УДК 004.9 **doi: 10.26906/SUNZ.2019.6.058**

Є. В. Мелешко, В. Д. Хох, В. В. Босько

Центральноукраїнський національний технічний університет, Кропивницький, Україна

ДОСЛІДЖЕННЯ МАТРИЧНИХ ФАКТОРИЗАЦІЙНИХ МОДЕЛЕЙ РЕКОМЕНДАЦІЙНИХ СИСТЕМ

Анотація. Об'єктом дослідження є процес створення списків рекомендацій відвідувачам веб-сайтів. Метою даної роботи є дослідження існуючих матричних факторизаційних моделей рекомендаційних систем. У рекомендаційних системах факторизація застосовується до матриці рейтингів з метою виявляння прихованих факторів, властивих об'єктам системи, що впливають на вподобання користувачів. Матричні факторизаційні моделі рекомендаційних систем досить популярні серед розробників та мають багато модифікацій. У даній роботі розглянуто наступні моделі: FunkSVD, SVD++, Asymmetric SVD та timeSVD. Факторизаційні моделі рекомендаційних систем використовуються у методах колаборативної фільтрації на рівні з моделями на основі сусідства. На відміну від моделей на основі сусідства, які використовують коефіцієнти подоби для створення списків рекомендацій, дані моделі використовують не подобу, а приховані фактори. Перевагами таких моделей є підвищена, порівняно з іншими моделями, робастність до атак ін'єкцією профілів та висока точність прогнозування вподобань користувачів. До недоліків досліджуваних моделей слід віднести погану масштабованість, довгий час навчання, а також необхідність повного перенавчання системи при появі нових даних, що частково вирішено лише у асиметричному SVD. Проведене дослідження показало, що існуючі моделі матричної факторизації дають можливість використовувати як явні зворотні зв'язки від користувачів (рейтинги об'єктів, виставлені користувачами), так і неявні зворотні зв'язки (перегляди об'єктів, написані коментарі, тощо), що дозволяє підвищувати точність роботи рекомендаційної системи на веб-ресурсах, де користувачі залишають багато неявного зворотного зв'язку. Такий принцип вперше був реалізований у SVD++. Факторизаційні моделі дозволяють також враховувати неперіодичні та періодичні зміни вподобань користувачів у часі, що, зокрема, реалізовано у timeSVD.

Ключові слова: рекомендаційні системи, матрична факторизація, SVD, приховані фактори, градієнтний спуск, прогнозування.

Вступ

На сьогоднішній день, при побудові рекомендаційних систем (РС) на основі колаборативної фільтрації, часто застосовують матричні факторизаційні моделі (МФМ) вподобань користувачів [1].

На відміну від моделей РС на основі сусідства, які використовують коефіцієнти подоби для створення списків рекомендацій, факторизаційні моделі виявляють та використовують приховані фактори (ПФ), що впливають на вподобання користувачів [1-3].

Факторизація – це процес декомпозиції об'єкту (зокрема, матриці) в набір інших об'єктів (факторів), добуток яких дає початковий об'єкт [4]. Факторизація дозволяє виділити ключові компоненти об'єкту факторизації.

У РС факторизація застосовується до матриці рейтингів з метою виявляння прихованих факторів, властивих об'єктам системи, що впливають на вподобання користувачів.

Метою даної роботи є дослідження існуючих матричних факторизаційних моделей для РС.

Найбільш відомими МФМ є FunkSVD, SVD++, Asymmetric SVD, timeSVD [1, 5-10]. Усі ці моделі одержали назву від методу факторизації матриць (ФМ) Singular value decomposition (сингулярний розклад матриць), хоча безпосередньо його вони не використовують, а лише засновані на спільній з ним ідеї – одержати для певної матриці деякі матриці, добуток яких дасть матрицю наближену до початкової.

У випадку з матрицями рейтингів, ця одержана наближена матриця буде містити наближені дані у відомих рейтингах, а в комірках, де в початковій матриці рейтинги були невідомі, з'являться прогнозовані рейтинги.

