

АЛГОРИТМИ І МЕТОДИ НАДАННЯ РЕКОМЕНДАЦІЙ ТА ЇХ ОПТИМІЗАЦІЯ

Розглянуто системи надання рекомендацій, їх застосування та роль у сучасному Інтернеті. Формалізована задача, яка має вирішуватись такими системами, та зроблено огляд методів систем сумісного фільтрування. Запропоновано удосконалення алгоритму надання рекомендацій та наведені практичні результати порівняння роботи алгоритмів.

Вступ

На сьогоднішній день в Інтернеті існує величезний обсяг інформації, який зростає швидкими темпами. Потреби користувачів у споживанні конкретної інформації та їх неможливість обробляти всю інформацію привели до стрімкого розвитку технологій пошуку інформації. Пошукові системи ефективно виконують задачі пошуку та роблять великі обсяги інформації доступними для широкого кола користувачів. Проте дуже часто пошукові системи повертають значно більше інформації, ніж користувач може обробити.

З ростом обсягів даних, що зберігаються у мережі і пропонуються користувачу, зростає актуальність проблеми випередження запиту користувача шляхом пропонування йому потенційно цікавої інформації. Цю проблему розв'язують системи надання рекомендацій. Основна відмінність алгоритмів систем надання рекомендацій від алгоритмів пошуку даних полягає у пропонуванні відповіді без явного запиту з боку користувача водночас, як пошукові алгоритми дають відповідь на конкретний запит користувача.

1. Розгляд систем надання рекомендацій

Згідно опублікованої в Інтернеті статті [1], провідні компанії – розробники пошукових систем (Yahoo!, Google) зазначають, що майбутнє всесвітньої мережі саме у системах надання рекомендацій. Пошук все ще має велике значення, але слід додавати до нього щось більш корисне – персоналізацію, що є для цих компаній ключовим напрямком дослідження.

Надання рекомендацій базується на перевагах користувача, що складають його профіль. Переваги можуть задаватися самим користувачем через присвоювання рейтингів елементам (як це реалізовано на www.yahoo.com, у системі пропонування фільмів www.movielenz.umn.edu чи на сайтах електронної комерції, наприклад, www.dell.com), або визначатися автоматично, базуючись на поведінці користувача, – які сторінки відвідував користувач, які файли завантажував, скільки часу витрачав на кожну сторінку, скільки разів сторінка відвідувалась і т. д. (прикладом такої реалізації є персональні рекомендації на www.amazon.com, рекомендовані списки музики для прослуховування на сайті mystrand.com, а також агенти, що фільтрують новини). Це породжує два підходи для отримання вищезазначеної інформації. Перший підхід базується на рейтингах, які користувачі призначають елементам, другий – на записах у системі про поведінку користувача. У системах, які розглянуті у даній роботі, переваги користувача визначено користувачем у вигляді рейтингів для елементів, тобто згідно першого з вищезазначених підходів.

Мета даного дослідження – розгляд методів і алгоритмів надання рекомендацій та їх оптимізація. Методи розглядалися у контексті систем електронного навчання, оскільки саме для таких систем планується їх практичне застосування.

Дана робота містить: огляд систем надання рекомендацій та опис обраної системи – системи сумісного фільтрування (перший розділ); позначення, які використані у формулах для експериментального

тестування методів систем сумісного фільтрування (другий розділ); огляд та оптимізацію методів систем сумісного фільтрування, запропонованих у світовій науковій літературі (третій розділ); додаткову оптимізацію методу, яка дозволяє покращити результат рекомендації (четвертий розділ); результати тестування роботи алгоритмів на реальних даних та порівняння результатів (п'ятий розділ).

Слід зазначити, що наведені в даній роботі методи та алгоритми можуть бути застосовані для будь-якої області, де доцільне надання рекомендацій, – Інтернет-магазинів, систем електронної комерції, систем новин, сайтів, що пропонують для завантаження фільми чи музику, та ін.

