

Ключові слова: глибоке машинне навчання; сверточное нейронна мережа; методи розпізнавання образів; візуалізація зображень; рентгенограма; алгоритм навчання.

Грицай А.С., Левицкая Т.А., Бухлал Н.А.

СИСТЕМАТИЗАЦИЯ ПОИСКА АНОМАЛИЙ В РЕНТГЕНОВСКИХ СНИМКАХ

Ускорить процесс диагностики аномалий, снизить количество ошибок и повторных опросов пользователей позволяют алгоритмы патолого-структурных структур, использующих нейронные сети. В статье рассматриваются способы машинной систематизации и распознавания рентгеновских снимков (СХР), а также проблемы совершенствования искусственных нейронных сетей, которые применяются с целью усиления свойств систематизации рентгенологических синдромов. В виду того, что для выявления заболевания вполне достаточно реализовать определенный алгоритм, нейронные сети являются идеальными для распознавания болезней с применением сканирования. Проанализировав исследований и публикаций по этой теме были сформированные основные задачи для моделирования системы. Также были классифицированы архитектуры нейронных сетей, с указанием их недостатков и преимуществ. Выявлено, что современные методы обнаружения аномалий в СХР имеют некоторые сложности, такие как недостающие число тренировочных сведений, типизацией изображений и предварительной сегментацией тренировочного набора. Детерминированные конкретные способы решения проблем нейронных сетей при анализе данных. Для реализации предложено применить методы глубокого машинного обучения, на основе сверточных нейронных сетей применяя предварительную сегментацию тренировочной выборки с обратным распространением ошибки и градиентного спуска с и применения трансферного обучения для систематизации заболеваний на медицинских изображениях. Для решения поставленных задач выбраны инновационные IT-технологии. В результате была реализована определенная архитектура интеллектуальной системы, позволяющая обнаруживать аномалии на рентгенограммах, которые позволяют нам создавать эффективные структуры нейронных сетей и повышать точность распознавания патологических структур на рентгенограммах.

Ключевые слова: глубокое машинное обучение; сверточная нейронная сеть; методы распознавания образов; визуализация изображений; рентгенограмма; алгоритм обучения.

Рецензент доц., к.т.н. Сорочан О.М.

Статья поступила 26.11.2019

УДК 004.032.24

Половинка О.Л., Нікулін Д.В., Дмитрієва О.А.

ДОСЛІДЖЕННЯ ШВИДКОДІЇ РОБОТИ АПРІОРНИХ АЛГОРИТМІВ НА ДАНИХ ВЕЛИКИХ ОБСЯГІВ

В статті обґрунтовано використання алгоритмів пошуку асоціативних правил для роботи з неструктурованими даними великих обсягів. Зазначено, що найбільш популярним алгоритмом пошуку асоціацій є алгоритм Apriori. Наведено результати аналізу роботи лінійних алгоритмів пошуку асоціацій з неструктурованими даними великих обсягів. Розглянуто найбільш відомі модифікації апріорних алгоритмів пошуку асоціативних правил,

отримано їх порівняльні характеристики. Вказані сильні та слабкі моменти кожного алгоритму. Наведено результати дослідження швидкодії роботи лінійних алгоритмів AIS, Apriori та AprioriTID на даних різних обсягів з використанням комп'ютерів зі стандартним обсягом оперативної пам'яті. Виявлено, що при обробці великих даних в заявлених умовах, робота алгоритмів Apriori та AprioriTID стає неможливою через нестачу оперативної пам'яті. Для вирішення означеної проблеми проведено подальший аналіз структурних особливостей алгоритмів з апіорної групи. За його результатами для обробки даних великих обсягів на комп'ютерах зі стандартним обсягом пам'яті, обрано алгоритм AIS, який є більш повільним, але не вимагає завантаження транзакцій в оперативну пам'ять комп'ютера, а, отже, не залежить від її обсягу. Наведені результати роботи алгоритму AIS на різних обсягах даних, на основі яких зроблено висновок, що запропонована реалізація не вимагає великого обсягу оперативної пам'яті, що є перевагою, однак є досить повільною і на великих обсягах даних потребує для виконання неприйнятний для практики час. Для вирішення цієї проблеми запропоновано звернутися до методів побудови паралельних або розподілених структур. Зроблено висновки про необхідність застосування для розв'язання сформульованої задачі паралельних алгоритмів.

