Jerzy Korczak, Piotr Skrzypczak

Algorytm FP-Growth w wyszukiwaniu wzorców zakupów klientów

Ekonomiczne Problemy Usług nr 67, 61-71

2011

ZESZYTY NAUKOWE UNIWERSYTETU SZCZECIŃSKIEGO

NR 650 EKONOMICZNE PROBLEMY USŁUG NR 67 2011

JERZY KORCZAK
Uniwersytet Ekonomiczny we Wrocławiu
PIOTR SKRZYPCZAK
Delikatesy Alma, Wrocław

ALGORYTM FP-GROWTH W WYSZUKIWANIU WZORCÓW ZAKUPÓW KLIENTÓW

Wprowadzenie

Jedną z najpopularniejszych i najczęściej stosowanych metod wyszukiwania wzorców zakupów klientów są metody wykorzystujące algorytmy reguł asocjacyjnych. Na przestrzeni kilkunastu lat powstało wiele algorytmów wyszukiwania reguł asocjacyjnych w zbiorach danych[1]. Wiele z nich ewoluowało, część z nich okazała się mniej użyteczna ze względu na swą szybkość działania na ogromnych zbiorach danych i zbyt duże wymagania dostępnej pamięci. Jedną z interesujących propozycji jest algorytm FP-Growth opracowany przez J. Hana, H. Pei i Y. Yina[2]. Algorytm ten jest szybki i skalowalny. W publikacji J. Han pokazał, że FP-Growth przewyższa wydajnością inne popularne metody wyszukiwania reguł asocjacyjnych, takie jak algorytm Apriori czy TreeProjection. W pracach[3] wykazano, że algorytm ten ma lepsze wyniki niż Eclat i Relim.

---

[1] D. Hand, H. Mannila, P. Smyth: *Eksploracja danych*, Wydawnictwo Naukowo-Techniczne WNT, Warszawa 2005; T. Morzy: *Eksploracja danych*, http://www.portalwiedzy.pan.pl/images/ stories/pliki/publikacje/nauka/2007/03/N_307_06_Morzy.pdf (czerwiec 2010); A. Pasztyła: *Analiza koszykowa danych transakcyjnych – cele i metody*, http://www.statsoft.pl/pdf/artykuly/bas ket.pdf (czerwiec 2010).

[2] J. Han, J. Pei, Y. Yin: *Mining Frequent Patterns without Candidature Generation*, w: Proc. of the 200 ACM SIGMOD Int. Conf. on Management of Data, Dallas 2000, s. 1–12.

[3] M. Zaki, S. Parthasarathy, M. Ogihara, W. Li: *New Algorithms for Fast Discovery of Association Rules. Proc.* 3rd Int. Conf. on Knowledge Discovery and Data Mining (KDD '97), AAAI Press 1997; C. Borgelt: *Keeping Things Simple: Finding Frequent Item Sets by Recursive*

Popularność i efektywność algorytmu FP-Growth doceniona została w wielu badaniach, w których w celu poprawy jego wydajności zaproponowano wiele zmian w oryginalnym algorytmie[4], Zmiany te dotyczyły głównie przyspieszenia procesu budowy *FP-Drzewa*, jego redukcji i zmniejszenia złożoności czasowej oraz pamięci algorytmu.

Pierwsza modyfikacja zainspirowana została przez C. Gyorödi[5]. Sygnalizuje on dwa problemy w algorytmie FP-Growth, mianowicie powstałe *FP-Drzewo* nie jest unikatowe dla tej samej „logicznie" bazy danych oraz to, że w celu utworzenia *FP-Drzewa* wymagane są dwa pełne skanowania bazy danych. W zaproponowanym algorytmie DynFP-Growth pierwszy problem rozwiązuje się przez wprowadzenie kolejności wsparcia według porządku leksykograficznego, zapewniając w ten sposób unikatowość *FP-Drzewa* dla różnych, lecz „logicznie równoważnych" baz danych. W celu rozwiązania drugiego problemu opracowano algorytm dynamicznej zmiany kolejności elementów *FP-Drzewa*, przeprowadzając wewnątrz tego algorytmu „promocję" (przesunięcie) do wyższego rzędu jednego najmniejszego wykrytego elementu. Ważną cechą tego rozwiązania jest to, że nie jest konieczna odbudowa *FP-Drzewa*, gdy baza danych jest aktualizowana.

Sposób zmniejszenia wielkości podano w algorytmie FP-Bonsai[6], który poprawia wydajność FP-Growth, przycinając *FP-Drzewo*, za pomocą techniki ExAnte redukcji danych[7]. Przycięte *FP-Drzewo* nazwano *FP-Bonsai*. Oryginalność tego rozwiązania polega na odrzuceniu już przy pierwszym skanowaniu zbioru danych pozycji, których wsparcie jest mniejsze niż założona minimalna wartość. Dodatkowo, po zakończeniu pierwszego skanowania i po utworzeniu tablicy nagłówkowej cała tablica jest sortowana według wartości wsparcia i ponownie są odrzucane pozycje ze wsparciem mniejszym od założonej minimalnej wartości. Dzięki temu wielkość *FP-Drzewa* zmniejsza się kilkakrotnie.

Ostatnia z wymienionych modyfikacji dotyczy czasu pracy oraz wymagań pamięci dla algorytmu FP-Growth. Algorytm NONORDFP[8] modyfikuje strukturę *FP-Drzewa*, która jest dzięki temu bardziej zwarta i nie potrzebuje odbudowywania go

---

Elimination, Workshop Open Source Data Mining Software (OSDM '05, Chicago, IL), ACM Press 2005.

