УДК 004.65: 004.9

Є. В. Левус, Н. І. Нечипір, Ю. В. Полиняк
Національний університет “Львівська політехніка”,
кафедра програмного забезпечення

АНАЛІЗ АЛГОРИТМУ APRIORI ДЛЯ СТРУКТУРОВАНИХ ТА НЕСТРУКТУРОВАНИХ ДАНИХ

Проаналізовано алгоритм Apriori як метод пошуку асоціативних правил у структурованих та неструктурованих даних з погляду кількості знайдених правил, швидкодії та потреб в обчислювальних ресурсах. Неструктуровані дані тісно пов’язані з терміном Big Data. Актуальним завданням інженерії даних є виявлення ефективних засобів опрацювання неструктурованої інформації. Для проведення обчислювальних експериментів розроблено програмну систему, що опрацьовує дані алгоритмом Apriori, предметною областю якої вибрано торгівлю. Така система може бути прототипом реальної рекомендаційної системи. Програмне рішення розроблено на стеку технологій Hadoop.

Ключові слова: алгоритм Apriori, структуровані дані, неструктуровані дані, асоціативне правило, обсяг даних.

Apriori algorithm is analyzed as a search method of associative rules in structured and unstructured data in terms of the number of discovered rules, performance and requirements for computing resources. Unstructured data are closely related to the term ’Big Data’. One of the main tasks of data engineering is the detection of unstructured information processing means. There has been developed a software system to perform computational experiments that processes data using Apriori algorithm, which subject area is trade. Such system can be a prototype for real recommendation system. The software solution is developed on stack of Hadoop technology.

Key words: Apriori algorithm, structured data, unstructured data, associative rule, data volume.

Вступ

Накопичена інформація містить величезний потенціал для отримання нових знань і може надавати багато нових можливостей для прийняття рішень [1]. Структуровані дані доволі легко аналізувати, тоді як неструктуровані дані опрацьовуються з похибками або й зовсім непридатні до опрацювання. Останніми роками спостерігається тенденція до зменшення зростання обсягу структурованих даних і значно більшого зростання обсягу частково структурованих чи неструктурованих даних. Тому актуальними є завдання виявлення серед відомих методів і засобів ефективних для опрацювання неструктурованої інформації.

В умовах конкуренції компаніям важливо знайти найкращий спосіб не просто виявляти приховані закономірності та взаємозв’язки у них, але й отримувати інформацію в режимі реального часу, в результаті чого швидко приймати управлінські рішення, основані на якісних даних [1, 2].

Алгоритми для знаходження асоціативних правил стали одним з найпопулярніших існуючих методів для виявлення закономірностей у даних [3]. Рішення провідних компаній, наприклад Oracle [4], містять всі необхідні функції інтелектуального аналізу даних, але надто дорогі, а вартість підтримки рішень такого класу дуже висока.

Опрацювання великих обсягів даних і задача пошуку асоціативних правил

Провідні бізнес-компанії й організації накопичили величезні обсяги даних про клієнтів, товари, певні види послуг, обладнання тощо. Проте ця інформація у багатьох випадках “простоює”

та не дає ніякої користі. Необхідно також зауважити, що практично 75 % усієї інформації є неупорядкованою та неструктурованою. Очевидно, що аналізувати неструктуровані дані набагато складніше, ніж структуровані. Щоб вирішити таку проблему, організації звернулися до низки різних програмних рішень, призначених для генерації важливої інформації з неструктурованих даних.

Термін Big Data тісно пов'язаний з неструктурованими даними. Великі дані можуть містити як структуровані, так і неструктуровані дані, проте, за оцінками фахівців, до 90 відсотків великих даних – саме неструктуровані дані. Багато інструментів, призначених для аналізу даних великих обсягів, можуть опрацьовувати неструктуровані дані [3].