2. Аналіз систем сумісного фільтрування

Незалежно від способів отримання інформації про переваги користувачів, згідно [2] існує три типи систем надання рекомендацій, що розрізняються методами обробки інформації. Перший тип – це системи, засновані на сукупності правил. Для таких систем правила прийняття рішень закладаються в них при їх розробці, а інформація про користувача, що використовується при виконанні правил, – це інформація про загальні характеристики користувача (демографічна інформація, сфера діяльності, посада, набір інтересів). Другий тип – це системи, що базуються на фільтруванні змісту. В таких системах користувачу рекомендуються елементи, подібні до тих, до яких користувач виявив інтерес. Третій тип – системи сумісного фільтрування.

Найпростішою реалізацією систем першого типу є вітання користувача по імені чи автоматичне налаштування мови інтерфейсу залежно від регіону, де користувач проживає. Для завдання більш складних правил, слід здійснювати аналіз даних у системі та виявляти закономірності. Такий аналіз можна вважати підготовчим при побудові системи електронного навчання. Закладені правила в таких системах не змінюються при збільшенні кількості курсів та при діях користувачів.

Саме через зазначені обмеження системи такого типу не розглядалися у цій роботі.

Системи другого типу пропонують у вигляді рекомендації тільки ті елементи, які схожі на елементи з профілю користувача. Найбільшим обмеженням даного підходу є те, що елемент ніколи не буде рекомендовано, якщо подібний елемент не було оцінено чи відвідано. Тому для систем електронного навчання більш цікавим і доцільним є використання систем третього типу та комбінованих систем третього і другого типів. У даній роботі ми обмежилися системами надання рекомендацій третього типу – системами сумісного фільтрування, та розглянули методи покращення якості рекомендацій для таких систем.

Сумісним фільтруванням (collaborative filtering) називають надання рекомендацій системами, що базуються на аналізі поведінки користувача. Користувач порівнюється з іншими користувачами системи, а відомі переваги користувачів використовуються для прогнозування переваг нового користувача. Залежно від типу системи елементами, для яких визначаються переваги користувачів, можуть бути, наприклад, веб-сторінки, категорії і підкатегорії новин чи найменування товарів. У системах електронного навчання елементами є електронні курси та новини. Іншими словами, якщо два користувачі мають схожі оцінки для певної кількості елементів, то вважається, що вони мають схожі смаки, і ми можемо вважати їх сусідами. Традиційно для визначення сусідства використовується підхід, що має назву *k*-Найближчий-Сусід (*k*-Nearest-Neighbor, *k*NN), при якому профіль користувача порівнюється з профілями інших користувачів для знаходження *k* користувачів, які мають подібні смаки або інтереси (ці користувачі в рамках даного підходу називаються найближчими сусідами, де *k* є натуральним числом) [3]. Такий підхід має різні реалізації у залежності від того, яким чином визначаються рейтинги та «схожість» між користувачами чи близькість сусідства, та як визначається, яка кількість елементів потрібна для визначення схожості. Цей

метод найкраще працює, коли користувачі оцінюють великі кількості елементів, та для користувачів, чия поведінка не є аномальною (тобто не відрізняється від середньостатистичної).

Системи сумісного фільтрування мають обмеження в застосуванні через малу масштабованість (оскільки обчислення мають відбуватись у режимі реального часу, збільшення кількості користувачів та елементів може призвести до запізнення надання рекомендації) та зменшення покриття елементів протягом роботи користувача внаслідок збільшення кількості елементів у базі даних (що призведе до менш надійних кореляційних зв'язків). Для уникнення цих обмежень у [3] запропоновані методи оптимізації обробки даних (такі як індексування схожих об'єктів, зменшення розмірності, розбиття запитів користувачів на кластери в режимі попередньої обробки даних та накладання обмежень на операції, що можуть виконуватись в реальному часі).

