УДК 004.9:530.1

Н.Б. Шаховська, Р.М. Камінський, Є.О. Засоба
Національний університет «Львівська політехніка»,
м. Львів, Україна, Natalia.b.shakhovska@lpnu.ua

МЕТОД ПОШУКУ АСОЦІАТИВНИХ ЗАЛЕЖНОСТЕЙ У ВЕЛИКИХ ДАНИХ

В роботі запропоновано метод аналізу Великих даних в умовах наявності різних джерел даних та різних методів опрацювання цих даних. Описано складові проблеми опрацювання різнотипних даних. Уведено поняття асоціативної залежності, розроблено метод пошуку залежностей, визначено ефективність та можливості його розпаралелення. Здійснено порівняльний аналіз ефективності методів пошуку асоціативних правил.

ВЕЛИКІ ДАНІ, АСОЦІАТИВНЕ ПРАВИЛО, ЗАЛЕЖНІСТЬ ДАНИХ, СКЛАДНІСТЬ АЛГОРИТМУ, ПАРАЛЕЛЬНЕ ОПРАЦЮВАННЯ

Шаховська Н.Б., Каминский Р.М., Засоба Е.А. Метод оиса ассоциативных зависимостей в Больших данных. В работе предложен метод анализа Больших данных в условиях наличия различных источников данных и различных методов обработки этих данных. Описаны составляющие проблемы обработки разнотипных данных. Введено понятие ассоциативной зависимости, разработан метод поиска зависимостей, определена эффективность и возможности его распараллеливания. Осуществлен сравнительный анализ эффективности методов поиска ассоциативных правил.

БОЛЬШИЕ ДАННЫЕ, АССОЦИАТИВНЫХ ПРАВИЛ, ЗАВИСИМОСТЬ ДАННЫХ, СЛОЖНОСТЬ АЛГОРИТМА, ПАРАЛЛЕЛЬНАЯ ОБРАБОТКА

Shakhovska N., Kaminsky R., Zsoba Ye. The method of associative rules mining in Big data. The paper proposes a method for Big dat analyzing a in the presence of different data sources and different methods of processing these data. The components of the problem of various data types processing are described. The concept of associative dependence was introduced, the method of finding dependencies was developed, efficiency and possibilities of its parallelism were determined. A comparative analysis of the effectiveness of methods for the search for associative rules is carried out.

GREAT DATA, ASSOCIATIVE RULES, DATA RISK, ALGORITHM COMPLEXITY, PARALLEL DEFENSE

Вступ

Сьогодні задача опрацювання різноманітних неузгоджених інформаційних ресурсів (пошуку, системної інтеграції тощо) виникає досить часто. Так, для університету прикладом інтеграції є формування наукових звітів, визначення показників успішності та якості навчання, формування рейтингу кафедри тощо; для обласної адміністрації – це визначення критичних показників розвитку регіону на основі даних, отриманих з організацій державної та недержавної форми власності.

Опрацюванням різнотипних неузгоджених даних дослідники займалися з 70-х років XX ст. Розроблені моделі та метамови опрацювання різнотипних даних. Проте, існуючі на сьогодні моделі та методи стосуються або лише наперед відомих типів даних (здебільшого, реляційні бази даних), або вирішують лише часткові задачі опрацювання різнотипних даних – наприклад, індексування для пришвидшення пошуку.

Тому виникає необхідність управління розрізненою інформацією, а саме її подання у зрозумілому для користувачів вигляді (навіть якщо вони не знають особливостей організації структур цього джерела даних) та опрацювання (пошуку, інтеграції, видобуванні нових знань тощо).

Одним із базових завдань опрацювання різнотипних даних є їхня інтеграція. Розроблені на сьогодні методи інтеграції даних за своєю функціональністю поділяються на два типи: інтеграція веб-застосувань та інтеграція на основі сховищ даних. Проте проведений аналіз літературних джерел показав, що для опрацювання інформації від усіх об'єктів галузі необхідно поєднати обидва типи інтеграції та вдосконалити наявні моделі даних.

Складові проблеми опрацювання різнотипних даних подані на рис. 1 [7].