Перша МФМ РС була запропонована Сімоном Фанком під час конкурсу від Netflix у 2006. У своїй публікації у власному блозі [5], Фанк визначив матрицю рейтингу користувача-об'єкта як добуток двох матриць пониженого рангу, перша – рядки прихованих факторів для користувачів, а друга – стовпчики прихованих факторів для об'єктів. Множення рядка користувача на стовпчик об'єкта дає прогнозований рейтинг для відповідної пари користувач-об'єкт.

Усі наступні матричні факторизаційні моделі є покращеними модифікаціями моделі FunkSVD.

Основний матеріал

Матричні факторизаційні моделі РС засновані на ідеї факторизації матриці рейтигів для виявлення та використання прихованих факторів, які впливають на вподобання користувачів. Приховані фактори дозволяють заповнити відсутні у матриці рейтингів комірки, тобто прогнозувати рейтинги.

Основна ідея факторизації матриць рейтингів рекомендаційної системи зображена на рис. 1.

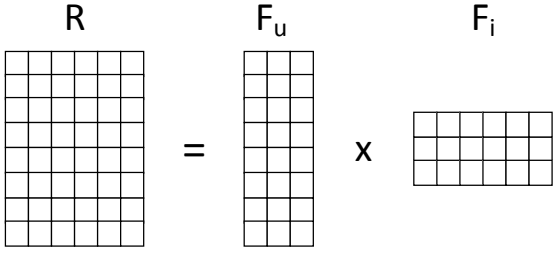

Рис. 1. Принцип факторизації матриці рейтингів

Матриця рейтингів R розмірності $n \times m$, де n – кількість користувачів, m – кількість об'єктів, факторизується на дві матриці: матрицю ПФ користувачів F_u, розмірності $n \times k$, де k – кількість ПФ, та матрицю ПФ об'єктів F_i, розмірності $k \times m$.

Загальний алгоритм ФМ рейтингів:

1. Ініціалізуємо матриці F_u та F_i випадковими значеннями.

2. Перемножуємо матриці F_u та F_i і порівнюємо результат з R, обчислюємо помилки.

3. Мінімізуємо помилки за допомогою деякого алгоритму машинного навчання (напр., градієнтного спуску, методу найменших квадратів, тощо).

Розглянемо конкретні реалізації даного загального принципу факторизації матриці рейтингів.

FunkSVD. Найперша модель рекомендаційних систем, що застосовує матричну факторизацію [5].

Дана модель полягає у наступному.

Спочатку треба визначити базові предиктори (зміщення) $b_{u,i}$, які складаються з базових предикторів окремих користувачів b_u і базових предикторів окремих об'єктів b_i, а також просто загального середнього рейтингу об'єктів у системі μ:

$$b_{u,i} = \mu + b_u + b_i . \quad (1)$$

Для прогнозування оцінки для пари користувач-об'єкт використовується така формула:

$$\hat{r}_{u,i} = \mu + b_u + b_i + q_i \cdot p_u . \quad (2)$$

де q_i – вектор факторів об'єкту i, а p_u – вектор факторів користувача u.

На початку роботи алгоритму треба обчислити глобальну середню оцінку та усі предиктори.

Потім треба знайти найкращі предиктори та фактори, що дозволяють прогнозувати рейтинги з найменшою помилкою.

Для визначення помилки використовується сума квадратів відхилень:

$$E = \sum_{(u,i) \in D} \left(r_{u,i} - \hat{r}_{u,i} \right)^2 , \quad (3)$$

$$E = \sum_{(u,i) \in D} \left(r_{u,i} - \mu - b_u - b_i - q_i \cdot p_u \right)^2 , \quad (4)$$

де $r_{u,i}$ – справжній рейтинг об'єкту i у користувача u; $\hat{r}_{u,i}$ – прогнозований рейтинг.

Дана функція оптимізується градієнтним спуском, беруться часткові похідні по кожному аргументу, а рух під час градієнтного спуску відбувається у сторону зворотню напрямку цих похідних. Для одержання адекватних результатів при роботі з реальними даними необхідно враховувати ймовірність оверфітінгу [11, 12] (перенавчання системи) та виконувати регуляризацію [11, 13], щоб подолати дану проблему.

