

Є.В. Мелешко, С.Г. Семенов, В.Д. Хох

Центральноукраїнський національний технічний університет, Кропивницький

ДОСЛІДЖЕННЯ МЕТОДІВ ПОБУДОВИ РЕКОМЕНДАЦІЙНИХ СИСТЕМ В МЕРЕЖІ ІНТЕРНЕТ

У роботі розглянуті основні види рекомендаційних систем у мережі Інтернет, засновані на методах контентної та колаборативної фільтрації. Розглянуто способи збору даних про користувачів з веб-ресурсів, необхідні для формування рекомендацій. Досліджено методи побудови класифікаторів для контентної фільтрації. Також досліджено способи обчислення коефіцієнту подібності користувачів або об'єктів у колаборативній фільтрації.

Ключові слова: рекомендаційні системи, інтелектуальні системи, контентна фільтрація, колаборативна фільтрація, метрики подібності, алгоритми класифікації.

Вступ

На сьогоднішній день у сфері інформаційних технологій для просування товарів та послуг все частіше використовують так звані рекомендаційні системи – інструменти автоматичної генерації рекомендацій послуг та товарів на основі вивчення персональних потреб користувачів веб-сайтів. Рекомендаційні системи застосовують, наприклад, такі популярні сайти як YouTube, Amazon, Facebook тощо.

Основною сферою використання рекомендаційних систем на сьогоднішній день є маркетинг, також вони використовуються для ранжування результатів пошукових запитів у пошукових системах, хоча їх застосування може бути значно ширшим. Так як дані системи дозволяють виявляти групи схожих користувачів або об'єктів, їх можна застосувати як альтернативні методи пошуку інформації у мережі Інтернет, тому що вони дозволяють виявити об'єкти, які не можуть бути знайдені традиційними пошуковими алгоритмами.

Також рекомендаційні системи можна використовувати для фільтрації шкідливого контенту, наприклад, спаму.

Рекомендаційні системи – програмне забезпечення, що використовується для прогнозування того, які об'єкти (товари, веб-сайти, фільми, новини тощо) будуть цікаві користувачу, на основі зібраної про нього інформації.

Існують три основні типи рекомендаційних систем в мережі Інтернет [1]:

1. Засновані на контентній фільтрації (content-based filtering).
2. Засновані на колаборативній фільтрації (collaborative filtering).
3. Гібридні методи.

Контентна фільтрація формує рекомендації на основі поведінки користувача, наприклад, на основі того, які веб-сторінки він відвідував раніше, які оцінки поставив товарам тощо.

Колаборативна фільтрація формує рекомендації для певного користувача на основі вподобань групи користувачів, схожих на нього.

Ідея колаборативної фільтрації заснована на тому, що ті користувачі, які однаково оцінювали будь-які предмети в минулому, схильні давати схожі оцінки інших предметів і в майбутньому [2, 5].

Гібридні методи поєднують обидва підходи. Поєднання контентної та колаборативної фільтрації дають можливість отримати більш точні рекомендації. В гібридних методах спочатку здійснюється контентна фільтрація, а потім її результати уточнюються за допомогою колаборативної фільтрації [3].

До найпоширеніших проблем рекомендаційних систем можна віднести проблему *холодного старту* та *бульбашки фільтрів*.

Проблема холодного старту виникає при додаванні нового об'єкту чи користувача до системи. До появи відповідних новому об'єкту чи користувачу оцінок, відсутня інформація для формування рекомендацій.

Проблема бульбашки фільтрів виникає у рекомендаційних та пошукових системах, коли алгоритм видачі інформації веб-сайту вибірково підбирає дані, враховуючи яку інформацію користувач хотів бачити, базуючись на інформації про його вподобання і, в результаті, користувачі відділяються від інформації, яка їх не цікавить або їм не подобається, фактично ізолюючи їх у власних «бульбашках».

Одним зі шляхів підвищення точності роботи рекомендаційних систем та вирішення проблем холодного старту і бульбашки фільтрів є використання контексту, в якому відбувається фіксація вподобань користувача, та контексту, в якому відбувається формування рекомендацій [3, 4].

У якості контексту можуть бути використані дата та час формування рекомендацій, пристрій, з якого користувач відвідав веб-сайт, його демографічні дані тощо.