Окремі предметні області, для яких актуальні задачі опрацювання даних з різнотипних джерел [8]:

– системи моніторингу та опрацювання новин (задача визначення правдоподібності подій на основі аналізу джерел, які ці події описували);

– системи анотування та реферування множини документів (визначення ваги терміну та речення, яке входить у кінцевий реферат);

– галузі виробництва (ієрархічна структура даних, передача даних з групуванням на вищі рівні ієрархії, об'єднання потокових даних та даних документообігу);

– медицина (визначення залежностей даних у невідомих джерелах));

– рекреаційно-туристична сфера (обмін даними між тур-операторами, інформаційними агенціями, органами керування);

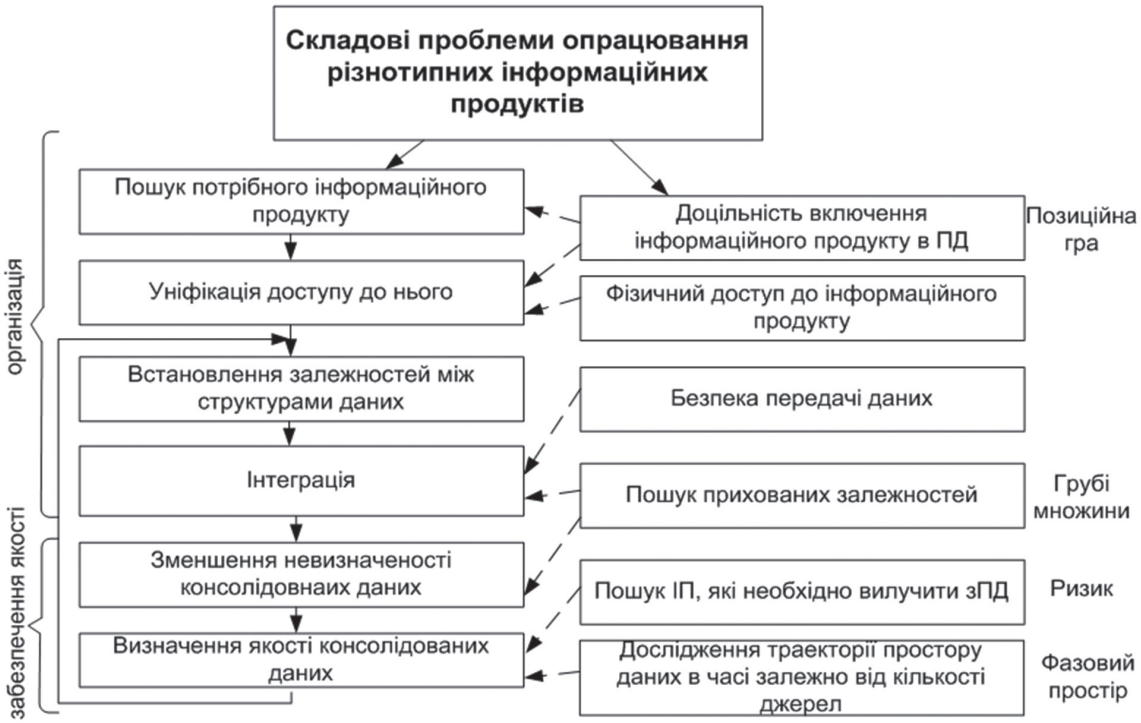

Рис. 1. Складові проблеми опрацювання різнотипних даних

– науково-навчальний процес (аналіз успішності студентів, облік публікацій, участі у конференціях, олімпіадах);

– аналіз наукових джерел (формування показника якості кафедр; визначення перспективності наукового напряму, унікальності робіт науковця, використання даних для міжгалузевих досліджень).

Наведені вище приклади перетинаються з іншим поширеним поняттям – Великі дані [9].

Великі дані (Big Data) в інформаційних технологіях – набір методів та засобів опрацювання структурованих і неструктурованих різнотипних динамічних даних великих обсягів з метою їх аналізу та використання для підтримки прийняття рішень. До цього класу відносять засоби паралельного опрацювання даних (NoSQL, алгоритми MapReduce, Hadoop). Визначаль-ними характеристиками для Великих даних є обсяг, швидкість, різноманіття.