Разом зі стрімким накопиченням інформації швидкими темпами розвиваються методи і технології аналізу даних. Інтереси споживача можна проаналізувати, і відповідно до побудованої моделі вибрано відповідну рекламу або конкретні пропозиції. Модель також може налаштовуватися і перебудовуватися в режимі реального часу, що було неможливо ще кілька років тому [5]. Тому аналіз ефективності застосування відомих алгоритмів інтелектуального аналізу даних в умовах інтенсивного зростання обсягів даних та їх різноманіття є актуальним науково-практичним завданням інженерії даних і знань [6–8].

Існують апаратно-програмні комплекси, які надають вже сконфігуроване рішення для опрацювання даних великих обсягів: Aster MapReduce appliance [9], Oracle Big Data appliance [4], Greenplum appliance [10]. До засобів для опрацювання даних великих обсягів зараховують і апаратно-програмні комплекси на базі традиційних реляційних систем керування базами даних – Netezza, Teradata, Exadata. Основним недоліком всіх цих систем є їх висока вартість, що є визначальним фактором для використання їх компаніями середнього і малого масштабів.

Пошук асоціативних правил належить до інтелектуального аналізу. Задачу пошуку асоціативних правил розділяють на дві підзадачі [11, 12]:

1. Пошук усіх наборів елементів, які відповідають порогу мінімальної підтримки. Такі набори елементів називаються такими, що часто використовуються.

2. Генерація правил з наборів елементів, знайдених згідно з достовірністю, що задовольняє поріг мінімальної достовірності.

Першим алгоритмом для пошуку асоціативних правил був AIS, його розробили у 1993 р. працівники центру досліджень IBM Almaden. У ньому кандидати множини наборів створюються і розраховуються "на льоту", під час сканування бази даних. Нові набори кандидатів генеруються, розширяючи ці великі набори, що зустрічаються з іншими предметами в цій транзакції.

Пік досліджень у цій сфері випав на середину 90-х років XX століття і з того часу постійно з'являються нові алгоритми [13–15]. Алгоритм SETM [15] створює кандидати наборів "на льоту", скануючи бази даних, але їх розраховують наприкінці проходу. Нові набори кандидатів генеруються так само, як і в алгоритмі AIS. Недоліком є те, що для кожного кандидата занадто багато записів, що за кількістю дорівнює значенню підтримки.

Недоліки алгоритмів AIS і SETM полягають у зайвій генерації та підрахунку занадто великої кількості кандидатів, які в результаті не є такими, що часто зустрічаються. Алгоритм Apriori враховував ці недоліки. До поширених алгоритмів пошуку асоціативних правил належать також Eclat, FP-growth, OPUS, серед основних недоліків яких – ускладнена практична реалізація [16,17].

Залежно від розміру найдовшого набору елементів, які часто трапляються, алгоритм Apriori сканує базу даних певну кількість разів. Різновиди алгоритму Apriori, які є його оптимізацією, запропоновано для зменшення кількості сканувань бази даних і/або кількості наборів-кандидатів.

Враховуючи переваги та недоліки вказаних вище алгоритмів та особливості їх реалізації, для аналізу вибрано алгоритм Apriori на великих обсягах неструктурованих та структурованих даних.

Мета досліджень

Метою дослідження є оцінювання результатів роботи алгоритму Apriori для великих обсягів неструктурованої та структурованої інформації відповідно. Очевидно, що опрацювання структурованої інформації відбувається простіше, ніж неструктурованої. Тому необхідно виявити,

наскільки якість та продуктивність роботи алгоритму для неструктурованих даних великих обсягів відрізнятиметься від варіанта роботи зі структурованими даними великих обсягів. Якість роботи алгоритму визначатиметься кількістю згенерованих правил для різних наборів коефіцієнтів підтримки та достовірності. Продуктивність визначатиметься часом роботи за необхідних обчислювальних ресурсів системи.

Отримані результати можна використовувати для обґрунтування оптимального вибору методів і технологій реалізації систем, призначених для інтелектуального аналізу даних.

Для обчислювальних експериментів необхідно було реалізувати прототип програмної системи рекомендаційного характеру для сфери торгівлі, в основі якої – реалізація алгоритму Apriori.

