіцієнтів w_{ij}.

Можлива також модифікація даного підходу з нормуванням по середній оцінці кожного з користувачів:

$$\min_w = \sum_{u \neq v} (r_{vi} - b_{vi} - \sum_{j \in N(i,u)} w_{ij}(r_{uj} - b_{uj}))^2,$$

$$\bar{r}_{ui} = b_{ui} + \sum_{j \in N(i,u)} w_{ij}(r_{uj} - b_{uj}).$$

.

Моделі виявлення скритих закономірностей (latent factor models) спрямовані на дослідженні скритих закономірностей і відношень між парами користувач, об’єкт. Нехай мємо X – множину тем, p(x|u) – неявний тематичний профіль користувача u, q(x|i) – неявний тематичний профіль об’єкта i. Ці моделі намагаються за оцінками користувача u знайти тематичний профіль цього користувача p(x|u) і запропонувати об’єкти тематичного профілю q(x|i).

Обмежена машина Больцмана (restricted Boltzmann machine) описана в [5] і застосовується як стохастичний алгоритм для обчислення наближеної оцінки.

З огляду на те, що машина Больцмана є нейронною сіткою, то їй притаманна не детермінованість процесу і довгий період навчання. До того ж, як показано в [5], її застосування можливе лише в обмеженому класі задач.

Беручи до уваги розмірність поставленої NetFlix задачі, застосування обмеженої машини Больцмана не є можливим.

Баєсовська модель відвідувань належить до моделей виявлення скритих закономірностей. Через термін відвідування позначається «присутність» оцінки у системі.

Дана модель за відомими ймовірностями, встановлених з відомих оцінок, максимізує ймовірність того, що дана оцінка тематичного профілю даного об’єкта (для якого виконується рекомендація) і даного користувача співпадуть

$$p(u,i) = \sum p(u)\,p(x \mid u)q(i \mid x) = \sum q(i)q(x \mid i)\,p(u \mid x);$$

$$q(i \mid x) = \frac{q(x \mid i)q(i)}{\sum_s q(x \mid s)q(s)};$$

$$q(u \mid x) = \frac{q(x \mid u)q(u)}{\sum_s q(x \mid v)q(v)}.$$

Надалі розв’язуємо задачу максимізації правдоподібності

$$\sum_{j=1}^{m} \ln p(u_j, i_j) \to \max_{p(x|u), q(x|i)}.$$

Дана модель є практично корисною для роботи на будь-яких об’ємах даних. Як показано в [1-4] вона є корисною для встановлення тих зв’язків у системі, що не можуть встановити методи найближчих сусідів, оскільки надає погляд на вихідні дані з більш високою абстракцією.

Наведемо список сильних сторін даної моделі:

- більш ефективно порівнюються користувачі і об’єкти системи;
- не потрібно великих структур даних тримати у пам’яті;
- можливо порівнювати користувача з об’єктом;
- тематичні профілі гарно інтерпретуються;
- вирішується проблема «холодного старту»;
- можливе «часткове навчання», коли тематичні профілі задаються лише в деяких об’єктів.

*Гібридні методи***.** Обидва з вищезазначених класів методів мають свої недоліки, що можуть бути усунуті взаємодоповненням обох класів алгоритмів. Тому був запропонований ряд моделей і алгоритмів з гібридним (комбінованим) підходом, що усувають одну, чи кілька вищезазначених проблем.

У теоретичній і практичній частині галузі систем рекомендацій існує багато проблем в аспекті якості оцінювання, розгортання системи, моделювання психології користувача і таке інше.

Проблема холодного старту. Як було зазначено в описі підходів, щодо побудови систем рекомендацій на базі колаборативної фільтрації, дані системи погано працюють на початку експлуатації, коли кожен користувач ще не оцінив достатню кількість об’єктів і об’єкти не мають достатньо великої кількості оцінок.

До цієї ж проблеми відносять проблему додавання нового користувача чи об’єкта у систему. У цьому випадку новий користувач чи об’єкт взагалі не має оцінок, з чого не можливо знайти схожий до нього (коефіцієнт схожості буде рівним нулю).

Як розв’язок проблеми холодного старту в [10] пропонується перехід від чистої системи рекомендацій на базі колаборативної фільтрації до гібридної. У цьому випадку, доки користувач чи об’єкт не наберуть достатньої кількості оцінок для них рекомендації будуть визначатися на базі аналізу змісту. В іншому – на базі колаборативної фільтрації. Також у [10] пропонуються інші варіанти розв’язання даної проблеми.