1. Збір даних про користувачів

Інформацію про вподобання можна збирати по-різному. Даними можуть виступати, наприклад, придбані товари, а вподобаннями – поставлені їм оцінки (зірочки, лайки/дизлайки тощо). Також на веб-сайті можна проводити анкетування користувачів.

Найбільш повна інформація для формування вподобань користувачів знаходиться у адміністраторів сайтів. В той же час може виникати необхідність проаналізувати вподобання користувачів тих чи інших інтернет ресурсів, доступу до управління якими немає. У такому випадку існує два шляхи – або користуватися API-функціями сайту [5] (якщо вони є), або проектувати власний веб-кроулер для збору даних, що буде аналізувати код веб-сторінок [6].

Хоча зазвичай API-функції легкі у використанні, не всі сайти надають можливість зібрати необхідну інформацію завдяки API-функціям – такі функції можуть бути повністю відсутніми, або мати обрізаний функціонал, їх використання може бути пов'язаним з рядом обмежень та умов. Також слід враховувати, що час від часу інтерфейс API-функцій змінюється, можуть виходити їх нові версії, що вимагає внесення змін у програмне забезпечення для збору даних.

Нові версії API-функцій можуть містити менше можливостей, так, наприклад, Facebook, починаючи з версії 2.0 Graph API, не дозволяє збирати список друзів будь-якого користувача (можна отримати список друзів тільки користувача даного ресурсу, який встановив ваш додаток) [7].

При збиранні даних за допомогою власного веб-кроулера обмежень значно менше – можна зібрати будь-яку відкриту у вільний доступ інформацію з сайту. Але написання власного веб-кроулера – значно складніша задача, необхідно розібратися в html-верстці чужого сайту, власноруч розробити функції для збору даних з html-коду сайту, вносити зміни або повністю переписувати веб-кроулер кожного разу, коли на досліджуємому сайті змінюється html-верстка.

Для аналізу html-коду веб-сторінок можна скористатися відкритими бібліотеками для парсингу веб-сторінок.

Одним з перспективних способів парсингу веб-сайтів є застосування бібліотеки для автоматизації дій веб-браузера та його тестування Selenium (для мов Java, C#, Ruby, Python, Javascript) [7]. Але основною проблемою буде необхідність розібратися в чужій html-верстці і розробити способи пошуку у ній потрібної інформації.

Пошук необхідних елементів на веб-сторінці можна здійснювати наступними способами:

- пошук за іменем html-тегу;
- пошук за XPath-виразом;
- пошук за cssSelector - даний тип локаторів заснований на описах таблиць стилів (CSS).
- пошук за атрибутом id html-тегу;
- пошук за атрибутом class html-тегу;
- пошук за посиланням з зазначеним текстом;
- тощо.

Наприклад, якщо необхідно отримати список товарів на певній веб-сторінці інтернет-магазину, треба дослідити особливості верстки даного сайту, назви товарів можуть, наприклад, міститися в посиланнях з певними атрибутами певних тегів. Після виявлення закономірностей у верстці сайту, задачу одержання списку товарів можна вирішити за допомогою пошуку по коду сторінки з використанням регулярних виразів та бібліотек для парсингу html-коду.

Методи парсингу веб-сторінок дозволяють зібрати більше інформації та дають більше свободи дій розробнику, але вони складніші в реалізації та більш повільні в роботі порівняно з використанням API-функцій.

2. Контентна фільтрація

Рекомендаційні системи з фільтрацією на основі змісту беруть до уваги схожість об'єктів з інформацією відомою про користувача.

Інформація про користувача може бути отримана з його профілю та/або зібрана з його дій на веб-сайті – написаних відгуків та коментарів, придбаних товарів, переглянутих веб-сторінок тощо.

Для створення рекомендацій такі системи аналізують інформацію про користувача, формують ключові слова про його інтереси та вподобання. Рекомендуються будуть об'єкти з бази даних, вибрані на основі визначених ключових слів.

Основні методи класифікації, на основі яких може бути здійснена контентна фільтрація [1, 5]:

1. Класифікатори на основі Байєсівських мереж.
2. Класифікатори на основі нейронних мереж.
3. Класифікатори на основі дерев рішень.
4. Класифікатори на основі алгоритмів кластеризації.