Алгоритм Apriori для пошуку асоціативних правил

За допомогою алгоритмів пошуку асоціативних правил можна отримати всі можливі правила виду "З А випливає В" з різними значеннями підтримки та достовірності. Підтримка (support) – скільки разів у всьому масиві використано елементи даних, що складаються з X та Y, і достовірність (confidence) – який відсоток від всіх одиниць, що містять X, містить також і Y. Однак здебільшого кількість правил необхідно обмежувати заздалегідь встановленими мінімальними і максимальними значеннями підтримки та достовірності. Основним алгоритмом, який застосовується для отримання асоціативних правил, є алгоритм Apriori, перевагою якого є властивість масштабованості. Його автор – співробітник Microsoft Research Rakesh Agrawal. Алгоритм Apriori призначений для пошуку всіх множин ознак, що часто повторюються. Алгоритм є поетапним, використовує стратегію пошуку в ширину.

Алгоритм Apriori складається з таких етапів [13–17]:

1. ФОРМАЛІЗАЦІЯ ДАНИХ.

Кожна транзакція (набір елементів) перетворюється з вигляду T={t1,t2,…tn} на двійкову послідовність, довжина якої дорівнює кількості елементів усіх транзакцій, причому на місці елемента, який наявний у транзакції, є 1, а на місці елемента, який відсутній, – 0.

2. ПОШУК ОДНОЕЛЕМЕНТНИХ НАБОРІВ, ЩО ЧАСТО ВИКОРИСТОВУЮТЬСЯ.

Порівняння кожного елемента з кожною транзакцією для визначення кількості входжень.

3. ПОШУК k+1-ЕЛЕМЕНТНИХ НАБОРІВ, ЩО ЧАСТО ТРАПЛЯЮТЬСЯ.

Використовується властивість антимонотонності : k+1-елементний набір може траплятися часто лише тоді, коли часто зустрічаються усі k-елементні набори, що до нього входять.

4. ПОБУДОВА АСОЦІАТИВНИХ ПРАВИЛ.

Складність цього алгоритму залежить від мінімального значення підтримки. Припустимо, що кількість вхідних транзакцій N, а кількість унікальних елементів R. Складність для генерації набору розміру i – $O(R^i)$, де i – час для обчислення підтримки для кожного набору може бути O(N), у разі використання HashMap.

Для проведення обчислювальних експериментів реалізовано систему засобами екосистеми Hadoop [16], яка дасть змогу знаходити закономірності між наборами продуктів у вигляді асоціативних правил. Порівняння здійснюється із реалізацією того ж алгоритму мовою Java на структурованій базі даних, що розміщена на MS SQL Server 2012.

Програмна система для опрацювання даних

Для розподіленої обробки великих обсягів даних використано платформу Apache Hadoop. Hadoop вважається де-факто галузевим стандартом для управління великими даними. Це проект з відкритим кодом, що управляється Apache Software Foundation. Технічно Hadoop складається з розподіленої файлової системи HDFS, основним завданням якої є зберігання даних, та системи MapReduce, яка призначена для обчислень і оброблення даних на кластері. Програмна модель MapReduce запозичена з функціонального програмування, хоча в реалізації Hadoop і має деякі семантичні відмінності від прототипу в функціональних мовах [19–21]. Hadoop дає змогу швидко опрацьовувати величезні кількості структурованих і неструктурованих даних [22]. Особливості

використання технології Hadoop та RDBMS (систем керування реляційними базами даних) наведено у табл. 1.

Таблиця 1

Переваги і недоліки технологій RDBMS і Hadoop

RDBMS	Hadoop
Є реляційною системою управління базами даних	Є вузлом на основі файлової структури
ACID-властивості (Atomicity, Consistency, Isolation і Durability)	Не містить таких властивостей
Висока вартість обчислювальної потужності та зберігання	Низька вартість обчислювальної потужності та зберігання
Швидше порівнює однакові набори даних	Висока ефективність опрацювання паралельних задач
Низька швидкодія	Висока швидкодія
SQL -скрипти працюють швидко	HQL-скрипти працюють значно повільніше
Інтегровані	Денормалізовані
Не потрібні значні обчислювальні ресурси	Потрібні величезні обчислювальні ресурси
Комерційність	Open source проект

Створена програмна система формування рекомендацій для точки торгівлі має функціональні можливості: попереднє опрацювання даних; аналіз даних; розпаралелене опрацювання даних; пошук або створення асоціативних правил.