Бріз та інші в [4] виділяють два підвиди підходу до побудов систем сумісного фільтрування. Перший, що базується на пам'яті, – це алгоритми по визначенню найближчих сусідів з різними метриками визначення сусідства, а також алгоритми кластеризації. Другий, що базується на моделях, – це алгоритми побудови моделей (наприклад, імовірнісних), в які можна вкласти поведінку користувача. Однак велика кількість прихованих параметрів, потрібних для побудови моделей, робить дані методи важкими для практичного застосування. Алгоритми, що базуються на пам'яті, переважно працюють у реальному часі, тоді як алгоритми, що базуються на моделях – спочатку виконують підготовчий етап, а далі видають рекомендацію у реальному часі. У даній роботі розглянуто алгоритми, що працюють у реальному часі та запропоновано модифікацію алгоритму, що дозволяє скоротити час обчислень.

3. Позначення

Введемо наступні позначення для формалізації математичної моделі. Нехай

U – це множина користувачів, $|U|=m$, $U = \{u_1, u_2, \dots, u_m\}$. I – множина елементів, $I = \{i_1, i_2, \dots, i_n\}$, $|I|=n$. Позначимо $I_a^{(u)}$ множину елементів, що не були оцінені (чи відвідані) користувачем u_a , а U_a – множину найближчих сусідів цільового користувача u_a .

Для користувача $u \in U$ профілем вважаємо вектор $u^{(n)}$ довжини n , що складається з упорядкованих пар: $u^{(n)} = \langle (i_1, s_u(i_1)), (i_2, s_u(i_2)), \dots, (i_n, s_u(i_n)) \rangle$, де $i_j \in I$, а s_u – функція, яка приписує елементам ваги, що відповідають ступеню інтересу для користувача u .

Профілі мають зберігатися протягом роботи системи для всіх користувачів. База даних всіх профілів може бути представлена як матриця розмірністю $m \times n$: $\Lambda = [s_{u_k}(i_j)]_{m \times n}$, де $s_{u_k}(i_j)$ є ступінь інтересу користувача u_k до елемента i_j .

4. Методи систем сумісного фільтрування

Системи сумісного фільтрування для надання рекомендацій мають містити алгоритми, що реалізують:

- підрахунок метрики для вимірювання близькості сусідства (ступеня схожості користувачів);
- метод для вибору множини сусідів для прогнозування;
- метод для прогнозування рейтингу елементів, ще не відвіданих користувачем;
- вибір певної кількості елементів з найбільшим значенням рейтингу, які і будуть результативною рекомендацією.

Система надання рекомендацій у вищенаведених позначеннях – це функція $СП_\Lambda: U \times I \rightarrow R \cup \{null\}$, де R – нова множина ваг елементів, за якими можна визначити множину рекомендованих елементів. Залежно від вибору алгоритму функція $СП$ може давати невизначений результат ($null$). Загалом функцію $СП$ можна розглядати як функцію прогнозування цікавості цільового користувача u_k щодо цільового елемента i_j : $СП_\Lambda(u_k, i_j) = s'_{u_k}(i_j)$.

У класичному варіанті методу сумісного фільтрування для вимірювання ступеня схожості користувачів найчастіше використовується метрика, що підраховується як нормалізований добуток двох векторів, які представляють собою вектори ступенів інтересу двох користувачів до елементів. Інший спосіб визначення сусідства згідно [5] полягає у використанні коефіцієнта кореляції Пірсона, який обрано у даній роботі для тестування роботи алгоритмів надання рекомендацій:

$$sim(u_a^{(n)}, u_b^{(n)}) = P_{a,b} =$$

$$= \frac{\sum_{j=1}^n (s_{u_a}(i_j) - \bar{s}_{u_a}) \times (s_{u_b}(i_j) - \bar{s}_{u_b})}{\sqrt{\sum_{j=1}^n (s_{u_a}(i_j) - \bar{s}_{u_a})^2 \times \sum_{j=1}^n (s_{u_b}(i_j) - \bar{s}_{u_b})^2}}$$

де \bar{s}_{u_a} та \bar{s}_{u_b} – середні рейтинги всіх елементів користувачів u_a та u_b відповідно.