Ключові слова: швидкодія, неструктуровані дані, пошук асоціативних правил, Apriori, AIS, AprioriTID, великі дані.

Постановка проблеми.

Останнім часом спостерігається постійне збільшення цифрової інформації, ефективна обробка якої дозволяє значно збільшити прибуток підприємства. Тому затребуваними є методи розв'язання задач, пов'язаних з необхідністю обробки великих масивів даних та пошуком прихованих закономірностей.

Для аналізу великих даних широко застосовуються нейронні мережі, дерева рішень, нечіткі моделі, методи регресійного і кластерного аналізу.

Однак ці методи, як правило, використовуються для обробки структурованих даних, представлених у вигляді масивів, що містять значення ознак і вихідних параметрів примірників вибірки. У той же час, обробка неструктурованих даних, в яких кожна одиниця зберігання не може бути представлена кінцевим числом ознак (атрибутів), подібними методами неможлива [1].

Для обробки великих масивів неструктурованих даних і вирішення зазначених задач доцільно використовувати методи пошуку асоціативних правил. Такі методи дозволяють виявляти в наявних даних нові закономірності виду "якщо - то". На основі виявлених залежностей можна синтезувати бази правил, зрозумілі експертам в прикладних областях.

На даний час в науковій літературі наведено значну кількість алгоритмів пошуку асоціативних правил та їх модифікацій. Але кожен з них має свої особливості й застосовується для розв'язання певного класу задач. Універсального алгоритму не існує.

Тому аналіз існуючих розробок та їх модифікація з метою обрання ефективного алгоритму пошуку асоціативних правил в умовах забезпечення заданого рівня підтримки й достовірності результату, необхідних для обробки великих даних без прив'язки до сфери застосування цих даних, є актуальною задачею, що має й практичне, й наукове значення.

Аналіз останніх досліджень і публікацій.

Питаннями збору, зберігання і обробки даних великих обсягів займається все більше вчених і дослідників. Алгоритми пошуку асоціативних правил використовуються у багатьох сферах діяльності. Спочатку він застосовувався для знаходження типових шаблонів покупок, що здійснюються в супермаркетах, але зараз асоціативні правила використовуються для відновлення зображень [2], виявлення взаємозв'язку місць проживання біологічних видів, обробки фольклорних ресурсів, аналізу діяльності громадських організацій, в медичній

діагностиці [3] і, навіть, в навчанні [4]. В роботах [5, 6] запропоновано використовувати методи пошуку асоціативних правил для обробки великих масивів неструктурованих даних, надано опис деяких алгоритмів.

Мета (завдання) дослідження.

Мета роботи полягає у дослідженні ефективності використання лінійних алгоритмів апріорної групи при обробці великих обсягів даних та отриманні показників швидкодії обробки існуючими програмними засобами.

Основний матеріал дослідження.

Базові алгоритми пошуку асоціацій та основні принципи їх формування описані, наприклад, в [6]. Зазначено, що основними оціночними характеристиками алгоритму є поняття підтримки та достовірності.

При цьому процес синтезу асоціативних правил розбивається на два етапи. Перший - генерування наборів X з рівнем підтримки, не нижче заданого експертом порогового значення $\text{minsupport}(X)$, в результаті чого формуються набори, які зустрічаються часто. Другий - генерування всіх правил $X \rightarrow Y$ з рівнем достовірності, не нижче заданого експертом порогового значення $\text{minconfidence}(X \rightarrow Y)$.

Найбільш поширеним алгоритмом пошуку асоціацій визначається алгоритм Apriori, що є модифікацією AIS, запропонованого у 1993 році Р. Агравалом [7]. Відомо ряд модифікацій алгоритмів апріорної групи, які дозволяють скоротити час пошуку асоціативних правил. Основна відмінність існуючих апріорних алгоритмів полягає в різних стратегіях генерації кандидатів.

У табл. 1 наведено результати порівняння найвідоміших апріорних алгоритмів.

Таблиця 1 – Порівняння апріорних алгоритмів.