Проблема неправдивості оцінок. Це проблема навмисного завищення чи заниження конкретним користувачем своїх оцінок. Це призводить до того, що в об’єктивність і правильність наближеної оцінки $\bar{r}_{ui}$ вноситься додатковий шум.

Тут можемо виділити дві проблеми:

- користувач ненавмисне природно здатен ставити всім об’єктам більші чи менші оцінки ніж інші;
- користувач навмисне ставить більші чи менші оцінки конкретним об’єктам без закономірності з метою зламування системи.

Як розв’язок першої проблеми пропонується нормалізація оцінок. У найпростішому випадку – це віднімання середнього значення оцінок корис-тувача від значення оцінки, з якою проводяться обчислення, і проведення операцій вже не з оцінками, а відхилами від середньої для даного користувача чи об’єкта.

Достатньо гарного розв’язку другого варіанта проблеми не існує. Пропонуються варіанти знаходження деяким чином подібних користувачів чи об’єктів і їх нейтралізація.

Проблема дисперсії оцінок користувача. Як було помічено – користувачі дають невідповідні друг другу оцінки в різні моменти часу одним і тим ж об’єктам. Якщо розглядати оцінки користувача в різні моменти часу як деякі значення випадкової величини, то можна встановити дисперсію оцінок користувача. Дана проблема належить більш галузі психології і не має теоретичних і практичних пропозицій вирішення.

Також існують інші проблеми в галузі систем рекомендацій, що не розглядаються в даній роботі.

Оцінювання алгоритмів систем рекомендацій – це окрема, достатньо велика і різноманітна, тема. Це є наслідком декількох причин, що описані в [9]. По-перше, різні алгоритми можуть бути гарними чи поганими для різних, за об’ємом, даних. По-друге, багато алгоритмів колаборативної фільтрації були розроблені спеціально для масивів даних, де користувачів значно більше, ніж об’єктів, чи навпаки. Такі алгоритми будуть працювати погано на масивах даних з іншими показниками. По-третє, оцінювати алгоритми систем рекомендацій важко, бо можуть розрізнятися цілі оцінювання. Найперші роботи в області оцінювання концентрувалися на точності алгоритмів колаборативної фільтрації у передбаченні невідомих оцінок (обчисленні наближених оцінок). У більш пізніх роботах (Shardanand і Maes) було показано, що коли системи рекомендацій використовуються з метою допомоги користувачу в прийнятті рішень, то важніше виміряти наскільки часто система приводить користувачів до невірних рішень. Ряд досліджень проводився на тему, на скільки система рекомендацій охоплює увесь набір об’єктів (Mobasher). Оскільки, як вже зазначалося раніше, одна з позитивних сторін систем рекомендацій є неочевидність рекомендацій для користувача, то можна оцінювати ступінь неочевидності зроблених рекомендацій (McNee). Можна оцінювати систему рекомендацій за ступенем задоволення потреб користувача (в комерційних системах це може визначатися кількістю купленого товару). Останній з підходів полягає у поєднані вищезазначених оцінок з різними ваговими коефіцієнтами, при цьому складність полягає у правильному підборі вагових коефіцієнтів для даної предметної галузі.

Актуальним для даної роботи є оцінювання точності алгоритму. Найбільш поширенішим підходом до оцінювання загальної точності алгоритму є обчислення глобальної похибки алгоритму RMSE (root mean square error) чи MSA (mean square error) – середньоквадратичної похибки між відомою оцінкою і наближеною, обчисленою за допомогою алгоритмів системи рекомендацій, на деякому тестовому масиві даних

$$RMSE = \sqrt{\frac{1}{|V|}\sum_{(u,i)\in V}(\bar{r}_{ui} - r_{ui})^2},$$

де V – множина індексів відомих оцінок; $r_{ui} \in R_n$ – множина відомих оцінок (які поставив користувач); $\bar{r}_{ui} \in \bar{R}_n$ – множина наближених оцінок, обчислених за допомогою алгоритмів системи рекомендацій.

Вибір моделі для реалізації та дослідження. Для реалізації при розв'язанні поставленої задачі, було обрано наступні моделі з відповідними міркуваннями:

1) Класичний метод найближчих сусідів, орієнтований на об'єкти.