Коли ступінь схожості цільового користувача з усіма іншими користувачами визначено, для подальших підрахунків з метою зменшення складності обчислень береться скінченна кількість користувачів з найбільшими значеннями ступеня схожості. Оптимальна кількість користувачів визначається для кожної системи індивідуально після тестування методів сумісного фільтрування на реальних даних системи і варіюється у середньому від 10 до 100. У даній роботі використана рекомендація Херлокера та інших, які показують доцільність використання 30 сусідів [5].

Після визначення множини сусідів користувача потрібен метод для підрахунку (прогнозування) рейтингів кожного елемента $i_j \in I_a^{(u)}$. Найпоширеніший підхід згідно [6] – це використання зваженої суми рангів. Очікуваний рейтинг $s_{u_a}'(i_j)$ елемента i_j для користувача u_a підраховується так:

$$s_{u_a}'(i_j) = \bar{s}_{u_a} + \frac{\sum_{u_k \in U_a} sim(u_a^{(n)}, u_k^{(n)}) \times (s_{u_k}(i_j) - \bar{s}_{u_k})}{\sum_{u_k \in U_a} sim(u_a^{(n)}, u_k^{(n)})}$$

де \bar{s}_{u_a} та \bar{s}_{u_k} – середні рейтинги всіх елементів користувачів u_a та u_k відповідно.

Для покращення якості рекомендацій у [7] пропонується ввести коефіцієнт значимості сусідства, що дозволить збільшити ступінь сусідства між користувачами, які оцінили разом велику спільну кількість елементів, та зменшити ступінь сусідства між користувачами, що мають мало спільних оцінок, проте у яких спільні оцінки дуже схожі. Даний коефіцієнт матиме тим більше значення, чим більше спільних елементів оцінили користувачі: $sg_{a,b} = \min\left(\frac{d}{50}, 1\right)$, де d – кількість елементів, які оцінили одночасно обидва користувачі a і b .

Іншою запропонованою в [7] оптимізацією методу є додавання до числа сусідів псевдо-активного користувача (переваги якого – це елементи $c_{u_a}(i_j)$, визначені за допомогою методу фільтрування змісту на профілі активного користувача). Для надання цьому псевдо-активному користувачу більшої значимості, вводиться коефіцієнт значимості псевдо-активного користувача $sw_a = \min\left(\frac{n_a}{50} \times cm, cm\right)$, де n_a – кількість елементів, яку оцінив активний користувач, а cm – ваговий коефіцієнт (пропонується покласти його 2 в [7]). Даний коефіцієнт тим більший, чим більшу кількість елементів оцінив активний користувач.

Враховуючи коефіцієнт значимості сусідства та додавання до числа сусідів псевдо-активного користувача, формула для обчислення очікуваного рейтингу $s_{u_a}'(i_j)$ елемента i_j для користувача u_a модифікується таким чином:

$$s_{u_a}'(i_j) = \bar{s}_{u_a} + \frac{sw_a (c_{u_a}(i_j) - \bar{s}_{u_a})}{sw_a + \sum_{u_k \in U_a} sg_{a,k} \times sim(u_a^{(n)}, u_k^{(n)})} + \frac{\sum_{u_k \in U_a} sg_{a,k} \times sim(u_a^{(n)}, u_k^{(n)}) \times (s_{u_k}'(i_j) - \bar{s}_{u_k})}{sw_a + \sum_{u_k \in U_a} sg_{a,k} \times sim(u_a^{(n)}, u_k^{(n)})}$$

(із сум доданків, що підраховуються на множині всіх сусідів, виносяться доданки, що відповідають доданому псевдо-

активному користувачу; кожен доданок, що відповідає сусіду, помножується на коефіцієнт значимості сусідства, а доданок, що відповідає псевдо-активному користувачу – на коефіцієнт значимості псевдо-активного користувача).