Алгоритм	Формування кандидатів	Підрахунок	Недоліки
AIS, SETM	під час сканування бази даних	під час сканування бази даних	зайве генерування і підрахунок занадто багатьох кандидатів
Apriori	На етапі candidate generation	На етапі candidate counting	Багаторазове сканування бази, ефективний на ранніх проходах
Apriori TID	На етапі candidate generation	За допомогою кодування кандидатів	Надмірне час роботи на ранніх проходах
Apriori Hybrid	На етапі candidate generation	Спочатку на етапі candidate counting далі за допомогою кодування кандидатів	вимагає залучення додаткових ресурсів

Розглянемо особливості роботи лінійних апріорних алгоритмів на великих масивах даних. Для цього застосуємо реалізації алгоритмів Apriori і AprioriTID. Псевдокоди розглянутих алгоритмів наведено в [8].

Експерименти проводились на даних результатів транзакцій, які були представлені у вигляді таблиці зі слабо структурованими даними. Особливістю таких даних є їх універсальність, тобто те, що аналогічні таблиці можна отримати з багатьох сфер діяльності. Основним є факт проведення транзакції й невизначена кількість параметрів у рядку. Ці дані можуть бути отримані не тільки з торговельної мережі (продажів фармацевтичної кампанії), а й при аналізі запитів користувачів до сайту, процесу генерації випадкових сигналів, роботі кол-центру й інше.

На першому етапі дослідження виконувалось перетворення початкової таблиці для формування структурованого файлу даних, придатного для обробки апріорними алгоритмами. Фрагмент підготовленої для обробки бази даних представлений у таблиці 2.

Наступним кроком було дослідження алгоритму *AprioriTid*. Особливістю алгоритму є те, що база даних не використовується для підрахунку підтримки кандидатів набору товарів після першого проходу. З цією метою використовується кодування кандидатів, виконане на попередніх проходах.

Таблиця 2 – Фрагмент готової до обробки бази даних.

Номер транзакції	Масив товарів
12921	44693,44775,45471
12922	45897
12923	45379
12924	44903,4591
12925	45410
12926	45426
12927	48169
12928	44199,46274,47473
12929	45256
12930	48082
12931	48001
12932	44108,44763,47976
12933	47578
12934	20716,34995,44649,45179,48169,48958

Таким чином, робота алгоритму значно прискорюється завдяки значній економії ресурсів. Результати роботи алгоритму *AprioriTid* наведено в табл. 3, та на рис. 1.

Таблиця 3 – Результати роботи алгоритму *AprioriTid*.

№	Кількість елементів	Об'єм бази даних, Мб	Час роботи алгоритму, години
1	100000	1.1 Мб	00:00:01
2	500000	5.6 Мб	00:00:05
3	1000000	11.1 Мб	00:00:28
4	2000000	25.7 Мб	00:01:32
5	4000000	52 Мб	00:03:22
6	8000000	101 Мб	00:05:04
7	14000000	236 Мб	00:10:19
8	20000000	258 Мб	00:13:20
9	26000000	320 Мб	MemoryError

Як видно з табл. 3, алгоритм характеризується прийнятним часом реалізації, навіть для великої кількості транзакцій. Однак при роботі з даними значних обсягів (більше, ніж 300 Мб, тобто 20000000 транзакцій) виявилася нестача оперативної пам'яті, що унеможливило подальшу обробку.

Третя група експериментів проводилася з застосуванням лінійної реалізації класичного алгоритму *Apriori*. При цьому нестача оперативної пам'яті проявилася на значно

менших, ніж у ArpioriTid обсягах бази. Така ситуація пов'язана з тим, що обидва алгоритми завантажують весь об'єм транзакцій до оперативної пам'яті комп'ютера.

Слід зазначити, що оригінальна версія алгоритму Arpiori (без жодних модифікацій) не зобов'язує розробника завантажувати усю множину транзакцій в пам'ять, таке рішення дозволяє значно пришвидшити роботу алгоритму та використовується у більшості існуючих розробок. Проведені експерименти показали, що результати роботи алгоритмів на великих обсягах даних (які перевищують кілька мільйонів транзакцій) є неприйнятними. Крім того, при малих показниках підтримки, та за наявності великої кількості унікальних елементів в транзакціях, кількість комбінацій переважно на другій ітерації є занадто великою, щоб вміститися в оперативну пам'ять звичайних комп'ютерів.