Класифікатори на основі Байєсівських мереж. Одним з найвідоміших класифікаторів є найвигідніший байєсівський класифікатор. В його основі лежить ймовірнісна модель теореми Байєса. Для роботи алгоритмів з використанням даного класифікатора необхідно створити модель Байєса для кожного користувача, який оцінював будь-які об'єкти, на основі ознак цих об'єктів (для фільмів це можуть бути актори або жанри, для новин – ключові слова тощо). Для знаходження найбільш ймовірної категорії необхідно обчислити умовні ймовірності приналежно-

сті будь-якого об'єкту до кожної категорії і вибрати ту, яка має найбільшу ймовірність (1):

$$\Pr(C | O) = \Pr(O | C) \cdot \Pr(C) / \Pr(O); \quad (1)$$

$$\Pr(O | C) = \Pr(O | C) = \prod_{i=1}^k \Pr(W_i | C) \quad (2)$$

де O – об'єкт (товар, послуга тощо); C – категорія; W – слово (ознака); n – кількість слів (ознак); $\Pr(C)$ – повна ймовірність того, що випадково обраний об'єкт потрапляє у категорію C ; $\Pr(O)$ – повна ймовірність появи об'єкту O ; $\Pr(W_i | C)$ – умовна ймовірність того, що при наявності слова W_i в описі об'єкту O , об'єкт O відноситься до категорії C .

Для кожної категорії можна задати порогові значення. Тоді щоб новий об'єкт був віднесений до деякої категорії, ймовірність його віднесення до цієї категорії повинна бути більша ймовірності його потрапляння в будь-яку іншу категорію на величину порогового значення.

Перевагою байєсівських класифікаторів є можливість навчання, простота алгоритму навчання, можливість перегляду даних про важливість ознак, одержану в процесі навчання. Основний недолік – неможливість враховувати залежність результату від поєднань ознак.

Класифікатори на основі нейронних мереж.

Штучні нейронні мережі можуть мати різну архітектуру та різні алгоритми навчання, але всі вони складаються з нейронів, загальна структурна схема яких наведена на рис. 1.

На рис. 1 позначено:

x_1 - x_n – входи нейрона (синапси);

w_1 - w_n – вагові коефіцієнти входів;

S – зважена сума входів нейрона;

$F(S)$ – функція активації нейрона;

T – порогове значення (значення, після якого нейрон переходить у стан збудження), є не у всіх типів штучних нейронів;

Y – вихід нейрона (аксон).

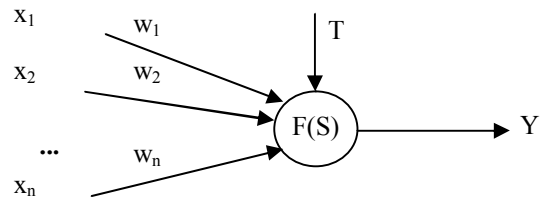


Рис. 1. Загальна структурна схема нейрона у штучній нейронній мережі [8, 9]

Зважена сума S обчислюється за наступною формулою:

$$S = x_1 \cdot w_1 + x_2 \cdot w_2 + \dots + x_n \cdot w_n \quad (3)$$

Функція активації $F(S)$ – визначає залежність сигналу на виході нейрона від зваженої суми сигналів на його входах. В якості функції активації можуть використовуватися: лінійна функція, порогова функція, сигмоїдальна функція тощо.

В найпростіших випадках для контентної фільтрації можна застосовувати багатoshаровий перцептрон, якщо є дані для попереднього навчання і мережу Кохонена, якщо доведеться вчити мережу в процесі використання. Розглянемо загальну структуру перцептронну для використання у контентній фільтрації (рис. 2). Входами нейромережі будуть коди ознак об'єктів, а виходами – коди категорій об'єктів. Нейромережа буде містити декілька прихованих прошарків, кількість прошарків та нейронів у них зазвичай визначаються експериментальним шляхом.

Загальною є рекомендація, що кількість нейронів у прихованих прошарках має бути більшою за кількість нейронів у вхідному прошарку.

Зазвичай при побудові нейронної мережі всі вузли створюються заздалегідь. У контентній фільтрації для рекомендаційних систем в [5] пропонується новий прихований вузол створювати тоді, коли зустрічається нова комбінація ознак та створювати для нього зв'язки з вагами за замовчуванням.