Визначено, що для представлення Великих даних використовують багатовимірну та об'єктну моделі. Багатовимірне представлення даних добре використовувати для задач візуалізації даних та їх аналізу, але у зв'язку з розрідженістю гіперкуба обсяг даних у такому випадку є більший порівняно з реляційним представленням, що є неприпустими до Великих даних. Об'єктне подання дає змогу зберігати об'єкти предметної області у вигляді атрибутів, їх характеристик та зв'язків між характеристиками. За певної модифікації воно може бути використане для Великих даних. Проте залишається нерозв'язаною задача трансформації різних типів даних (текстових, напівструктурованих) в об'єктну модель.

Багато існуючих методів аналізу даних непридатні до застосування для Великих даних, оскільки

• розмірність даних надзвичайно велика,

• за своєю структурою та через різноманітні джерела походження дані є неочищеними, є багато відхилень,

• необхідно застосовувати паралельну обробку даних.

Тому метою статті є модифікація методу пошуку асоціативних правил для роботи з Великими даними.

Правила асоціації та генерація правил широко використовуються, і вони стикаються з багатьма проблемами, головним з яких є наявність великих даних та багатовимірних наборів даних [4]. Однопроцнсорні системи з нормальною швидкістю процесора не можуть впоратися з такими великими даними, що робить алгоритм неефективним для використання. В останніх розробках зростання мережевих технологій та особливо хмарних платформ дало нові ідеї щодо генерації мультиатрибутних правил, використовуючи паралельне середовище, таке як Hadoop [5]. MapReduce був популярним і більше використовуваним для обчислення великої кількості даних з тих пір, як компанія Google її запустила на своїй платформі. Розподілена файлова система Google (GFS) та вебслужба Amazon (AWS) використовують платформу Hadoop та MapReduce для надання своїх послуг.

У випадку генерації асоціативних правил в MapReduce, Mapper відповідає за завдання отримати різні комбінації елементів як "ключ", а "значення" використовується для відстеження кількості входів або підрахунку підтримки. Задачею Редуктора є зменшити отриманий від Mapper набір для кожного ключового значення та обчислити остаточну підтримку для всіх наборів даних кандидатів.

Таким чином асоціативні правила можуть бути створені з максимальною підтримкою та впевненістю.

Алгоритм Apriori має великі проблеми з великими обсягами Big Data, оскільки він сканує всю базу даних кілька разів. Це означає, що час виконання збільшується відповідно до кількості транзакцій. У нашому дослідженні ми використовували Spark та ієрархічний метод формування правил для покращення алгоритму Apriori.

Основний матеріал

Визначення асоціативної залежності

Введемо поняття асоціативної залежності. Шукатимемо залежність на відношенні *r*. Це відношення може бути сформоване як для реляційних джерел даних, так і для нереляційних (NoSQL) шляхом формування пари значень - назви об'єкта та його характеристик. Для різних об'єктів кількість характеристик може бути різною. У такому випадку відношення *r* формуватиметься як CROSS(*r*). Далі по тексту вживатимемо тільки позначення *r*.

Асоціативна залежність (АЗ) – це продукційне правило в селекції відношення *r*, яке справджується для значущої кількості об'єктів цієї селекції. Поріг значущості повинен визначатись експертним шляхом, або виходячи з розрахунків імовірності помилкового виділення цієї залежності.

$$\begin{gathered} F_I(S;T):(s;t)=\left(r_S(R);r_T(R)\right), \\ i=\overline{1..n_s},\forall i: A_i\in R, \\ j=\overline{1..n_T},\forall j: A_j\in \boldsymbol{R}, \\ s=r_S(R)=\sigma_{S(\{A_i\})}\left(r(R)\right), \\ t=r_T(R)=\sigma_{T(\{A_j\})}(s), \end{gathered} \quad (1)$$

де S, T – предикати селекції умовної та результуючої частини відповідно, s, t – результати операцій селекції за цими предикатами з основного відношення. Позначення $s(\{A_i\})$ означає, що предикат використовує множину атрибутів $\{A_i\}$.

Основними параметрами традиційних асоціативних правил є рівень довіри та рівень підтримки.