Регуляризація – додавання деякої додаткової інформації, щоб знайти рішення некоректно поставленої задачі, або щоб уникнути перенавчання.

Загалом регуляризуючий вираз $R(f)$ додається до значення помилки, перед тим як визначати аргумент, що дає найменше значення помилки:

$$f_* = \arg\min_f \sum_{i=1}^{n} E\left(f\left(\hat{x}_i\right), \hat{y}_i \right) + \lambda \cdot R(f) , \quad (5)$$

де E – функція, що визначає похибку передбачення $f(x)$ для значень y, а параметр λ визначає важливість доданка для регуляризації. Зазвичай $R(f)$ визначається як штраф за складність функції f. Зокрема, поняття складності включає обмеження на гладкість та на норму векторного простору.

В FunkSVD оптимізаційний вираз можна записати таким чином:

$$b_*, q_*, p_* = \arg\min_{b,q,p} \sum_{(u,i)}^{n} (r_{u,i} - \mu - b_u - b_i - q_i \cdot p_u)^2 + \lambda \left(\sum_u b_u^2 + \sum_i b_i^2 + \|q_i\|^2 + \|p_u\|^2 \right), \quad (6)$$

де λ – параметр регуляризації.

Якщо взяти від (6) часткові похідні по кожній із змінних, що оптимізуються, отримаємо прості правила для градієнтного спуску.

Під час градієнтного спуску у FunkSVD використовуються наступні правила для оптимізації змінних, що впливають на результат:

$$b_u = b_u + \gamma \left(e_{u,i} - \lambda b_u \right), \quad (7)$$

$$b_i = b_i + \gamma \left(e_{u,i} - \lambda b_i \right), \quad (8)$$

$$q_{i,k} = q_{i,k} + \gamma \left(e_{u,i} \cdot p_{u,k} - \lambda q_{i,k} \right), \quad (9)$$

$$p_{u,k} = p_{u,k} + \gamma \left(e_{u,i} \cdot q_{i,k} - \lambda p_{u,k} \right), \quad (10)$$

де $e_{u,i} = r_{u,i} - \hat{r}_{u,i}$ – помилка на наборі даних для навчання; γ - швидкість навчання.

Можна очікувати більшої точності, виділивши окремі швидкості навчання γ_n і регуляризації λ_n для кожного типу досліджуваного параметра. Так, напр., рекомендується використовувати різні швидкості навчання для зсувів користувачів, зсувів об'єктів і самих факторів.

Важливо, що при такому підході невідомо, які саме характеристики об'єктів відповідають факторам. Тому дані моделі є неінтерпретуємими.

SVD++. Відрізняється від FunkSVD тим, що крім рейтингів (явного зворотного зв'язку від користувача) використовує також неявну інформацію про вподобання користувачів, напр., перегляди об'єктів, написання коментарів, тощо [6, 7].

Точність прогнозування у SVD++ поліпшується за рахунок врахування неявного зворотного зв'язку, який забезпечує додаткову індикацію вподобань користувачів. Це особливо корисно для тих користувачів, які надали більше неявного зворотного зв'язку, ніж явного.

Для врахування неявного зворотного зв'язку (одного типу) від користувачів використовується другий набір факторів об'єктів, що пов'язує кожен об'єкт із вектором факторів $y_i \in R(u)$. Ці нові фактори об'єктів використовуються для характеристики користувачів на основі набору об'єктів, які вони неявно оцінили. У такій моделі рейтинги прогнозуються наступним чином:

$$\hat{r}_{u,i} = \mu + b_u + b_i + q_i \times \times \left(p_u + \left|R(u)\right|^{-1/2} \sum_{j \in R(u)} y_j \right). \quad (11)$$

Набір $R(u)$ містить усі об'єкти, що були неявно оцінені користувачем u.