З умов поставленої задачі видно кілька фактів, при яких доцільно обирати даний метод. Об'єктів (фільмів) у системі NetFlix набагато менше за користувачів. Оскільки NetFlix надає достатньо великий масив робочих даних з реальної системи з достатньо гарною наповненістю, то немає проблеми холодного старту. Це найбільш швидка реалізація з методів колаборативної фільтрації.

2) Метод найближчих сусідів, орієнтований на об'єкти, з узагальненими вагами.

Даний метод поєднує у собі такі переваги як достатньо велика швидкість роботи порівняно з іншими алгоритмами колаборативної фільтрації, більш висока точність оцінювання, порівняно з попереднім обраним методом.

3) Авторська модифікація класичного методу найближчих сусідів, орієнтованого на об'єкти.

Метою створення даної модифікації та її вибору було збільшення точності, порівняно з класичним методом найближчих сусідів, орієнтованого на об'єкти.

Постановка задачі. Компанія NetFlix працює в галузі надання рекомендацій про перегляд фільмів на DVD щодо покращення результатів оцінювання.

Для розробників NetFlix надає тестовий масив даних зі своєї системи рекомендацій. У системі є 480189 користувачів та 17770 фільмів. У тестовому масиві даних більш ніж 100 000 000 оцінок, проставлених користувачами.

Разом з цим надано файл для дослідження роботи алгоритму системи рекомендацій. Він містить множину індексів підмножини відомих оцінок. Використання даного файлу дає можливість обчислити середньоквадратичну похибку алгоритму (різниця відомої у системі оцінки та оцінки обчисленої алгоритмом системи рекомендації). Опис структур даних, наданих NetFlix можна подивитися на сайті компанії.

Потрібно побудувати алгоритм для надання рекомендацій про перегляд фільму та підрахувати його похибку на тестовій множині.

Формально задача виглядає як (1)-(5) з урахуванням реальних обмежень.

Алгоритм розв'язку. Як було зазначено вище, для побудови своєї реалізації системи обрано два методи: класичний метод найближчих сусідів та його узагальнений варіант метод найближчих сусідів з узагальненими вагами. Враховуючи переваги і недоліки обох методів запропоновано модифікацію класичного методу найближчих сусідів – метод най-ближчих сусідів з інформативністю об'єктів. Мотивацією у побудові даної модифікації було збільшення точності класичного методу найближчих сусідів з прийнятним збільшенням обчислювальної складності.

Метод найближчих сусідів з інформативністю об'єктів. Позначимо терміном інформативності об'єкта – коефіцієнт, що показує відношення кількості оцінок об'єкта до загальної кількості можливих оцінок (це кількість користувачів).

Позначимо через r_i – множину оцінок об'єкта i (чи просто об'єкт i). Позначимо кількість оцінок об'єкта $c_i = |r_i|$. Нехай c_{min} – мінімальна кількість оцінок у об'єкта у системі, c_{min} – максимальна кількість оцінок у об'єкта у системі.

Визначимо коефіцієнт інформативності

$$\gamma_i = \alpha + \frac{c_i - c_{\min}}{c_{\max} - c_{\min}}(\beta - \alpha),$$

де α, β – емпірично підібрані коефіцієнти. Сенс даних коефіцієнтів у встановленні того факту, наскільки інформативність об'єкта є вагомою в обчисленнях потенційної оцінки.

При практичній реалізації даного алгоритму приймалися α=0,8, β=1,2 з міркувань балансу між покращенням точності роботи алгоритму та можливого збільшення похибки за рахунок шумів при обчисленнях при значному відхилі α, β від 1.

Очевидно, що чим більша інформативність об'єкта, тим більша об'єктивність його оцінок. Користувачі схильні надавати більш популярним об'єктам подібні оцінки і навпаки для більш специфічних – різні.

Модифікація розрахункової формули (12) з урахуванням інформативності користувачів виглядає наступним чином:

$$\bar{r}_{ui} = b_{ui} + \frac{\sum_{j\in N(k,u)} \gamma_i s_{ij}(r_{uj} - b_{uj})}{\sum_{j\in N(k,u)} \gamma_i s_{ij}}.$$

Розглянемо надалі етапи роботи побудованої системи оцінювання.