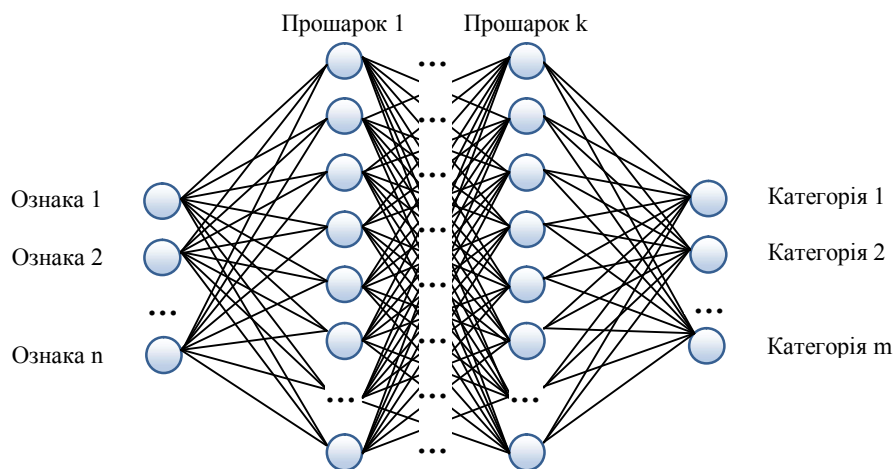


Рис. 2. Загальна схема нейронної мережі для класифікації об'єктів у рекомендаційній мережі

Нейронні мережі здатні справлятися зі складними нелінійними функціями та знаходити залежності між різними вхідними даними. Нейронні мережі допускають адаптивне навчання в процесі використання.

Основні недоліки – вони працюють як чорний ящик, відсутні тверді правила по вибору структури та розміру мережі, швидкості навчання.

Класифікатори на основі дерев рішень. Дерево рішень – це модель, яка являє собою сукупність правил для прийняття рішення, в даному разі – рішення, до якої категорії віднести об'єкт. Графічно її можна уявити у вигляді дерева, де вузли – умови переходу, а листки – назви категорій.

Якщо для даного об'єкту умова у вузлі істина то здійснюється перехід по лівому ребру, якщо ж ні, то по правому.

Залежно від рішення, прийнятого у вузлах, об'єкт відноситься до певної категорії.

Метод дерев рішень реалізує *принцип рекурсивного поділу*. Ця стратегія також називається «розділяй і володарюй». У вузлах, починаючи з кореневого, вибирається ознака, значення якої використовується для розбиття всіх даних на 2 класи. Процес триває до тих пір, поки не виконається критерій зупинки. Це можливо в таких ситуаціях:

- Всі (або майже всі) дані даного вузла належать одному і тому ж класу;
- Не залишилося ознак, за якими можна побудувати нове розбиття;
- Дерево перевищило заздалегідь заданий «ліміт зростання» (якщо ліміт було встановлено).

Існують різні чисельні алгоритми побудови дерев рішень. Одним з найбільш відомих є алгоритм під назвою C5.0 [11], розроблений програмістом Джоном Квінланом.

Фактично алгоритм C5.0 є стандартом процедури побудови дерев рішень. Ця програма реалізується на комерційній основі, але версія, вбудована в пакет R доступна безкоштовно.

Дерева рішень корисні не тільки для класифікації, а також і для інтерпретації результатів. На відміну від баєсівського класифікатора легко визначають взаємозалежні ознаки. Але дерева рішень не підтримують адаптивне навчання в процесі використання.

Класифікатори на основі алгоритмів кластеризації. Кластеризація (кластерний аналіз) – це задача розбиття множини об'єктів на групи, які називаються кластерами.

Методи кластерного аналізу діляться на: ієрархічні та ітеративні.

Ієрархічні в свою чергу поділяються на агломеративні і дивізімні.

Ієрархічні агломеративні методи послідовно об'єднують окремі об'єкти в кластери.

Ієрархічні дивізімні методи кластеризації полягають, навпаки, у виділенні в окремий кластер об'єктів, що мають найменші показники схожості, при тому, що спочатку вся мережа розглядається як окремий кластер.