Характеристики користувача u моделюються як

$$p_u + \left|R(u)\right|^{-\frac{1}{2}} \sum_{j \in R(u)} y_j .$$

Оскільки y_j – центровані навколо нуля (регуляризацією), сума нормалізується на $\left|R(u)\right|^{-\frac{1}{2}}$, щоб стабілізувати її дисперсію в межах діапазону спостережуваних значень $\left|R(u)\right|$.

Параметри моделі визначаються шляхом мінімізації відповідної регуляризованої квадратичної функції помилок за допомогою стохастичного градієнтного спуску. Оптимізація змінних, що впливають на результат прогнозу, обчислюється таким чином:

$$b_u = b_u + \gamma\left(e_{u,i} - \lambda_1 b_u\right), \quad (12)$$

$$b_i = b_i + \gamma\left(e_{u,i} - \lambda_1 b_i\right), \quad (13)$$

$$q_{i,k} = q_{i,k} + \gamma \times \times \left(e_{u,i} \cdot \left(p_{u,k} + \left|R(u)\right|^{-\frac{1}{2}} \sum_{j \in R(u)} y_j \right) - \lambda_2 q_{i,k} \right), \quad (14)$$

$$p_{u,k} = p_{u,k} + \gamma\left(e_{u,i} \cdot q_{i,k} - \lambda_2 p_{u,k}\right), \quad (15)$$

$$\forall j \in R(u) : y_j \leftarrow y_j + + \gamma(e_{u,i} \cdot \left|R(u)\right|^{-\frac{1}{2}} q_{i,k} - \lambda_2 y_j), \quad (16)$$

Можна враховувати і декілька типів неявного зворотного зв'язку. Напр., якщо користувач u має два типи неявного оцінювання об'єктів: додавання в обране $N_1(u)$ та перегляд сторінки $N_2(u)$, тоді рейтинги можна прогнозувати таким чином:

$$\hat{r}_{u,i} = \mu + b_u + b_i + q_i \times \times \left(\begin{array}{c} p_u + \left|N_1(u)\right|^{-\frac{1}{2}} \sum_{j \in N_1(u)} y_{1j} + \\ + \left|N_2(u)\right|^{-\frac{1}{2}} \sum_{j \in N_2(u)} y_{2j} \end{array} \right). \quad (17)$$

Відносна важливість кожного джерела неявного зворотного зв'язку буде автоматично визначена алгоритмом шляхом встановлення відповідних значень параметрів моделі.

Asymmetric SVD. Асиметричний SVD дозволяє додавати до моделі нових користувачів з декількома рейтингами, без необхідності перенавчати всю модель [8, 9]. При додаванні нового користувача, його приховані фактори обчислюються наступним чином:

$$p_{u,k} = \sigma\left(b + \sum_{j \in R(u)} w_{r_{uj}} s_j \right), \quad (18)$$

де σ – сигмоїдна функція:

$$\sigma(x) = \frac{1}{1 + e^{-x}};$$

r_{uj} рейтинг, який користувач u поставив об'єкту j; $R(u)$ – множина об'єктів, оцінених користувачем u, з відомими рейтингами; w, b, s – параметри, що шукаються градієнтним спуском.

Якщо необхідно використовувати неявні зворотні зв'язки, тоді для Asymmetric SVD можна застосувати наступну формулу:

$$p_{u,k} = \left|R(u)\right|^{-\frac{1}{2}} \sum_{j \in R(u)} r'_{u,j} \cdot s_j, \quad (19)$$

де $r'_{u,j} = r_{i,j} - \left(\mu + b_u + b_i\right)$;

r_{uj} рейтинг, який користувач u поставив об'єкту j; $R(u)$ – множина об'єктів, оцінених користувачем u, з відомими рейтингами; μ, b, s параметри, що шукаються градієнтним спуском.

TimeSVD++. Це факторизаційна модель з врахуванням часу [10]. Вподобання користувачів можуть залежати від часу, саме це враховує дана модель.