5. Удосконалення методу сумісного фільтрування

Проаналізувавши вищезазначені алгоритми та запропоновану оптимізацію, в даному розділі запропоновано додаткову модифікацію методу, що дозволило покращити деякі показники, а саме – великий час обчислень та велику розрідженість матриці переваг, які не були враховані у вищенаведеному методі.

Складність обчислень, тобто час, необхідний для надання рекомендацій новому користувачу в реальному часі, для алгоритму визначення найближчих сусідів складає $O(m)$. Для невеликої системи із 20 тисяч користувачів та 100 елементів час підрахунку рекомендації для користувача складає близько 20 с, що є дуже значним, оскільки за цей час користувач може вирішити припинити перегляд сторінки чи сайту, або перейти на іншу сторінку чи зазначити нові рейтинги, і це призведе до того, що сформована рекомендація буде застарілою та невірною. Дане обмеження можна обійти, якщо робити попередні обчислення. Проте такий метод не буде ефективним для систем, які мають велику кількість користувачів та велику кількість елементів, що постійно поповнюються.

З іншого боку, матриця переваг користувачів Λ у переважній більшості випадків дуже розріджена, тому визначити користувачів, значно схожих на цільового користувача, дуже складно. Багато науковців приділяли увагу розрідженості матриці переваг та способам заповнення матриці. Наприклад, виконувалась заміна нульових значень загальними рейтингами або найбільш імовірнісними значеннями, виділення принципів елементів [8, 9]. Прайор також показав, що використання лише множини принципів елементів зменшує розмірність матриці та робить обчислення більш ефективними [9]. Пропонувалось скористатись методом

фільтрування змісту для заміни нульових елементів ненульовими, що визначатимуться як середнє арифметичне рейтингів елементів, схожих із цільовим елементом за тематикою і оцінених користувачем, чи просто як середнє арифметичне рейтингів елементів, схожих із цільовим елементом за тематикою і визначених на множині всіх профілів [7].

Оскільки у випадку системи електронного навчання, яка розглядається, інформація про самі елементи – невідома, то методом фільтрування змісту скористатись неможливо. Тому запропоновано новий метод визначення множини суттєвих користувачів, що дозволяє, з одного боку, зменшити час обчислень, а з іншого – дає можливість зменшити розрідженість матриці переваг користувачів.

Матриця переваг користувачів Λ модифікується таким чином: в ній залишаємо тільки ті рядки, що відповідають користувачам, які оцінили принаймні 70 % елементів із множин елементів, оцінених цільовим користувачем. Даний відсоток елементів отриманий експериментально: це оптимальне значення, яке дозволяє суттєво скоротити розмір матриці переваг і відповідно час надання рекомендації, практично без зменшення якості рекомендації.

За рахунок розрідженості матриці Λ ця модифікація дозволила отримати значне скорочення кількості користувачів за постійний час і відповідну вхідну множину для обчислення ступеня кореляції. Дане перетворення також врахувало значимість сусідства, тому коефіцієнт $sg_{a,b}$ у формулі обчислення рекомендації покладаємо 1.

Оскільки система надання рекомендацій, що розглядалась, не містить інформацію про зміст елементів та відповідно не дозволяє визначити схожі елементи, запропонована наступна додаткова модифікація алгоритму. Вектор рейтингів псевдо-активного користувача визначено як середнє арифметичне рейтингів конкретного елемента для всіх користувачів. Така модифікація не є реалізацією методів фільтрування змісту і дозволила врахувати суттєвість кожного елемента, а також через дане перетворення було враховано вплив

користувачів, яких на першому кроці алгоритму було вилучено із множини користувачів, з якої обиралися найближчі сусіди. Формула для обчислення очікуваних рейтингів цільового користувача не змінилась.