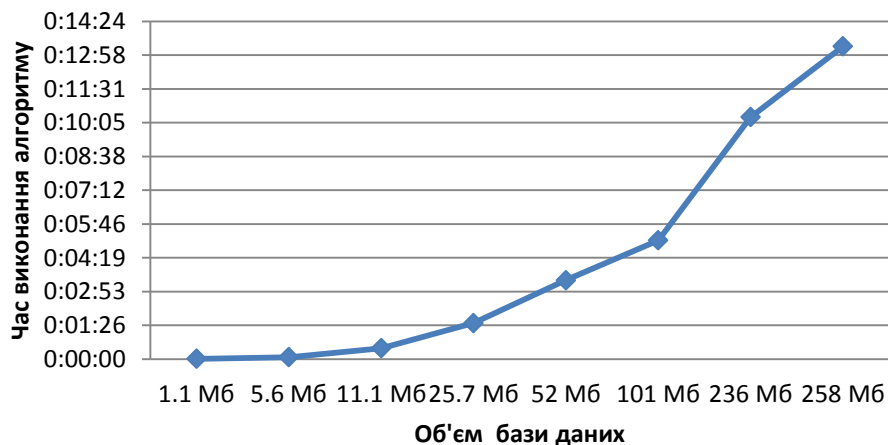


Рисунок 1 – Робота алгоритму ArpioriTID

В результаті аналізу було встановлено, що алгоритм AIS не вимагає завантаження транзакцій в оперативну пам'ять комп'ютера, а, отже, не залежить від її обсягу. Особливістю алгоритму є те, що він багаторазово проходить по базі даних. Результуючий набір складається з унікальних наборів елементів. Він змінюється (розширюється) під час кожного проходу.

При цьому на кожному проході вимірюється підтримка знайдених наборів елементів. Ці набори, які носять назву набори-кандидати, є похідними від кортежів в базі даних і наборів елементів, що зафіксовано в результуючому наборі.

Для кожного набору елементів створюється лічильник, що зберігає кількість транзакцій, в яких з'явився відповідний набір елементів. Лічильник відновлюється при створенні нового набору.

Спочатку граничний набір складається тільки з одного елемента, що є порожнім. В кінці проходу підтримка для набору елементів-кандидатів порівнюється із заданою мінімальною підтримкою (*minsupport*), щоб визначити, чи є він найбільшим набором елементів. У той же час визначається, чи повинен цей набір елементів бути доданим до граничного набору для наступного проходу. Алгоритм завершується, коли граничний набір стає порожнім. Лічильник підтримки для набору елементів зберігається, коли набір елементів додається в більший або приграничний набір [8].

Результати роботи алгоритму AIS наведено у табл. 4.

Запропонована реалізація не вимагає великого обсягу оперативної пам'яті, що є перевагою, однак є досить повільною і на великих обсягах даних буде потребувати для виконання неприйнятний для практики час.

Таблиця 4 – Результати роботи алгоритму AIS.

№	Кількість елементів	Об'єм бази даних, Мб	Час роботи алгоритму, години
1	100000	1.1 Мб	00:00:56
2	500000	5.6 Мб	00:05:12
3	1000000	11.1 Мб	00:10:20
4	2000000	25.7 Мб	00:24:09
5	4500000	52 Мб	00:41:55
6	8000000	101 Мб	01:10:23

Таким чином, всі запропоновані алгоритми, в умовах роботи з даними великих обсягів, є погано масштабованими, а також малоефективними. Для вирішення цієї проблеми пропонується використання паралельних і розподілених алгоритмів для виконання великомасштабних обчислень.

ВИСНОВКИ

У статті зазначено, що найбільш популярним алгоритмом пошуку асоціацій є алгоритм Apriori. Розглянуто основні модифікації апріорних алгоритмів пошуку асоціативних правил. Наведено їх порівняльну характеристику, розглянуто сильні сторони кожного алгоритму, а також їх недоліки.