Проект складається з двох модулів: опрацювання неструктурованих даних і опрацювання структурованих даних. Розроблена система опрацьовує дані, які розміщені на Hadoop кластері, а Hadoop кластер розгортається на віртуальній машині Oracle VirtualBox (програмний продукт віртуалізації для операційних систем). Для організації кластера використано дистрибутив Cloudera. За збереження та організацію даних у кластері Hadoop відповідає розподілена файлова система Hadoop Distributed File System (HDFS), яка забезпечує високошвидкісний доступ до даних застосунку.

Для забезпечення максимальної простоти взаємодії користувача із системою написано веб-клієнт, який дозволяє здійснювати пошук за іменем товару, з урахуванням місця та частини асоціативного правила в індексі на пошуковому сервері Elasticsearch. Алгоритм пошуку асоціативних правил реалізовано мовою програмування Java у середовищі розробки NetBeans IDE. Для поставленої мети використано базу даних товарів, що містить інформацію про всі транзакції над певним набором продуктів. Неструктуровані дані завантажено з *.csv файлів, які зберігаються у файловій системі HDFS та обробляються обчислювальним фреймворком для розподілених задач MapReduce. Структуровані дані розміщено у базі даних “Products” MS SQL Server 2012. Алгоритм опрацьовує однаковий за змістом, але різний за структурою набір даних.

У результаті опрацювання створено два файли з асоціативними правилами, перший з яких містить усі правила, згенеровані на основі неструктурованих даних, другий, відповідно, структурованих. Спочатку кожна транзакція перетворюється з набору елементів на двійкову послідовність, що дорівнює довжини елементів усіх транзакцій.

Результатом оброблення неструктурованих даних алгоритмом Apriori є асоціативні правила, що створені на сервері Elasticsearch та відображаються як json файли. Приклад такого файла подано на рис. 1.

Створені файли містять значення підтримки та достовірності для кожного правила, “Head” і “Tail”, у цьому випадку X – це “голова”, а Y – “хвіст”.

Результат роботи веб-системи для структурованих даних наведено на рис. 2.