Популярність товару може змінюватися з часом. Це можна врахувати, трактуючи зміщення об'єкту b_i як функцію часу. Користувачі з часом можуть змінювати свої вподобання. Це можна також врахувати, прийнявши зміщення користувача b_u як функцію часу. Тоді базові предиктори будуть розраховуватися наступним чином:

$$b_{u,i} = \mu + b_u\left(t_{u,i}\right) + b_i\left(t_{u,i}\right). \quad (20)$$

Тут, $b_u\left(t_{u,i}\right)$ та $b_i\left(t_{u,i}\right)$ – реальні функції, які змінюються з часом. Точний спосіб побудови цих функцій повинен відображати розумний спосіб параметризації задіяних часових змін. Напр., у випадку рейтингу фільму очікується, що прихильність до фільму щодня трохи коливатиметься, а сильно змінюватиметься протягом більш тривалих періодів. З іншого боку вподобання користувачів можуть змінюватися щодня, відображаючи природню для поведінки клієнтів мінливість.

Це вимагає обрання менших проміжків часу при моделюванні поведінки користувача та більших

проміжків часу для моделювання часових ефектів, пов'язаних з об'єктами.

Базові предиктори окремих об'єктів можна визначити за такою формулою:

$$b_i(t) = b_i + b_{i,Bin(t)}, \quad (21)$$

де b_i – незмінна у часі частина предиктору об'єкта; $b_{i,Bin(t)}$ – частина предиктору об'єкта, що змінюється у часі, $bin(t)$ – номер часового проміжку, на які поділені усі наявні для навчання системи дані.

Для параметризації часової поведінки користувача з різною складністю та точністю можна розглядати різні функції.

Простий спосіб моделювання використовує лінійну функцію для фіксації можливого поступового зміщення вподобань користувачів. Для кожного користувача u позначаємо середню дату рейтингу за допомогою t_u.

Тепер, якщо користувач оцінив фільм у момент часу t, то пов'язане з цим відхилення часу визначається як:

$$dev_u(t) = sign(t - t_u) \cdot |t - t_u|^{\beta}, \quad (22)$$

де $|t - t_u|$ вимірює кількість умовних часових одиниць (напр., днів) між датами t і t_u.

Значення β встановлюється шляхом перехресної перевірки.

Базові предиктори окремих користувачів можна визначити за допомогою формули (23) або (24).

$$b_i(t) = b_u + \alpha_u \cdot dev_u(t), \quad (23)$$

де α_u – параметр алгоритму, який слід визначати під час градієнтного спуску.

Ця проста лінійна модель для наближення поведінки користувача вимагає визначення двох параметрів b_u та α_u для кожного користувача.

Більш гнучку параметризацію пропонують сплайни. Нехай u є користувачем, пов'язаним з n_u рейтингами. Часові точки k_u розподілені рівномірно по датах рейтингів користувача u як ядра, які керують такою функцією:

$$b_i(t) = b_u + \frac{\sum_{l=1}^{k_u} e^{-\sigma|t-t_l^u|} b_{t_l}^u}{\sum_{l=1}^{k_u} e^{-\sigma|t-t_l^u|}}, \quad (24)$$

де параметри b_u та t_l асоціюються з контрольними точками (ядрами) і автоматично дізнаються з даних.

Таким чином, вподобання користувача формується як зважена в часі комбінація цих параметрів. Кількість ядер k_u, врівноважує гнучкість та ефективність обчислень. Постійна σ визначає плавність сплайну.

При використанні формул (21-23) для визначення базових предикторів, одержемо наступну формулу для запису оптимізаційного виразу:

$$b_{u*}, b_{u,t*}, b_{i*}, b_{i,Bin(t)*}, б_{u*} = \arg\min \sum_{(u,i)}^{n} \begin{pmatrix} r_{u,i} - м - b_u - \\ -б_u dev_u(t_{u,i}) - \\ -b_{u,t} - b_i - b_{i,Bin(t)} \end{pmatrix}^2 + л\left(b_u^2 + б_u^2 + b_{u,t}^2 + b_i^2 + b_{i,Bin(t)}^2\right), \quad (25)$$

Можна використовувати ту саму методологію для отримання більшої кількості часових ефектів. Напр., для фіксації періодичних ефектів. Деякі об'єкти можуть бути більш популярними в конкретні пори року або впродовж певних свят. Періодичні ефекти можна знайти і для користувача. Напр., користувач може мати різні схеми купівлі протягом вихідних у порівнянні з робочим тижнем. Спосіб моделювання таких періодичних ефектів – виділити параметр для комбінацій періодів часу з об'єктами або користувачами.