Рівень довіри – відношення кількості об'єктів, для яких має місце така АЗ до кількості об'єктів в селекції:

$$Conf(S\to T)=P(S\to T)=\frac{\left|\sigma_{S\wedge T}(r)\right|}{\left|\sigma_S(r)\right|}. \quad (2)$$

Рівень підтримки – характеристика предиката селекції на відношенні, що обчислюється як відношення кількості об'єктів, які задовольняють предикат P до загальної кількості об'єктів у відношенні:

$$Supp(P)=\frac{\left|\sigma_P(r)\right|}{|r|}. \quad (3)$$

У випадку обчислення рівня підтримки для АЗ умовний та результуючий предикат залежності об'єднуються знаком кон'юнкції:

$$Supp(S\to T)=Supp(S\wedge T)=\frac{\left|\sigma_{S\wedge T}(r)\right|}{|r|}. \quad (4)$$

З використанням цього поняття рівень довіри можна обчислити, як

$$Conf(S\to T)=\frac{Supp(S\to T)}{Supp(S)}. \quad (5)$$

Рівень покращення обчислюється, як відношення рівнів довіри та підтримки АЗ:

$$Imp(S\to T)=\frac{Conf(S\to T)}{Supp(T)}=\frac{Supp(S\wedge T)}{Supp(S)\cdot Supp(T)}.$$

Рівень підтримки та рівень покращення є "симетричними", тобто, $Sup(X\to Y)=Sup(Y\to X)$ та $Imp(X\to Y)=Imp(Y\to X)$. Те ж справедливо й для повної взаємної інформації. Рівень довіри є "направленим" параметром, тобто, $Conf(X\to Y)\neq Conf(Y\to X)$. Цим пояснюється непряма залежність інформативності від рівня довіри.

Розширено поняття R-"цікавості" асоціативного правила, яке використовується для відбору виявлених асоціативних залежностей для скорочення їх кількості. R-"цікавість" визначається, як перевищення рівнем довіри або підтримки очікуваних значень у R разів. Очікувані значення розраховуються для асоціацій, що мають не менше двох ознак у лівій частині правила з припущенням, що вони статистично незалежні. Інформативність дає змогу оцінити "цікавість" якщо, наприклад, рівень підтримки правила в R_1 разів більший очікуваного значення, а рівень довіри – у R_2 разів менший і навпаки.

"Цікавість" асоціації визначається як перевищення інформативності асоціації очікуваного значення. Таким чином інформативність розглядається, як узагальнення поняття R-"цікавості" асоціативного правила.

Метод пошуку асоціативної залежності

Побудуємо метод пошуку АЗ [3].

Вхідні дані

1. Відношення CROSS(*r*), схема *R* визначена лише на аналізованій селекції.

2. Хеш-функції для кожного атрибуту відношення *R*: $h_j(A_j)$.

3. Порогове значення рівня довіри залежностей, що шукаються – p_0. Замість даного параметра рівноцінно може використовуватись кількість кортежів, на яких повинна бути визначена шукана залежність – minSupport.

4. Порогове значення рівня довіри залежностей, що враховуються при утворенні нових залежностей $= p^*$.

Вихідні дані

Множина асоціативних залежностей, що відповідає вказаним критеріям:

$\{S_i\to T_i\},\forall i: Conf(S_i\to T_i)\geq p_0$

І етап. Аналіз вхідних даних.

1. Створити дерево хеш-таблиць статистики значень атрибутів відношення. Хеш-таблиця кожного атрибута A_j міститиме відповідність значень цього атрибута структурі (кількість повторів значення; масив описів умовних значень інших атрибутів). Опис умовних значень – хеш-таблиця значень атрибута A_k та кількості їх повторень на множині кортежів $X = R_{A_j = x_i[A_j]}$.

Вершини парних рівнів дерева розгалужуються по імені атрибута, на який здійснюється проекція; непарних рівнів – значення атрибута батьківської вершини.

Хеш-таблиці непарних рівнів дерева використовують наперед задані хеш-функції $h_j(A_j)$; для парних рівнів використовується внутрішня хеш-функція порівняння атрибутів на рівність.