Проведено дослідження ефективності роботи лінійних алгоритмів Apriori і AprioriTID на даних різних обсягів з використанням комп'ютерів зі стандартним об'ємом пам'яті. Виявлено, що при обробці великих даних в заявлених умовах, робота алгоритмів Apriori та AprioriTID стає неможливою через нестачу оперативної пам'яті.

Для вирішення означеної проблеми проведено подальший аналіз структурних особливостей алгоритмів з апріорної групи. За його результатами для обробки даних великих обсягів на комп'ютерах з невеликим обсягом пам'яті, обрано алгоритм AIS, який є більш повільним, але не вимагає завантаження транзакцій в оперативну пам'ять комп'ютера, а, отже, не залежить від її обсягу.

Для вирішення проблеми швидкодії алгоритму, у подальших дослідженнях пропонується розробка паралельної модифікації апріорних алгоритмів, спрямованої на можливість розподіленої обробки даних.

Перелік використаних джерел:

1. Зайко Т.А. Ассоциативные правила в интеллектуальном анализе данных / Т.А. Зайко, А.А. Олейник, С.А. Субботин // Вісник НТУ «ХПІ». Серія : Інформатика і моделювання. – Харьков, 2013. – № 39 (1012). – С. 82-96.
2. Белим С.В. Использование ассоциативных правил для восстановления зашумленных изображений / С.В. Белим, А.О. Майоров-Зильбернагель, С.А. Селиверстов // Вестник ОмГУ. Серія : Компьютерные и информационные науки. – Омск, 2013. – № 4 (70). – С. 197-200.
3. Биллиг В.А. Построение ассоциативных правил в задаче медицинской диагностики / В.А. Биллиг, О.В. Иванова, Н.А. // Программные продукты и системы. – 2016. – №2 (114). – Режим доступа: <https://cyberleninka.ru/article/n/postroenie-assotsiativnyh-pravil-v-zadache-meditsinskoj-diagnostiki>.
4. Маслова Н.О. Застосування методів пошуку асоціацій при створенні тестів з інформаційної безпеки / Н.О. Маслова, О.Л. Половинка // Збірник наукових праць

Донецького національного технічного університету. Серія: Інформатика кібернетика та обчислювальна техніка. – Покровськ, 2019. – №1(29). – С. 47-53.

5. Интеллектуальные информационные технологии проектирования автоматизированных систем диагностирования и распознавания образов: монография / С.А. Субботин, Ан.А. Олейник, Е.А. Гофман, С.А. Зайцев, Ал.А. Олейник. – Харьков: Компания Смит, 2012. – 317 с.

6. *Adamo J.-M.* Data mining for association rules and sequential patterns: sequential and parallel algorithms. – New York: Springer-Verlag, 2001. – 259 p.

7. *Agrawal Rakesh.* Mining association rules between sets of items in large databases / R. Agrawal, T. Imielinski, and A. Swami // In Proc. of the ACM SIGMOD Conference on Management of Data. – 1993. Режим доступа: <https://rakesh.agrawal-family.com/papers/sigmod93assoc.pdf>.

8. *Agrawal Rakesh.* Fast algorithms for mining association rules / *Agrawal Rakesh and Ramakrishnan Srikant* // Proc. 20th int. conf. very large data bases, VLDB. – 1994. – Vol. 1215. – P. 487-499.

Половинка О.Л., Никулин Д.В., Дмитриева О.А.