В такому разі базові предиктори можна розраховувати за наступними формулами:

$$b_i(t) = b_i + b_{i,Bin(t)} + b_{i,period(t)}, \quad (26)$$

$$b_u(t) = b_u + \alpha_u \cdot dev_u(t) + b_{u,t} + b_{u,period(t)}. \quad (27)$$

Приховані фактори користувачів для timeSVD без врахування періодичності, а лише з врахуванням зміщень у часі, можна визначити так:

$$p_{u,k}(t) = p_{u,k} + \alpha_{u,k} \cdot dev_u(t) + p_{u,k,t}, \quad k = 1,...,f. \quad (28)$$

Рейтинги у TimeSVD++ можна прогнозувати за наступною формулою:

$$\hat{r}_{u,i} = \mu + b_u(t) + b_i(t) + q_i \cdot \left(p_u(t) + |R(u)|^{-\frac{1}{2}} \sum_{j \in R(u)} y_j\right). \quad (29)$$

Отже, існує багато різних моделей поведінки користувача РС, заснованих на ФМ для визначення прихованих факторів. Усі вони успішно застосовуються на різних існуючих веб-ресурсах.

Перевагами таких моделей є висока робастність до атак та висока точність прогнозування вподобань. До недоліків слід віднести погану масштабованість, довгий час навчання, а також необхідність перенавчання РС при появі нових даних, цей останній недолік частково вирішено у Asymmetric SVD.

Висновки

Було проведено дослідження існуючих матричних факторизаційних моделей рекомендаційних систем. А саме були розглянуті наступні моделі: FunkSVD, SVD++, Asymmetric SVD та timeSVD.

МФМ дозволяють виявляти приховані фактори, що впливають на вподобання користувачів.

Дані моделі РС більш стійкі до атак ін'єкцією профілів, ніж, напр., моделі на основі сусідства з використанням коефіцієнтів подоби.

Крім явного зворотного зв'язку від користувача (рейтингів), факторизаційні моделі можуть також враховувати неявні зворотні зв'язки (перегляди, коментарі, додавання у вибране, тощо), що дозволяє підвищувати точність роботи системи. Такий принцип вперше був реалізований у SVD++.

Переважна більшість матричних факторизаційних моделей потребує при додаванні нових користувачів повного перенавчання усієї моделі. Безумовно це є недоліком та вимагає затрат великої кількості ресурсів та часу.

Тим не менше, цей недолік можна усунути, що реалізовано у Asymmetric SVD – дана модель дозволяє додавати нових користувачів без перенавчання усієї моделі, а лише визначати їх приховані фактори для інтеграції їх до моделі.

Факторизаційні моделі дозволяють враховувати неперіодичні та періодичні зміни вподобань користувачів у часі, що, зокрема, реалізовано у timeSVD.

Дані моделі підходять для РС, у яких існує багато параметрів у об'єктів, важливий захист від атак ін'єкцією профілів і немає високих вимог до масштабування системи, а також уже накопичена деяка статистика для попереднього навчання системи.