2. Заповнити структуру, описану в п.1. Заповнюється не все дерево повністю, а лише гілки, що відповідають наперед заданим критеріям якості залежностей. Додаткове уточнення статистики можливе, використовуючи принцип відкладених обчислень:

a. Для кожного кортежу $x \in r(R)$:
b. Ініціалізувати поточну вершину дерева статистики v коренем дерева: $v = a$.
c. Ініціалізувати $startAttr = 0$ – номер атрибуту, з якого починати розгалуження дерева.
d. Для кожного атрибуту $A_j \in R, j \geq startAttr$:
e. Якщо не існує гілка дерева статистики $v[j]\left[x\left[A_j\right]\right]$,
f. Створити вершини $v[j]$ та $v[j]\left[x\left[A_j\right]\right]$;
g. Ініціалізувати атрибути

$v[j]\left[x\left[A_j\right]\right].count = 0$,
$v[j]\left[x\left[A_j\right]\right].childs = \varnothing$,
$v[j]\left[x\left[A_j\right]\right].nextAttrId = startAttr + 1$,

Якщо $v.count > splitThreshold$,
h. Рекурсивно перейти до кроку 2d.

Таким чином, у структурі даних а містяться елементарні залежності виду

$$A_j = x_i\left[A_j\right] \rightarrow A_k = x_i\left[A_k\right],$$

а також забезпечується можливість обчислення кількості кортежів довільних проекцій

$$\left|\sigma_{A_{i_1}=v_{i_1} \wedge A_{i_2}=v_{i_2} \wedge \ldots \wedge A_{i_l}=v_{i_l}}(r)\right|.$$

3. Оголосити список залежностей $z[l]\{S:set; T:set; NS:N; NT:N\}$. Кожна залежність представляє собою структуру з атрибутами: S – множина значень атрибутів умовної частини залежності, на яких вона визначена; T – множина значень атрибутів результуючої частини залежності, на яких вона визначена; NS – кількість кортежів, для яких залежність справджується; NT – кількість кортежів, для яких виконується умовна частина залежності.

4. Заповнити список Z залежностями $Conf\left(A_j = x_i\left[A_j\right] \rightarrow A_k = x_i\left[A_k\right]\right) \geq p_0$ проходженням по структурі даних а.

ІІ етап. Агрегування залежностей.

1. Скопіювати список Z у окремий список залежностей Aggr;
2. Оголосити хеш-таблицю h_{aggr} множин значень результуючих частин КАЗ. Це дасть змогу ефективно шукати АЗ, з якими можна агрегувати кожну конкретну залежність d_{aggr} ;
3. Для кожної залежності $z_1 \in Aggr$:
4. Для кожної залежності:
5. Утворити залежність $z_3 = z_1 + z_2$;
6. Якщо $Conf(z_3) \geq p^*$, додати z_3 в кінець списку Aggr для подальшого агрегування з іншими залежностями.
7. Занести залежність z_1 у хеш-таблицю h_{aggr} з ключем $h(z_1[\Pr T])$.

ІІІ етап. Побудова АЗ.

1. Оголосити список залежностей y ;
2. Ініціалузувати $Y = Aggr$;
3. Оголосити хеш-таблицю результуючих частин предикатів pr для ефективного пошуку множини залежностей, що мають однакову результуючу частину предикату;
4. Для кожної залежності $F_I \in Y$ занести АЗ F_I у відповідний список хеш-таблиці: $pr\left[h\left(F_I[\Pr T]\right)\right]$.

Побудований алгоритм пошуку асоціативних залежностей дає змогу проводити ефективний аналіз однотипних даних на наявність АЗ, сумарна складність якого по часу складає

$$t = O\left(minSupport \cdot \left(\frac{n}{minSupport}\right)^{1+\log_{avgD}(m)} + \right.$$
$$\left. + \frac{Z_{al}^2}{m \cdot D(A)} + \frac{Z_{aggr}^2 \cdot \log\left(sz_{aggr}\right)}{m \cdot D(A)}\right) =$$
$$= O\left(minSupport \cdot \left(\frac{n}{minSupport}\right)^{1+\log_{avgD}(m)} + \right.$$
$$\left. + \frac{Z_{aggr}^2 \cdot \log\left(sz_{aggr}\right)}{m \cdot D(A)}\right).$$

Складність алгоритму по пам'яті рівна

$$M = O\left(M_{stat} + M_{aggr} + M_{ma}\right) =$$
$$= O\left(\left(\frac{n}{minSupport}\right)^{1+\log_{avgD}(m)} + Z_{aggr} \cdot sz_{aggr} + Z_{ma} \cdot sz_{ma}\right) =$$
$$= O\left(\left(\frac{n}{minSupport}\right)^{1+\log_{avgD}(m)} + Z_{ma} \cdot sz_{ma}\right).$$

Друга компонента суми домінується третьою як у випадку часу аналізу, так і для використаної пам'яті. Порівнювати ж перший і третій член суми неможливо в загальному випадку: їхній розмір

по-різному залежить від обсягу даних та структури даних.