Маємо наступні вхідні дані:

Масив даних, наданих компанією NetFlix включає до себе:

- множину відомих оцінок R;
- метаінформацію про оцінки, користувачів, об'єкти;
- множину індексів невідомих оцінок (4).

Перший етап це препроцесінг, в якому база даних оцінок, матриця коефіцієнтів схожості, та масив середніх значень оцінок об'єктів обчислюються і зберігаються аналогічно попередньому методу. Крім того, обчислюємо множину значень c_i (для усіх об'єктів), c_{min}, c_{max}.

Далі для кожного індексу невідомої оцінки виконуємо:

1. Дістаємо із бази даних список із найближчих сусідів об'єкта і з відповідними коефіцієнтами схожості. Сортуємо список за не зростанням.
2. Будуємо новий список сусідів, де проставлена оцінка у користувача u. Для цього виконуємо наступні дії: йдучи по списку, починаючи з початку, перевіряємо чи поставлена в даного сусіда оцінка користувача u:
- якщо ні, то йдемо далі по списку;
- якщо так, то додаємо в новий список;
- якщо ми дійшли до кінця списку, то переходимо на 3.
3. Обчислюємо коефіцієнт інформативності γ_i для даного об'єкта.
4. Обчислюємо наближений рейтинг $\bar{r}_{ui}$.

Обчислювальний експеримент проходив у кілька етапів. Як зазначалося раніше перед стартом роботи конкретного методу необхідний препроцесінг – заповнення бази даних та розрахунок і збереження матриці коефіцієнтів схожості між об'єктами, яку використовують всі методи у процесі своєї роботи.

Після препроцесінгу було почергово досліджено кожен із алгоритмів. Підрахунок похибки RMSE відбувався 4 рази для кожного із алгоритмів, для різних значень параметра k = {10, 20, 30, 40}. Потім досліджувалася похибка на кожному об'єкті із підрахунком дисперсії оцінок об'єкта задля встановлення зв'язку між ними.

Далі результати були порівняні між собою для різних алгоритмів і різних значеннях параметра k.

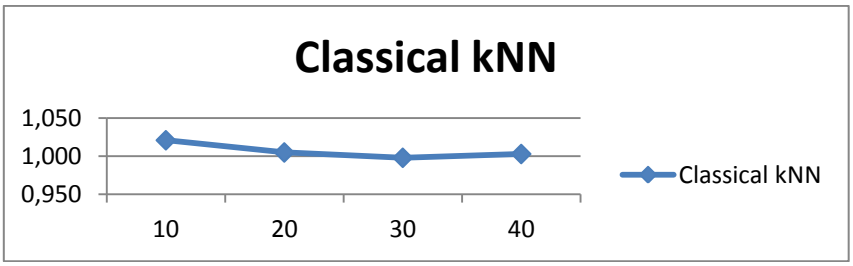
В якості тестової вибірки даних було використано наданий компанією NetFlix масив. Розрахунок RMSE відбувався на тестовій вибірці для елементів, які мають точні оцінки в базі.

Тестова множина даних, що надається компанією NetFlix у вигляді текстових файлів має об'єм трохи більше за 2 Гб. Для роботи з таким масивом даних перед початком препроцесінгу була створена спеціальна база даних, об'єм якої складав біля 4 Гб. Далі виконувалася операція включення головних ключів у базі даних, при цьому об'єм збільшився майже вдвічі і склав 7 Гб.

На наступному етапі відбувся підрахунок матриці коефіцієнтів схожості. При цьому база даних, в якій зберігалася дана матриця, мала розмір 1,5 Гб.

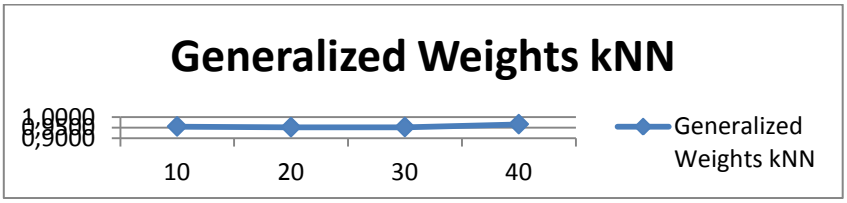
Першим проводилося дослідження залежності похибки від значення параметра k – кількості сусідів.