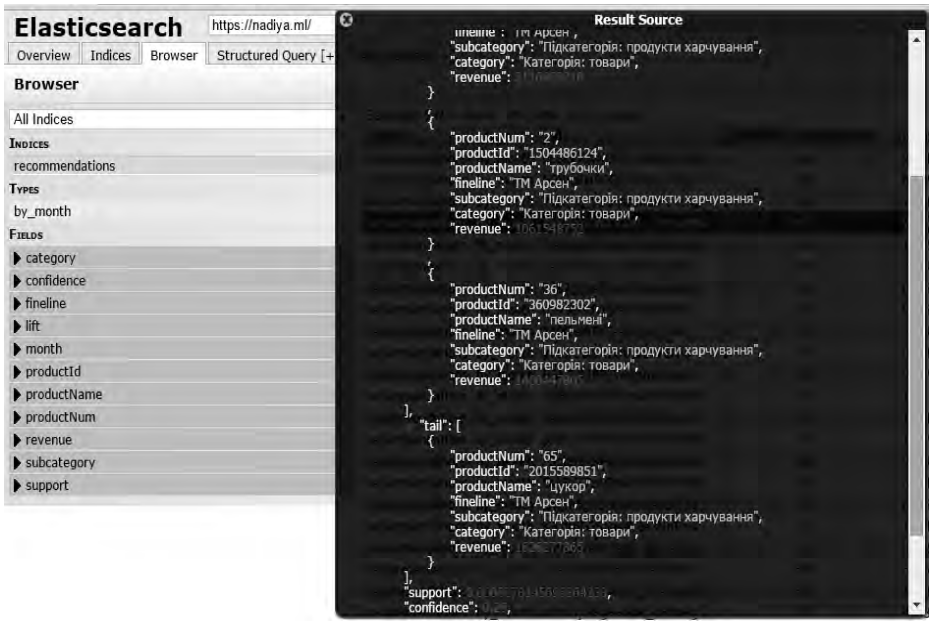

Рис. 1. Вигляд асоціативного правила на сервері Elasticsearch

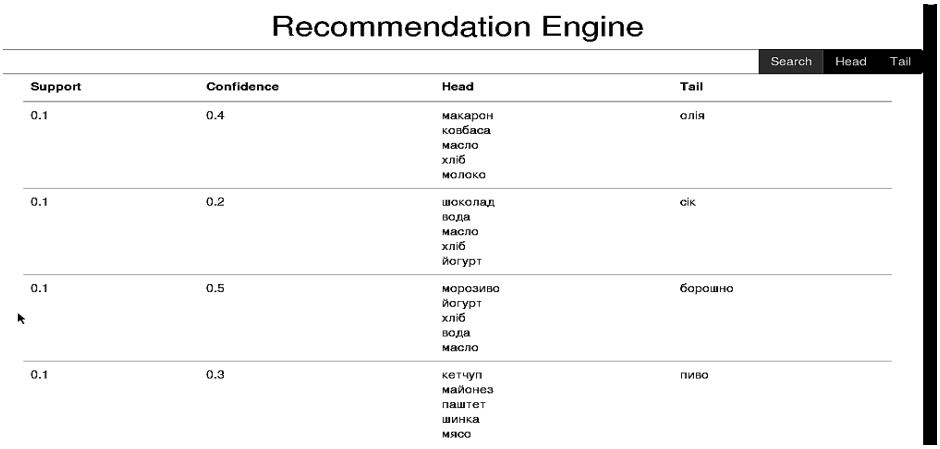

Рис. 2. Приклад роботи модуля опрацювання структурованих даних

Отримано інформацію про те, як саме набір одних продуктів може асоціюватися (бути пов'язаним) з іншими продуктами. Така інформація вказує, наприклад, як вигідно розміщувати товари на полицях для збільшення продажів; які рішення приймати стосовно замовлень певних товарів для збільшення доходів.

Результати обчислювальних експериментів

Для оцінювання результатів роботи програмної системи виконано процес на 450 Мб, 1 Гб та 2,5 Гб відповідно структурованих та неструктурованих даних. Дані містять інформацію про продукти торговельної мережі й відповідні транзакції над ними.

Система для опрацювання неструктурованих даних потребує мінімум 16 Гб оперативної пам'яті та 8 ядер процесора, тоді як структурованих – мінімум 2 Гб оперативної пам'яті та 2 ядра процесора [23]. Для коефіцієнтів підтримки та достовірності вибрано середні значення – 0,4 та 0,5 відповідно. Результати пошуку асоціативних правил для 350 назв продуктів та різної кількості транзакцій наведено на рис. 3.

Якщо вибрати коефіцієнти підтримки та достовірності, вищі від середнього, наприклад, 0.6 та 0.7 відповідно, то для структурованих даних знайдено лише три правила, а для неструктурованих – жодного. Це вказує на складність опрацювання неструктурованих даних і обмеження в застосуванні алгоритму Apriori. Для середніх коефіцієнтів підтримки та достовірності

алгоритм знаходить у структурованих даних на близько 20 % більше асоціативних правил, ніж у неструктурованих. У табл. 2 наведено результати порівняльного аналізу роботи алгоритму на обох типах даних різного обсягу.