Низька асимптотична складність розробленого алгоритму виявлення АЗ та широкий набір типів даних, що підтримуються для аналізу, дають змогу застосовувати розроблений алгоритм практично у всіх предметних галузях, що працюють із асоціативними залежностями в даних.

Наведемо реалізацію пошуку асоціативних залежностей в R:

```
install.packages("arules")
library(arules)
library(Matrix)
data("Adult")
## find only frequent itemsets which do not contain small or large income
is <- apriori(Adult, parameter = list(support= 0.1, target="frequent"),
        appearance = list(none = c("income=small", "income=large"), default="both"))
itemFrequency(items(is))["income=small"]

## find itemsets that only contain small or large income and young age
is <- apriori(Adult, parameter = list(support= 0.1, target="frequent"),
        appearance = list(items = c("income=small", "income=large", "age=Young"), default="none"))
inspect(head(is))
## find only rules with small or large income in the right-hand-side.
rules <- apriori(Adult, parameter = list(support=0.2, confidence = 0.5),
                 appearance = list(rhs = c("income=small", "income=large"), default="lhs"))
inspect(head(rules))
install.packages("arulesViz")
library(arulesViz)
plot(rules, method="graph", control=list(type="items"))
plot(rules)
plot(rules,method="paracoord",control=list(reorder=TRUE))
```

Результат аналізу подано на рис. 2.

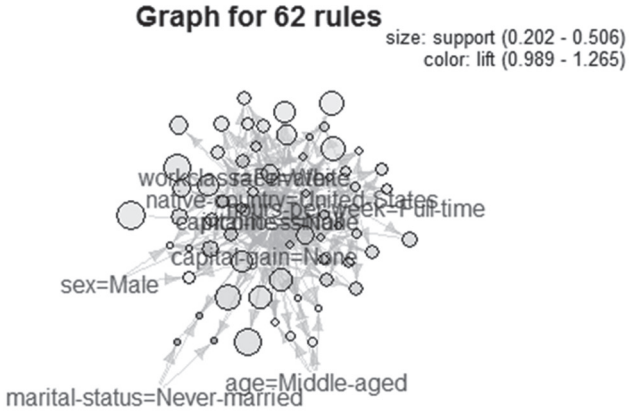

Рис. 2. Візуалізація знайдених асоціативних залежностей

Порівняльний аналіз алгоритмів виявлення асоціативних залежностей у даних за кількісними критеріями

Для порівняння ефективності розробленого методу аналізу даних було обрано три найближчих за призначенням методи: Apripri, HybridApriori та метод FP-tree.

За кількісні критерії було обрано кількість корисних залежностей, відсоток корисних залежностей у множині знайдених залежностей що виявив метод, а також час аналізу даних.

Корисна залежність – це залежність у даних, яка має коефіцієнт корисності не менше 0.5 при оцінюванні експертами предметної області, тобто

$$\frac{\left(\frac{1}{n_e}\sum_{i=1}^{n_e} m_i\right) - markMin}{markMax - markMin} \geq 0.5,$$

де m_i – оцінка i-го експерта, n_e – кількість експертів, що беруть участь у оцінюванні, $markMin$ – нижня границя шкали оцінювання, $markMax$ – верхня границя шкали оцінювання.

Тобто, корисними залежностями вважаються ті, що експерти вважають корисними для знання предметної області.

Очевидно, для визначення цієї оцінки експертам необхідно проаналізувати усі залежності, знайдені в даних автоматичними засобами аналізу. Необхідність залучення експертів на цьому етапі (інтерпретація результатів) не є проблемою, оскільки завданням автоматизованих засобів аналізу даних якраз і є висунення припущень про важливі закономірності в даних, а не судження про важливість знайдених залежностей. Тобто, інтерпретація результатів аналізу – все ще відповідальність людини.

Відсоток корисних залежностей – відношення кількості залежностей, що експерти вважають корисними до загальної кількості залежностей, що знайшов метод. Даний критерій досить важливий, оскільки визначає коефіцієнт ефективності роботи експертів по інтерпретації даних. Очевидно, що метод аналізу даних, який знаходить тисячі важливих залежностей, але "загублених" поміж мільйонів очевидних фактів, не буде корисним і не може мати практичного застосування.