Перевагами ієрархічних методів кластеризації є їх наочність і можливість отримати детальне уявлення про структуру даних. Ієрархічні алгоритми пов'язані з побудовою дендрограм, які описують близькість окремих точок і кластерів один до одного та представляють в графічному вигляді послідовність об'єднання (розділення) кластерів. Недоліки ієрархічних методів кластеризації – обмеження об'єму набору даних; вибір міри близькості; негнучкість отриманих класифікацій. Ієрархічні методи використовуються при невеликих об'ємах наборів даних. При великій кількості даних вони не придатні. У таких випадках використовують ітеративні методи.

Ітеративні методи – методи кластеризації, в яких кластери формуються виходячи з умов розбиття, які можуть бути змінені користувачем для досягнення бажаної цілі. Ці методи можуть призвести до утворення перетину кластерів, коли один об'єкт може одночасно належати декільком кластерам. В процесі поділу нові кластери формуються до тих пір, поки не буде виконано правило зупинки. Існує два підходи до розділення набору даних на певну кількість окремих кластерів.

Перший полягає у визначенні меж кластерів як найбільш щільних ділянок в багатовимірному просторі початкових даних, тобто визначення кластера там, де є велике "згущення" точок.

Другий підхід полягає в мінімізації міри відмінності об'єктів.

Найбільш поширений метод ітеративної кластеризації – *метод k-середніх*.

Ітеративні методи можна використовувати для великих об'ємів даних, також вони виявляють вищу стійкість по відношенню до шумів і викидів, некоректного вибору метрики, включення незначущих змінних в набір, що беруть участь в кластеризації. Недоліком ітеративних методів є те, що треба заздалегідь визначити кількість кластерів.

Якщо невідоме число кластерів, треба використовувати ієрархічні алгоритми, або декілька разів використати ітеративні методи з різною кількістю кластерів.

Перевагою контентної фільтрації є те, що для початку роботи рекомендаційної системи не потрібно великої кількості зареєстрованих користувачів. Головним недоліком даного підходу є неможливість системи рекомендувати нові об'єкти, які не прив'язані до інтересів користувачів.

При цьому виникає проблема підтримки зв'язку з користувачем.

3. Колаборативна фільтрація

Першим кроком алгоритмів заснованих на колаборативній фільтрації є обчислення коефіцієнтів подоби користувачів або об'єктів, обчислення здійснюються на основі оцінок, які користувачі виставляють об'єктам.

Зібрані дані про оцінки користувачів записуються у матрицю (рис. 3).

	Об'єкт 1	Об'єкт 2	...	Об'єкт n
Користувач 1	5	3	...	3
Користувач 2	4	-	...	2
...
Користувач m	-	4	...	4

Рис. 3. Матриця користувачів-об'єктів

В матриці користувачів-об'єктів значення є оцінками конкретного користувача для конкретного об'єкту. Ті осередки, в яких немає значень, є невідомими, тобто для даного об'єкту користувач не виставив оцінку.

Для обчислення коефіцієнтів подоби можуть використовуватися такі алгоритми:

1. Евклідова відстань.
2. Манхетенська відстань.
3. Коефіцієнт кореляції Пірсона.

Евклідова відстань. Це метрика, що визначає відстань між двома точками в евклідовому просторі, яка обчислюється за теоремою Піфагора (4):

$$k(u_1, u_2) = \sqrt{\sum_{k=1}^n (|u_{1k} - u_{2k}|)^2}, \quad (4)$$

де u_1, u_2 - користувачі та їх оцінки;
 n - кількість предметів в матриці.

Алгоритми, що застосовують даний коефіцієнт подоби дають недостатню точність при погано заповненій матриці, але прості в реалізації

Манхетенська відстань. Манхетенська відстань або відстань міських кварталів є одним з базових методів обчислення відстані між двома точками, розраховується так:

$$k(u_1, u_2) = |u_{1k} - u_{2k}|, \quad (5)$$

де u_1, u_2 - користувачі та їх оцінки;
 n - кількість предметів в матриці;

Даний спосіб має недолік в точності при малому заповненні матриці, але також простий в реалізації та має високу швидкість виконання.