6. Експериментальні результати

З метою перевірки ефективності запропонованого удосконалення методу сумісного фільтрування використовувалась база даних Джестер. Система Джестер – електронна система рекомендації жартів, створена Кеном Голдбергом та його командою у Берклі. База даних містить рейтинги 100 жартів, зібраних за період від квітня 1999 до травня 2003 року.

Дослідження існуючих публічних баз даних про поведінку користувачів у Інтернеті показують, що зазвичай розрідженість матриці переваг користувачів є великою і складає у середньому від 70 до 99 %. Для отримання адекватних результатів тестування проводилось не лише на розріджених матрицях. Внаслідок експериментів з декількома базами даних було виявлено, що за умови використання вищезазначеного перетворення при лінійному збільшенні розрідженості матриці, розмірність матриці зменшується з не меншою швидкістю, що призводить до відповідного зменшення часу формування рекомендацій. При цьому точність надання рекомендацій не зменшується або її зміна дуже незначна.

Кожен набір даних розділено на 2 частини – експериментальну (тренувальну) та тестову. Частина тестових даних використовувалась як вхідні дані про цільових користувачів. На базі тренувального набору виконувались алгоритми персоналізації та обчислювались рекомендації, точність яких обраховувалась шляхом порівняння з іншою частиною тестових даних.

Тестування проводилось на двох наборах даних. Перший містив 20000 записів тренувального набору та 4983 записів тестового набору. Розрідженість матриці була дуже низька і складала 27,5 %. Навіть за таких умов описане перетворення матриці рейтингів дозволило скоро-

тити тренувальний набір до 12977 записів (на 35 %). Другий набір даних містив 20000 записів тренувального набору та 4938 записів тестового набору. Для цього набору розрідженість матриці була висока і складала 75,3 %. Описане перетворення матриці рейтингів дозволило скоротити тренувальний набір до 532 записів (на 97,3 %). Як вище зазначалось, для визначення значення ступеня сусідства використовувався коефіцієнт кореляції Пірсона, а для формування рекомендації обиралися 30 найближчих сусідів. Коефіцієнт sw_a покладался 0, 1 та 2 для різних серій експериментів для того, щоб оцінити вплив псевдо-активного користувача на точність рекомендацій.

Для оцінювання ефективності алгоритму використовувались два показники: нормалізована абсолютна похибка і покриття.

Нормалізована абсолютна похибка (середня абсолютна похибка між рекомендацією та реальним результатом) відображає статистичну точність даних і для кожного користувача визначалась за

$$\text{формулою } NMAE_{u_k} = \frac{\sum_{j=1}^r |s_{u_k}(i_j) - s'_{u_k}(i_j)|}{r(s_{\max} - s_{\min})},$$

де $s_{u_k}(\cdot)$ – значення функції переваг користувача u_k ; $s'_{u_k}(\cdot)$ – значення рекомендації для цього користувача; r – кількість елементів, для яких обчислювались рекомендації; s_{\max} та s_{\min} – відповідно найбільше та найменше значення переваг (для системи Джестер складає відповідно +10 та -10). Для визначення загальної похибки обчислювалось середнє арифметичне нормалізованої абсолютної похибки за користувачами із тестового набору даних.

Покриття – відсоток елементів, цікавих для користувача, для яких алгоритм повертає рекомендації. Рекомендацією у експерименті вважались тільки ті елементи, для яких рекомендований рейтинг був більший за 0. Результати експерименту наведені в таблиці.

Таблиця. Результати роботи алгоритмів на двох наборах даних у відсотках (з розрідженою та нерозрідженою матрицею переваг)

Метод та набір даних	NMAE	Покриття
Класичний метод, нерозріджена матриця	15,31	82,4
Модифікований метод, нерозріджена матриця, коефіцієнт 0	15,54	86,4
Модифікований метод, нерозріджена матриця, коефіцієнт 1	15,32	87,0
Модифікований метод, нерозріджена матриця, коефіцієнт 2	15,17	87,14
Класичний метод, розріджена матриця	17,39	91,1
Модифікований метод, розріджена матриця, коефіцієнт 0	18,22	88,7
Модифікований метод, розріджена матриця, коефіцієнт 1	17,74	88,9
Модифікований метод, розріджена матриця, коефіцієнт 2	17,30	89,3

Під коефіцієнтом мається на увазі коефіцієнт sw_a .