На тестовому масиві даних з приблизно 1 300 000 оцінок було проведено чотири обчислення похибки при різних значеннях k = {10, 20, 30, 40}. Результати обчислень приведені на діаграмі (1).



Діаграма 1. Залежність похибки класичного методу найближчих сусідів від кількості сусідів

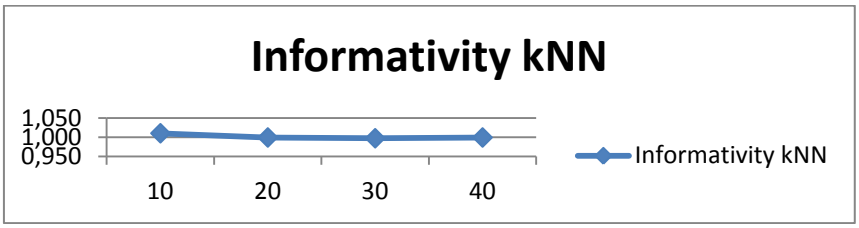
Другий з методів для дослідження – метод k найближчих сусідів з узагальненими вагами підтвердив факт, наведений авторами алгоритму про збільшення обчислювальної вартості алгоритму лінійно, порівняно з класичною реалізацією. При цьому метод показав більш точні значення загальної похибки алгоритму. Результати наведені на діаграмі (2).



Діаграма 2. Залежність похибки методу найближчих сусідів з узагальненими вагами від кількості сусідів

Третім з досліджуваних алгоритмів був запропонований алгоритм з інформативністю об'єктів. Передбачалося поліпшення якісті роботи алгоритму, порівняно з класичним аналогом.

У ході експерименту було підтверджено дану гіпотезу. Алгоритм мав майже такий самий час роботи, як і в класичного аналога, проте покращені результати роботи. Результати наведені на діаграмі (3).



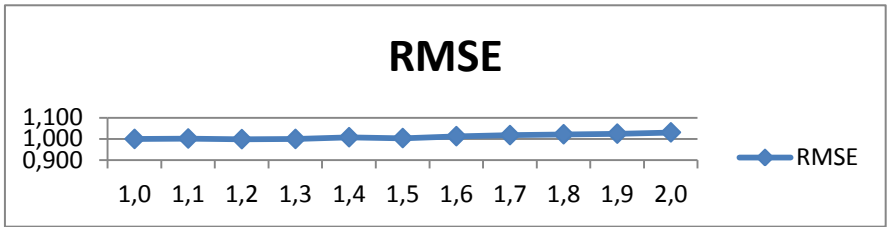
Діаграма 3. Залежність похибки методу найближчих сусідів з інформативністю об'єктів від кількості сусідів

Другим етапом експерименту було дослідження залежності похибки при оцінці одного об'єкта від дисперсії оцінок даного об'єкта. Висунуто гіпотезу про існування залежності, такої, що при збільшенні дисперсії буде збільшуватися похибка алгоритму. Це може бути наслідком того, що алгоритм не може достатньо точно знайти об'єктивне значення оцінки об'єкта, при великій дисперсії його оцінок.

При обчисленні використовувався наступний підхід: для кожного об'єкта обчислювалася дисперсія, а також обчислювалися похибки всіх оцінок, що знаходилися для даного об'єкта. Оскільки для більшості об'єктів їх дисперсія оцінок лежить у проміжку від одного до двох, то було обрано даний інтервал для дослідження. Він був розбитий на 10 підпроміжків, довжиною 0.1. Далі для всіх об'єктів, дисперсія яких попадає в конкретний проміжок, підраховувалася середня похибка.

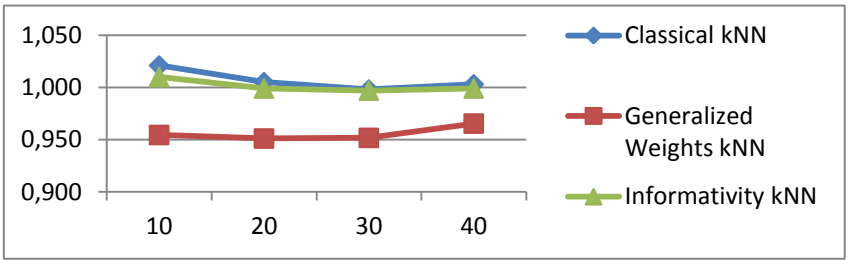
Результатом даного дослідження була середня похибка для кожного проміжку на шкалі дисперсії оцінок об'єктів.