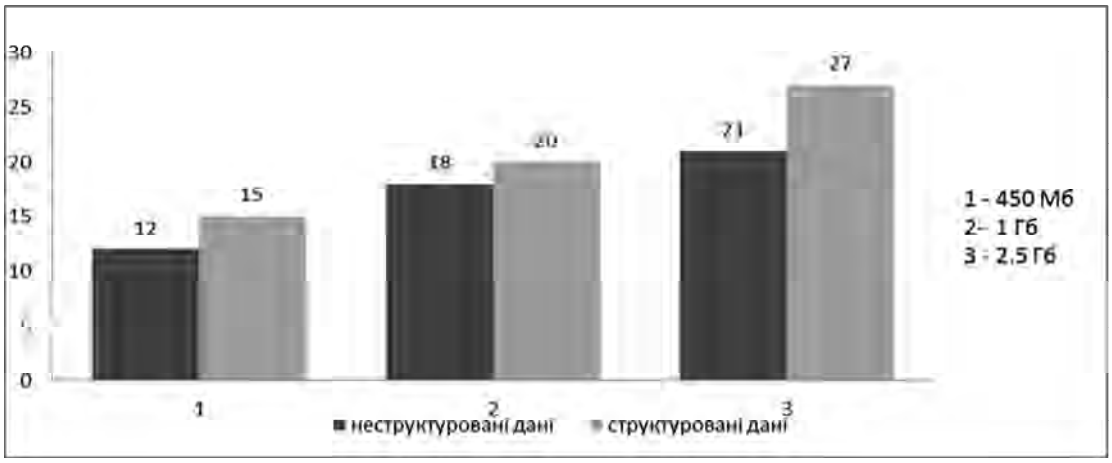

Рис. 3. Кількість знайдених правил для різних обсягів даних

Таблиця 2

Аналіз часових витрат роботи алгоритму Apriori

№ експерименту	Дані	Час, секунд	
		НЕСТРУКТУРОВАНІ ДАНІ	СТРУКТУРОВАНІ ДАНІ
1	490 Мб	187 с	47 с
2	1 Гб	528 с	306 с
3	2,5 Гб	737 с	656 с

Алгоритм Apriori працює на малих обсягах структурованих даних утричі швидше від аналогічного випадку для неструктурованих. Зі зростанням обсягів даних переваги у швидкодії для структурованих даних стають значно меншими. Для експерименту 2 час обчислень у випадку неструктурованих даних більший на 70 %, а для експерименту 3 – лише на 12 %.

Реалізована система дає змогу виявити закономірності в даних. Оцінювання отриманих закономірностей сприяє покращенню розуміння проблемної області й формуванню ефективніших розв'язків задач, що виникають у певній предметній області.

Висновки

Розглянуто актуальне завдання опрацювання алгоритмом Apriori великої кількості структурованих і неструктурованих даних для пошуку залежностей у них. Результати роботи алгоритму Apriori засвідчують, що він достатньо ефективний для опрацювання також і неструктурованої інформації.

1. Втрати у швидкодії для неструктурованої інформації порівняно зі структурованою інформацією незначні зі зростанням обсягів інформації, і зокрема, для випадку обсягу даних 2,5 Гб зростання часу роботи становить близько 12 %. А на порівняно невеликих обсягах даних для структурованих даних алгоритм працює утричі швидше порівняно із неструктурованими.

2. Кількість знайдених правил для неструктурованої інформації на близько 25 % менша, ніж для структурованої інформації.

3. Для коефіцієнтів підтримки та достовірності, вищих від середнього, для неструктурованих даних не знайдено жодного правила, а для структурованих результат – від 2 до 5 правил.

Отримані результати аналізу роботи алгоритму Apriori можна використовувати для обґрунтування оптимального вибору технологій реалізації даних і методів їх опрацювання.