Отримані результати показали досить високу точність, оскільки для надання рекомендації враховувалась велика кількість зроблених оцінок цільовим користувачем. Як видно із таблиці, значне скорочення часу обчислень (для першого набору – у 2 рази, для другого набору – у 50 разів) було отримане без погіршення якості рекомендацій: нормалізована абсолютна похибка зменшилась, а покриття рекомендацій збільшилось у першому випадку і варіювалось у невеликому діапазоні у другому випадку. Слід зазначити, що найкращий результат показав модифікований метод із найбільшим значенням коефіцієнта значимості псевдо-активного користувача, що свідчить про позитивний вплив модифікації алгоритму на точність рекомендацій.

Висновки

Актуальність систем надання рекомендацій зростає разом із зростанням кількості електронної інформації. Такі системи визнані пріоритетним напрямком розробок провідних світових компаній. У даній

роботі розглянуто типи систем надання рекомендацій та алгоритми і методи, що використовуються у системах сумісного фільтрування. Визначено, які типи систем надання рекомендацій актуально застосовувати для систем електронного навчання. Запропоновано оптимізацію алгоритму надання рекомендацій та показано ефективність запропонованого методу шляхом його тестування на великому масиві реальних даних. Запропонований підхід можна застосовувати також до інших систем, де доцільне надання рекомендацій.

1. *Catone J.* Yahoo!: The Web's Future Is Not In Search. – June 4 2007 // http://www.readwrite.com/archives/yahoo_personalization.php.
2. *Mobasher B., Dai H., Luo T., Nakagawa M.* Improving the Effectiveness of Collaborative Filtering on Anonymous Web Usage Data // IJCAI 2001, Workshop on Intelligent Techniques for Web Personalization (ITWP). – Seattle. – August 2001. – P. 53–60.
3. *Mobasher B., Brusilovsky, P., Kobsa A., Nejdl W.* Data mining for Web personalization. The Adaptive Web: Methods and Strategies of Web Personalization // Lecture Notes in Computer Science. – Springer-Verlag, Berlin-Heidelberg New York. – 2006. – 762 P. – P. 90–135.

4. *Breese J., Heckerman D., Kadie C.* Empirical analysis of predictive algorithms for collaborative filtering // Fourteenth Annual Conference on Uncertainty in Artificial Intelligence. – July 1998. – P. 43–52.
5. *Herlocker J., Konstan J., Borchers A., Riedl J.* An algorithmic framework for performing collaborative filtering // 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR). – 1999. – P. 230–237.
6. *Anand S.S., Mobasher B.* Intelligent Techniques for Web Personalization // LNAI 3169. – Springer-Verlag, Berlin-Heidelberg New York. – 2005. – P. 1–36.
7. *Melville P., Mooney R., Nagarajan R.* Content-booster collaborative filtering for improved recommendations // 18th National Conference on Artificial Intelligence (AAAI). – 2002. – P. 187–192.
8. *Billsus D., Pazzani M.* Learning collaborative information filters // Fifteenth International Conference on Machine Learning. – Madison, USA. – 1998. – P. 46–54.
9. *Pryor M.* The effects of singular value decomposition on collaborative filtering // Dartmouth College CS Technical Report, PCS-TR98-338. – 1998.

Отримано 09.10.2007

Про автора:

Горностай Марія Павлівна,
аспірантка факультету кібернетики
Київського національного університету
імені Тараса Шевченка.

Місце роботи автора:

Київський національний університет
імені Тараса Шевченка.
Київ, просп. Академіка Глушкова 2,
корпус б.
Тел.: +38(050) 352 5252.
e-mail: Gornostay.m@gmail.com