Кореляція Пірсона. Даний коефіцієнт подоби дає більшу точність, ніж попередні. Показує наскільки добре два набори даних лягають на пряму. Визначається за формулою (6), де u_1, u_2 - користувачі та їх оцінки; n - кількість предметів в матриці; \bar{u}_1 і \bar{u}_2 - середня оцінка користувачів:

$$k(u_1, u_2) = \frac{\sum_{i=0}^n (u_{1i} - \bar{u}_1)(u_{2i} - \bar{u}_2)}{\sqrt{\sum_{i=0}^n (u_{1i} - \bar{u}_1)^2} \cdot \sqrt{\sum_{i=0}^n (u_{2i} - \bar{u}_2)^2}}, \quad (6)$$

Значення $k(u_1, u_2)$ може бути від -1 до 1, де -1 відповідає абсолютній розбіжності користувачів, а 1 - абсолютній подібності.

Кореляція Пірсона є досить популярним алгоритмом в сфері колаборативної фільтрації.

Види колаборативної фільтрації [1, 5]:

1. Фільтрація, заснована на схожості користувачів.

2. Фільтрація, заснована на схожості об'єктів.

Фільтрація, заснована на схожості користувачів. Після того, як для рекомендаційної системи зібрані дані та побудована матриця користувачів-об'єктів (рис. 3), для формування рекомендацій певному користувачу необхідно здійснити наступну послідовність дій:

1. Обчислити множину коефіцієнтів подоби даного користувача з усіма іншими користувачами (7):

$$K_t = \{k_{t1}, k_{t2}, \dots, k_{tj}\}, \quad (7)$$

де k_{ij} - коефіцієнт подоби між i -тим та j -тим користувачами.

2. Ранжувати об'єкти, обчисливши зважену суму оцінок користувачів (8), щоб думка користувача з подібними вподобаннями вносила більший вклад у формування рекомендацій:

$$S_q = \{e_{q1} \cdot k_{t1}, e_{q2} \cdot k_{t2}, \dots, e_{qj} \cdot k_{tj}\}. \quad (8)$$

де e_{qj} - оцінка q -го об'єкту j -м користувачем.

3. Розділити кожну зважену суму на суму коефіцієнту подоби всіх користувачів, що ставили оцінки відповідному об'єкту (6), щоб об'єкти, що отримали більше оцінок не одержали перевагу.

$$P_q = e_{q1} / k_{t1}. \quad (9)$$

Вираз (9) дає прогноз оцінки i -го користувача q -му об'єкту.

4. Відсортувати множину $\{P_1, P_2, \dots, P_q\}$, одержану після третього кроку та рекомендувати користувачу перші n об'єктів у списку.

Алгоритми, засновані на схожості користувачів інтуїтивно зрозумілі і прості в реалізації. Для великих об'ємів даних (коли багато користувачів та оцінок) ці алгоритми не підходять, так як вимагають багато часу на виконання розглянутих дій.

Фільтрація, заснована на схожості об'єктів. Основна ідея даних методів полягає у тому, щоб для кожного об'єкту заздалегідь визначити множину схожих на нього об'єктів. Тоді для формування рекомендацій певному користувачу достатньо буде знайти ті об'єкти, яким він поставив найбільші оці-

ники, та створити зважений список п об'єктів, максимально схожих на них. Результати порівняння об'єктів змінюються не так часто, як результати порівняння користувачів. Тож на першому кроці необхідно дослідити всі наявні дані, а подальші перерахунки можна робити рідко, вибираючи моменти часу, коли навантаження на веб-сайт мінімальне.

Методи фільтрації, заснована на схожості об'єктів працюють швидше, ніж методи засновані на схожості користувачів, так як багато обчислень можна здійснити заздалегідь. Тож їх можна з успіхом використовувати на великих об'ємах даних.

Висновки

У статті розглянуто методи побудови рекомендаційних систем у мережі Інтернет для формування рекомендацій користувачам веб-ресурсів з метою просування товарів та послуг. Було досліджено основні типи рекомендаційних мереж – з контентною та колаборативною фільтрацією.

Розглянуто основні методи побудови класифікаторів для контентної фільтрації та основні способи обчислення коефіцієнту подібності користувачів/об'єктів у колаборативній фільтрації.

Досліджено способи збору з веб-сайтів інформації про користувачів для вхідних даних у рекомендаційних системах, проведено основні принципи парсингу веб-сторінок.