ИССЛЕДОВАНИЕ БЫСТРОДЕЙСТВИЯ РАБОТЫ АПРИОРНЫХ АЛГОРИТМОВ НА ДАННЫХ БОЛЬШИХ ОБЪЕМОМ

В статье обосновано использование алгоритмов поиска ассоциативных правил для работы с неструктурированными данными больших объемов. Отмечено, что наиболее популярным алгоритмом поиска ассоциаций является алгоритм Apriori. Приведены результаты анализа работы линейных алгоритмов поиска ассоциаций с неструктурированными данными больших объемов. Рассмотрены наиболее известные модификации априорных алгоритмов поиска ассоциативных правил, приведены их сравнительные характеристики. Указаны сильные и слабые стороны каждого алгоритма. Приведены результаты исследования быстродействия работы линейных алгоритмов AIS, Apriori и AprioriTID на данных различных объемов с использованием компьютеров со стандартным объемом оперативной памяти. Обнаружено, что при обработке больших данных в заявленных условиях, работа алгоритмов Apriori и AprioriTID становится невозможной из-за недостатка оперативной памяти. Для решения этой проблемы проведения дальнейший анализ структурных особенностей алгоритмов априорной группы. По его результатам для обработки данных больших объемов на компьютерах со стандартным объемом памяти, выбран алгоритм AIS, который является более медленным, но не требует загрузки транзакций в оперативную память компьютера, а, следовательно, не зависит от ее объема. Приведены результаты работы алгоритма AIS на разных объемах данных, на основе которых сделан вывод, что предложенная реализация не требует большого объема оперативной памяти, что является преимуществом, однако является достаточно медленной и на больших объемах данных требует для выполнения неприемлемое для практики время. Для решения этой проблемы предложено обратиться к методам построения параллельных или распределенных структур. Сделаны выводы о необходимости применения параллельных алгоритмов для решения сформулированной задачи.

Ключевые слова: *быстродействие, неструктурированные данные, поиск ассоциативных правил, Apriori, AIS, AprioriTID, большие данные.*

Olga Polovynka, Danil Nikulin, Olga Dmitrieva

RESEARCH OF THE PERFORMANCE OF THE APRIORI GROUP ALGORITHMS ON BIG DATA

The using of associative rules search algorithms for working with unstructured data of large volumes are substantiated in the article. It is noted that the most popular associative rules search algorithms is the Apriori algorithm. The results of the analysis of the operation of linear algorithms for associative rules search with unstructured data of large volumes are presented. The most famous modifications of Apriori algorithms for finding associative rules are considered. Comparative characteristics of the considered algorithms are given, the strengths and weaknesses of each algorithm are indicated. The results of the research of the efficiency of linear algorithms AIS, Apriori and AprioriTID on the data of different volumes using computers with standard RAM volume are presented. It is found that when processing big data in the stated conditions, the operation of the Apriori and AprioriTID algorithms becomes impossible due to lack of RAM. To solve this problem, a further analysis of the structural features of apriori group algorithms is carried out. According to its results, the AIS algorithm was selected for processing Big data on computers with a standard amount of memory. This algorithm work slower, but does not require downloading transactions to the computer's RAM, and therefore does not depend on its size. Based on the results of using the AIS algorithm for different amounts of data, it was concluded that the proposed implementation does not require a large amount of RAM, which is an advantage, but with large amounts of data it is quite slow and the time for implementation is unacceptable for practice. To solve this problem, it is proposed to turn to methods for constructing parallel or distributed structures.

Keywords: performance, unstructured data, associative rules search algorithms, Apriori, AIS, AprioriTID, Big data.

*Стаття надійшла _____
Рецензент: д.т.н. Годлевський М.Д.*

УДК 004.04

Сирмаміїх І.В., Таранов І.Р., Романюта М.О.

РОЗРОБКА СИСТЕМИ АВТОМАТИЗОВАНОГО ОБЛІКУ ДОСЯГНЕНЬ СТУДЕНТІВ НА NODE.JS З ІНТЕГРАЦІЄЮ FIREBASE

Впровадження інформаційних технологій відбувається практично у всіх галузях, але найбільш гостро вирішення цього питання вимагає система освіти, зокрема організація управління на факультетах закладів вищої освіти.

Стаття присвячена розробці програмного продукту для автоматизованого контролю досягнень студентів зі статистикою, звітністю, зручним і швидким форматом взаємодії, який замінить паперовий ручний формат обробки документів. Мета при розв'язуванні задачі стояла - використовувати прогресивні технології та інструментарій. Серверна платформа сервісу - Node.js на мові TypeScript з сервісами Firebase від Google. Клієнтська частина створена з використанням Vue.js (frontend фреймворк для створення Single page application (SPA) - односторінкових додатків), Nuxt.js (фреймворк для Server-Side Rendering (SSR) - серверного рендерингу), Typescript (строго типізована мова програмування веб-додатків). Взаємодія з сервером виконується за допомогою Representational State Transfer API (REST API). Основна реалізація - у веб-додаток. Уся необхідна звітність по