СПИСОК ЛІТЕРАТУРИ

1. "Recommender Systems Handbook" (2010) Editors Francesco Ricci, Lior Rokach, Bracha Shapira, Paul B. Kantor, 1st edition, New York, NY, USA: Springer-Verlag New York, Inc., 842 p., doi: https://doi.org/10.1007/978-0-387-85820-3
2. Jones, M. (2013) "Recommender systems, Part 1. Introduction to approaches and algorithms. Learn about the concepts that underlie web recommendation engines", URL: https://www.ibm.com/developerworks/opensource/library/os-recommender1/index.html?s_tact=105agx99&s_cmp=cp
3. Meleshko, E.V., Semenov, S.G., Khokh, V.D. (2018) "Research of methods of building advisory systems on the internet", Academic Journal "Control, Navigation and Communication Systems", Issue 1(47), Poltava National Technical Yuri Kondratyuk University, Poltava, pp. 131–136, doi: https://doi.org/10.26906/SUNZ.2018.1.131 (in Ukrainian)
4. Krupnik, I. (1992) "Decomposition of a monic matrix polynomial into a product of linear factors", Linear Algebra Appl, P. 239-242.
5. Funk, S. (2006) "Netflix Update: Try This at Home", URL: https://sifter.org/~simon/journal/20061211.html
6. Cao, J., Hu, H., Luo, T., Wang, J., Huang, M., Wang, K., Wu, Zhonghai, Zhang, X. (2015). "Distributed Design and Implementation of SVD++ Algorithm for E-commerce Personalized Recommender System", Communications in Computer and Information Science. 572. Springer Singapore. pp. 30-44. doi: https://doi.org/10.1007/978-981-10-0421-6_4
7. Jia, Ya. (2014) "Users' brands preference based on SVD++ in recommender systems", IEEE Workshop on Advanced Research and Technology in Industry Applications. pp. 1175-1178. doi: https://doi.org/10.1109/wartia.2014.6976489
8. Koren, Y. (2008) "Factorization meets the neighborhood", Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, doi: https://doi.org/10.1145/1401890.1401944
9. Töscher, A., Jahrer, M., Bell, R.M. (2009) "The BigChaos Solution to the Netflix Grand Prize", Netflix prize documentation, URL: https://www.netflixprize.com/assets/ GrandPrize2009_BPC_BigChaos.pdf
10. Koren, Ye. (2009) "Collaborative filtering with temporal dynamics", Proceeding KDD '09 Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, P. 447-456.
11. "Overfitting in machine learning: what it is and how to prevent it", (2017) URL: https://elitedatascience.com/overfitting-in-machine-learning
12. Brownlee, J. (2016) "Overfitting and underfitting with machine learning algorithms", URL: https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms
13. Neumaier, A. (1998) "Solving ill-conditioned and singular linear systems: A tutorial on regularization", SIAM Review 40, P. 636–666.

Received (Надійшла) 11.10.2019
Accepted for publication (Прийнята до друку) 13.11.2019

The research of matrix factorization models of recommendation systems

Ye. Meleshko, V. Khokh, V. Bosko

Abstract. The subject matter of the article is the process of creating recommendation lists for website users. The goal is to research the existing matrix factorization models of recommendation systems. In recommendation systems, factorization is applied to a rating matrix in order to identify latent factors inherent in system objects that affect user preferences. Matrix factorization models of recommendation systems are very popular among developers and have many modifications. In this paper, the following models are considered: FunkSVD, SVD++, Asymmetric SVD, and timeSVD. Factorization models of recommender systems along with neighborhood models are used in collaborative filtering methods. Unlike neighborhood models, which use similarity coefficients for create lists of recommendations, factorization models do not use similarity, but latent factors. The advantages of such models are: increased robustness to attacks of profile-injection, in comparison with other models, and high accuracy in predicting user preferences. The disadvantages of the researched models include poor scalability, a long training time, and the need for a complete retraining of the system when new data appears, which is partially eliminated only in asymmetric SVD. The research showed that the existing matrix factorization models make it possible to use both explicit feedbacks from users (item ratings put up by users) and implicit feedbacks (views of items, comments, etc.), which allows to increase the accuracy of a recommendation system on web-resources where users give a lot of implicit feedback. This principle was first implemented in SVD++. Factorization models also allow taking into account non-periodic and periodic changes in user preferences over time, which, in particular, is implemented in timeSVD.

Keywords: recommendation systems, matrix factorization, SVD, latent factors, gradient descent, forecasting.