1. Montgomery Karen. Big Data Now: 2014 Edition. O'Reilly Media. – Junuary, 2015. – 165 p. 2. Майер-Шенбергер Виктор, Кукьер Кеннет. Большие данные. Революция, которая изменит то, как мы живём, работаем и мыслим = Big Data. A Revolution That Will Transform How We Live, Work, and Think / пер. с англ. И. Гайдюк. – М.: Манн, Иванов, Фербер, 2014. – 240 с. 3. Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data. John Wiley & Sons. 2014-12-19. 300p. 4. Big Data Applience [Електронний ресурс] // Oracle Big Data: сайт. – Режим доступу https://www.oracle.com/engineered-systems/big-data-appliance/index.html. 5. Almasi, G.S. and A. Gottlieb (2009). Highly Parallel Computing. Benjamin // Cummings publishers, Redwood City, CA. – 235 с. 6. Шаховська Н. Б. Організація великих даних у розподіленому середовищі / Н. Б. Шаховська, Ю. Я. Болюбаш, О. М. Верес// Наукові праці ДонНТУ. Серія: обчислювальна техніка та автоматизація. – 2014. – № 2(27). – С. 147–155. 7. Павич Н. Я. Оцінювання ефективності опрацювання даних великих обсягів технологіями Spark та Hive / Н. Я. Павич, О. П. Крохмальна // Вісник Нац. ун-ту "Львів. політехніка" "Комп'ютерні системи та мережі". – 2015. – № 830. – С. 128–135. 8. Сєдушев О. Ю. Методи видобування даних з баз нечітких знань / О. Ю. Сєдушев, Є. В. Буров // Вісник Нац. ун-ту "Львів. політехніка" "Інформаційні системи та мережі". – 2014. – № 783. – С. 193–203. 9. Mapreduce Appliance. [Електронний ресурс] // MapReduce: сайт. – Режим доступу http://www.teradata.com/products/Aster_MapReduce_Appliance. 10. GreenPlum. [Електронний ресурс]//: сайт. – Режим доступу http://www.emc.com/campaign/global/greenplumdca/index.htm. 11. Zhu Yixia, Yao Liwen, Huang Shuiyuan, Huang Longjun. A association rules mining algorithm based on matrix and trees[J]. Computer science. 2006, 33(7):196-198. 12. Tong Qiang, Zhou Yuanchun, Wu Kaichao, Yan Baoping. A quantitative association rules mining algorithm[J]. Computer engineering. 2007. 13. Agrawal R., Imielinski T., Swami A. Mining association rules between sets of items in large database, In Proc. of the 1993 ACM-SIGMOD Int'l Conf. on Management of Data, 1993: 207-216. 14. Agrawal R. and Srikant, R. Fast algorithms for mining association rules. In Proc.20th Int. Conf. Very Large Data Bases, Santiago, Chile, 1994. 487–499. 15. Purdom P. W., Guch D. V., Groth D. P. Avarage case performance of the apriori algorithm – SIAM Journal on Computing, 33(5): 1223–1260, 2004. 16. Mohammed J. Zaki. Scalable algorithms for association mining – IEEE Transactions on Knowledge and Data Engineering, 12(3):373–390, 2000. 17. Brin S., Rajeev Motwani, Ullman J., Tsur S. Dynamic itemset couting and implication rules for market basket data// Proc. ACM SIGMOD Intern. Conference on Management of Data, 255–264 p., USA, 1997. 18. Apache Hadoop. [Електронний ресурс]// Big Data: сайт. – Режим доступу https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html 19. Harris, Dereck Intel jettisons its Hadoop distro and puts millions behind Cloudera (27 March 2014). 20. Уайт, Том Hadoop. Подробное руководство= Hadoop: The Definitive Guide. – СПб., 2013. – 672 с. 21. Hadoop File System. [Електронний ресурс]// hadoop-distributed-file-system: сайт. – Режим доступу https://www. safaribooksonline.com/blog/2013/02/13/the-hadoop-distributed-file-system. 22. White T. Hadoop: The Definitive Guide, 4th Edition. O'Reilly Media. – March, 2015 – 756 p. 23. Нечипір Н. І. Опрацювання великих обсягів неструктурованих та структурованих даних алгоритмом Apriori / Н. І. Нечипір, Є. В. Левус // Математичне та програмне забезпечення інтелектуальних систем: матер. XIII Міжнар. наук.-практ. конф. – Дніпропетровськ: Вид-во Дніпропетр. Нац. ун-ту ім. Олеся Гончара, 2015. – С. 34–36.