З результатів (діаграма 4) видно, що похибка зростає зі збільшенням дисперсії об'єкта.



Діаграма 4. Залежність похибки методу від дисперсії об'єкта

При порівнянні методів можемо побачити, що метод з узагальненими вагами значно випереджає два інші аналоги. Порівнюючи метод з узагальненими вагами та метод з інформативністю користувачів можна зробити висновок, що при обчисленні узагальнених вагових коефіцієнтів у них враховується інформація про інформативність. Однаковою рисою всіх алгоритмів є найкраща робота при виборі кількості сусідів у діапазоні 25 – 30.



Діаграма 5. Порівняння похибок реалізованих алгоритмів в залежності від кількості сусідів

Таким чином, у результаті обчислювального експерименту, показано залежність похибки від кількості сусідів в усіх методах. Оптимальним значенням кількості сусідів є приблизна кількість 25-30. Показано, що метод з узагальненими вагами показує значно меншу загальну похибку, при лінійному збільшенні обчислювальної складності методу, порівняно з класичним. Чисельно підтверджено, що модифікований метод показує меншу загальну похибку, ніж класичний. Підтверджено гіпотезу про залежність між дисперсією оцінок об'єкта і якістю оцінювання на базі класичного методу.

Висновки. За виконаною роботою проаналізовано галузь систем рекомендацій і обрано два найбільш придатних до розв'язання поставленої задачі методи. Створено модифікацію класичного методу найближчих сусідів, орієнтованого на об'єкти. Досліджено і використано практично такі підходи програмування щодо роботи з великими об'ємами даних, як оптимізація БД і кешування. Створено програмний фреймворк для систем рекомендацій, що може використовуватися практично для створення системи рекомендації з необхідними характеристиками. Створено програмний продукт на базі розробленого фреймворку з реалізацією обраних алгоритмів. Проведено обчислювальний експеримент, у ході якого отримана оцінка роботи обраних алгоритмів. Модифікований метод показує більш гарну оцінку роботи, ніж класичний, проте гіршу за оцінку

роботу алгоритму з узагальненими вагами. Показано залежність між дисперсією оцінок об'єкта і якістю оцінювання.

Бібліографічні посилання

1. **Adomavicius G., Tuzhilin A.** Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions // Transactions on Knowledge and Data Engineering. – 2005. – Vol. 17, N 6. – P. 734–749.
2. **Bell R., Koren Y.** Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights // AT&T Labs Research. IEEE Conference on Data Mining (ICDM'07), – 2007. P. 42–49.
3. **Bell R., Koren Y., Volinsky C.** Modeling Relationships at Multiple Scales to Improve Accuracy of Large Recommender Systems // AT&T Labs Research. Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'07), – 2007. P. 33–41.
4. **Linden G., Smith B. York J.** Amazon.com Recommendations: Item-to-Item Collaborative Filtering // IEEE Internet Computing. – January/February 2003. – Vol. 7, N 1. – P. 76–80.
5. **Salakhutdinov R., Mnih A., Hinton G.** Restricted Boltzmann Machines for Collaborative Filtering // Proc.24th Annual International Conference on Machine Learning. – 2007.
6. **Sarwar B., Karypis G., Konstan J., Riedl J.** Item-based Collaborative Filtering Recommendation Algorithms // Proc. 10th International Conferenceonthe World Wide Web, P.285–295, 2001.
7. **Wang J., Vriesand A., Reinders M.** Unifying User-based and Item-based Collaborative Filtering Approaches by Similarity Fusion // Proc.29th ACM SIGIR Conference on Information Retrieval. – 2006. P.501–508.
8. **Melville P., Mooney R., Nagarajan R.** Content-boosted collaborative filtering for improved recommendations // 18th National Conference on Artificial Intelligence (AAAI). – 2002. – P. 187–192.
9. **Herlocker J., Konstan J., Terveen L., Riedl J.** Evaluating collaborative filtering recommender systems // ACM Translations on Information Systems. Vol. 22(1). – 2004. – P. 296–302.
10. **Schein A., Popescul A., Ungar L., Pennock D.** Methods and Metrics for Cold-Start Recommendations // 25th Annual. International ACM SIGIR Conference on Research and Development in Information Retrieval. – 2002. – P. 253–260.

Надійшла до редколегії 12.12.09