На основі проведеного дослідження можна зробити висновок, що перспективними є гібридні рекомендаційні системи, з використанням алгоритмів машинного навчання, а також врахуванням контексту, в якому відбувається фіксація вподобань користувача та формування рекомендацій.

Список літератури

1. Jones M. *Recommender systems, Part 1. Introduction to approaches and algorithms. Learn about the concepts that*

underlie web recommendation engines / M. Jones – 2013. [Електронний ресурс] – Режим доступу: https://www.ibm.com/developerworks/opensource/library/os-recommender1/index.html?s_tact=105agx99&s_cmp=cp

2. Xiaoyuan Su and Taghi M. Khoshgoftaar *A Survey of Collaborative Filtering Techniques A Survey of Collaborative Filtering Techniques // Hindawi Publishing Corporation, Advances in Artificial Intelligence archive, USA : журнал. – 2009. — P. 1 - 19.*

3. Меньшикова Н.В. *Обзор рекомендательных систем и возможностей учета контекста при формировании индивидуальных рекомендаций / Н. В. Меньшикова, И.В. Портнов, И.Е. Николаев. // ACADEMY. – 2016. – №6. – с. 20–22.*

4. Пономарев А. В. *Обзор методов учета контекста в системах коллаборативной фильтрации // Труды СПИИРАН, 2013. № 7 (30), С. 169-188.*

5. Сегаран Т. *Программируем коллективный разум. – Пер. с англ. – СПб: Символ-Плюс, 2013. – 368 с.*

6. Охотный С.М. *Збирання даних про користувачів віртуальної соціальної мережі за допомогою веб-кроулера / С.М. Охотний, С.В. Мелешко // Збірник тез II Міжнародної науково-практичної конференції «Інформаційна безпека та комп'ютерні технології». м. Кропивницький. 20-22 квітня 2017 р. – Кропивницький: ЦНТУ. – 2017. – С. 157-159.*

7. *API Graph [Електронний ресурс] – Режим доступу: <https://developers.facebook.com/docs/graph-api/>*

8. *What is Selenium? [Електронний ресурс] – Режим доступу: <http://docs.seleniumhq.org/>*

9. Глибовець М.М., Олецький О.В. *Штучний інтелект: Підручник. – К.: Вид. дім "КМ Академія", 2002. – 366 с.*

10. Болотова Л. С. *Системы искусственного интеллекта: модели и технологии, основанные на знаниях / Л. С. Болотова – Москва: «Финансы и Статистика», 2012. – 663 с.*

11. *C5.0: An Informal Tutorial [Електронний ресурс] – Режим доступу: <https://www.rulequest.com/see5-unix.html>*

Надійшла до редколегії 30.11.2017

Рецензент: д-р техн. наук, проф. О.О. Можасв, Національний технічний університет «Харківський політехнічний інститут», Харків.

ИССЛЕДОВАНИЕ МЕТОДОВ ПОСТРОЕНИЯ РЕКОМЕНДАТЕЛЬНЫХ СИСТЕМ В СЕТИ ИНТЕРНЕТ

Е.В. Мелешко, С.Г. Семенов, В.Д. Хох

В работе рассмотрены основные виды рекомендательных систем в сети Интернет, основанные на методах контентной и коллаборативной фильтрации. Рассмотрены способы сбора данных о пользователях с веб-ресурсов, необходимые для формирования рекомендаций. Исследованы методы построения классификаторов для контентной фильтрации. Также исследованы способы вычисления коэффициента сходства пользователей или объектов в коллаборативной фильтрации.

Ключевые слова: рекомендательные системы, интеллектуальные системы, контентная фильтрация, коллаборативная фильтрация, метрики сходства, алгоритмы классификации.

RESEARCH OF METHODS OF BUILDING ADVISORY SYSTEMS ON THE INTERNET

E.V. Meleshko, S.G. Semenov, V.D. Khokh

The paper considers the main types of recommender systems on the Internet, based on the methods of , content-based and collaborative filtering. The article examines ways to collect user data from web-resources that is needed to formulate recommendations. Methods of constructing classifiers for content-based filtering were investigated. Methods of calculating the similarity coefficient of users or objects in collaborative filtering are also researched.

Ключевые слова: recommender systems, intellectual systems, content-based filtering, collaborative filtering, similarity metrics, classification algorithms.