Швидкість аналізу даних – відносно простий для вимірювання критерій і визначається часом, необхідним обчислювальній системі на виконання запиту аналізу даних.

Тестування усіх методів відбувалось на тих же даних. Набір даних, що використовується для цього прикладу – це набір даних «Adult», розміщений в Machine Learning Repository UCI. Він містить близько 2320000 спостережень з 15 змінними:

- age – вік людини,
- type_employer – тип роботодавця (фізична чи юридична особа, урядовець, військовий тощо,
- fnlwgt – кількість людей, що підтвердили істинність цього спостереження,
- education – найвищий рівень освіти, досягнутий цією людиною,
- education_num – найвищий рівень освіти в числовій формі,

- marital – сімейний стан,
- occupation – рід занять,
- relationship – сімейні відносини,
- race - раса особи,
- sex - стать,
- capital_gain – зафіксований приріст капіталу,
- capital_loss – зафіксовані капітальні втрати,
- hr_per_week – кількість відпрацьованих годин за тиждень,
- country – країна походження,
- income – булева змінна, так, якщо річний дохід особи більший за $ 50000.

Результати застосування розробленого методу генерації залежностей та порівняння його з існуючими подані в Таблиці 1.

Як бачимо, кількість знайдених залежностей не зростає експоненсіально залежно від розміру вхідних даних, як можна було б припустити. Більше того, ця величина наче обмежена зверху деякою границею. Це можна пояснити тим, що дані, які аналізувались, досить однорідні.

Таблиця 1

Кількість знайдених корисних залежностей різними методами від об'єму проаналізованих даних

Кількість записів	Розроблений метод	Метод FP-tree [2]	Метод Apriori	Метод HybridApriori [1]
200000	2856	2199	720	849
400000	5220	3627	888	1008
600000	6657	4530	1011	1158
800000	7656	5136	1053	1278
1000000	8043	5274	1104	1329
1242961	8184	5382	1125	1377

Тому в кожному наступному блоці даних все більше залежностей повторювались із попередніх блоків, відповідно, ці дані не зумовлювали генерацію нових залежностей, а лише збільшували рівень підтримки існуючих. Ті залежності, що внаслідок таких змін стали відповідати критеріям пошуку, і були додані у результуючу множину.

Ще одним фактором, що міг вплинути на кількість залежностей, було сортування даних по даті народження водія. Таким чином, кожен наступний блок даних усе ж вносив дійсно нові залежності, і їх не могло бути виявлено в попередніх блоках даних. Тому все ж можна припустити, що кількість залежностей необмежена зверху якоюсь чіткою межею, але прямує до прямо пропорційної залежності із досить малим кутовим коефіцієнтом від кількості кортежів даних.

Для даних, що мають практичну цінність, кількість залежностей Z_{aggr} складає не більше декількох тисяч у той час, як кількість кортежів n може обчислюватися мільярдами. Тому у випадку паралельних обчислень на значній кількості комп'ютерів другим членом функції t_n можна знехтувати. З тих же міркувань $\log_2(n) = o(minSupport)$.

Таким чином, асимптотична оцінка часу виконання алгоритму на системі із k комп'ютерів складає

$$t_n = O\left(minSupport^{-\log_{avgD}(m)}\right)$$

Отримана оцінка часу виконання алгоритму є субполіноміальною, а отже, розроблений алгоритм є ефективним паралельним алгоритмом.

Розроблений алгоритм дає змогу стверджувати, що задача виявлення асоціативних залежностей у розподілених базах даних належить до класу P-задач. Отже, алгоритм пошуку асоціативних залежностей добре вирішувати з допомогою MapReduce.

Крім кількох послідовних реалізацій, паралельні реалії для роботи з великими даними не є широко доступними. Одним із прикладів серійної реалізації добре відомий статистичний обчислення з пакетом R, що називається «arules».

Паралельна реалізація програми FP-Groth доступна в бібліотеці для вивчення комп'ютера з відкритим вихідним кодом (MLlib) Apache Spark та Apache Mahout.

Висновки

Асоціативні правила асоціації широко використовуються у великій кількості додатків. Наш алгоритм може бути використаний в деяких з цих програм, в яких використання традиційних алгоритмів не є життєздатним через величезну кількість даних, які потрібно обробляти. Це може бути дуже корисним, наприклад, у сенсорних мережах, які генерують величезну кількість даних за короткі проміжки часу, або в соціальних мережах, у яких є мільйони користувачів. Нарешті, слід зазначити, що ці процедури можна узагальнити на інші методи видобування даних, які використовують правила асоціації, такі як винятки та аномалії [4], поступові залежності [5, 6] та ін.

Список літератури: 1. *Zaki*, M. J. (2000). Scalable algorithms for association mining. IEEE Transactions on Knowledge and Data Engineering12(3): 372–3900c. **2.** *J. Han, H. Pei, and Y. Yin*. Mining Frequent Patterns without Candidate Generation. In: Proc. Conf. on the Management of Data (SIGMOD'00, Dallas, TX). ACM Press, New York, NY, USA 2000. **3.** *Пшеничний О. Ю.* Математичне та програмне забезпечення виявлення кон'юнктивних асоціативних залежностей у великих масивах даних : автореферат дисертації на здобуття наукового ступеня кандидата технічних наук : 01.05.03 – математичне та програмне забезпечення обчислювальних машин і систем / Олександр Юрійович Пшеничний ; Національний університет «Львівська політехніка» . - Львів, 2012. - 24 с. **4.** Delgado, M., Ruiz, M.D. & S nchez, D., New approaches for discovering exception and anomalous rules. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 19(02), pp. 361–399, 2011. **5.** *Berzal, Fernando, et* al., A new framework to assess association rules. In Advances in Intelligent Data Analysis, Springer Berlin: Heidelberg, pp. 95–104, 2001. **6.** *H llermeier, E.,* Association rules for expressing gradual dependencies. In Principles of Data Mining and Knowledge Discovery, Springer Berlin:

Heidelberg, pp. 200–211, 2002. **7**. *Шаховська Н.Б.* Програмне та алгоритмічне забезпечення сховищ та просторів даних:мМонографія / Міністерство освіти і науки України, Національний університет «Львівська політехніка». – Львів, 2010. – 194 с. **8**. *Шаховська Н.Б.* Структура та задачі простору даних // Складні системи і процеси. — 2008. — № 1. — С. 73—86. **9**. Big Data Dimensions, http://www.klarity-analytics.com/392-dimensions-of-big-data.html **10.** *Agrawal, Rakesh, Imielinski, Tomasz & Swami, Arun,* Mining association rules between sets of items in large databases. *ACM SIGMOD Record*, 22, pp. 207–216, 1993.

RESUME

N.B. Shakhovska, R.M.Kaminskiy, Eu.O.Zasoba

The method of associative rules mining from Big data

The paper proposes a method for analyzing Big data in the presence of different data sources and different methods of processing these data. It is determined that multidimensional and object models are used to represent the Great Data. The multidimensional view of the data is well used for data visualization and analysis tasks, but due to the hypercube dissipation, the amount of data in this case is greater than the relational representation that is not acceptable to the Big Data. Object representation allows you to store object in the form of attributes, their characteristics and relationships between characteristics. For some modification, it can be used for Big Data. However, the problem of the transformation of various types of data (text, semi-structured) into an object model remains unresolved. Associative rules are widely used in a large number of applications. The notion R- "curiosity" of the associative rule is expanded, which is used to select identified associative dependencies to reduce their number. R- "curiosity" is defined as an excess of trust or support of the expected values in R times. Expected values are calculated for associations with at least two attributes on the left side of the rule with the assumption that they are statistically independent. Informativity allows to evaluate "curiosity" if, for example, the level of support rule in R1 is more than expected value, and the level of trust - in R2 times more than the other and vice versa. Our algorithm can be used in some of these programs, in which the use of traditional algorithms is not viable because of the huge amount of data that needs to be processed. This can be very beneficial, for example, in sensor networks that generate a huge amount of data at short intervals or in social horror that has millions of users. Finally, it should be noted that these procedures can be used in general to other methods of data mining using the right-wing associations, such as exceptions and anomalies, gradual dependencies etc. The concept of associative dependence is introduced, the method of finding dependences is developed, efficiency and possibility of its parallelization are determined.

Поступила в редколлегию 14.09.2017