[4] C. Gyorödi, R. Gyorödi, T. Cofeey, S. Holban: *Mining association rules using Dynamic FP-trees*, In proceedings of The Irish Signal and Systems Conference, University of Limerick, s. 76–82, 2003; B. Rácz, *Nonordfp: An FP-Growth Variation without Rebuilding the FP-Tree*, 2nd Int'l Workshop on Frequent Itemset Mining Implementations FIMI2004; M. Zaki, S. Parthasarathy, M. Ogihara, W. Li: *New Algorithms…*, *op. cit*.

[5] C. Gyorödi, R. Gyorödi, T. Cofeey, S. Holban: *Mining association…*, *op. cit*., s. 76–82.

[6] *Ibidem.*

[7] F. Bonchi, F. Giannotti, A. Mazzanti and D. Pedreschi: *Exante: Anticipated data reduction in constrained pattern mining*, In Proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD '03), Cavtat–Dubrovnik 2003, s. 3–7.

[8] D. Hand, H. Mannila, P. Smyth: *Eksploracja danych…*, *op. cit*.

dla każdego warunkowego kroku. Nowa reprezentacja *FP-Drzewa* w pamięci umożliwia szybsze przeszukiwanie drzewa, szybszy przydział i ewentualnie projekcję.

W artykule przedstawimy wyniki badań pilotażowych zachowań klientów jednego z wrocławskich sklepów sieci Delikatesy Alma, na podstawie danych sprzedaży z sierpnia i września 2009[9] oraz badań zrealizowanych według nowego projektu procesu na danych z okresu sierpień 2009 – styczeń 2010. W badaniach wykorzystaliśmy system bazodanowy MySQL, autorskie oprogramowanie DM Cafe oraz pakiet Rapid Miner, w którym zaimplementowany został algorytm FP-Growth. Celem głównym stworzonej platformy było wspomaganie decydentów w wyszukiwaniu interesujących, nietrywialnych reguł asocjacyjnych, które pozwolą na lepsze poznanie klientów i ich preferencji, a także na poprawienie wyników sprzedaży.

Artykuł został podzielony na cztery rozdziały. W pierwszym opisano algorytm FP-Growth i scharakteryzowano główne parametry algorytmu. W drugim rozdziale przedstawiona została baza danych zawierająca dane transakcyjne oraz informacje o towarach. W trzecim opisano proces ekstrakcji reguł asocjacyjnych w środowisku *Rapid Miner*[10]. Przedstawiono również ideę modyfikacji procesu ekstrakcji reguł asocjacyjnych w odniesieniu do platformy badawczej użytej w pracy[11]. W rozdziale opisano zmodyfikowany proces eksploracji danych oraz zbadano wpływ zmian w procesie na szybkość platformy testowej.

1. Opis algorytmu FP-Growth

Algorytm zawiera dwa podstawowe kroki: kompresję zbioru danych do *FP-Drzewa* oraz eksplorację *FP-Drzewa*[12].

W pierwszym kroku algorytm przeszukuje bazę danych w celu znalezienia wszystkich jednoelementowych zbiorów częstych. Kolejnym krokiem jest usunięcie nieczęstych elementów z transakcji T_i, co w efekcie daje zmodyfikowany zbiór transakcji $T = T_1,..., T_n$, składający się wyłącznie z jednoelementowych zbiorów częstych. Następnie zbiór transakcji jest sortowany malejąco według wsparcia każdej transakcji. Po tym kroku transakcje są transformowane do postaci *FP-Drzewa*. *FP-Drzewo* jest ukorzenionym grafem acyklicznym, etykietowanym w wierzchołkach. Korzeń grafu posiada etykietę *null*, pozostałe wierzchołki grafu, zarówno wierzchołki wewnętrzne, jak i liście, reprezentują jednoelementowe zbiory częste.

---

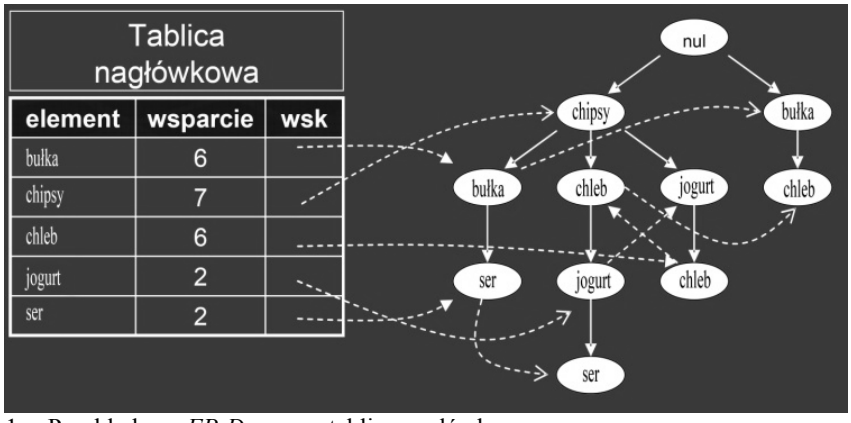
[9] P. Skrzypczak: *Modelowanie wzorców zachowań klientów Delikatesów Alma przy wykorzystaniu reguł asocjacyjnych*, Uniwersytet Ekonomiczny, Wrocław 2010.

[10] M. Bereta: *Data Mining z wykorzystaniem programu RapidMiner*, http://michalbereta.pl/dydaktyka/ZSI/Lab%20Data%20Mining%201.pdf (czerwiec 2010).

[11] P. Skrzypczak: *Modelowanie wzorców…*, *op. cit*.

[12] J. Han, J. Pei, Y. Yin: *Mining Frequent…*, *op. cit*., s. 1–12.

Z każdym wierzchołkiem grafu, z wyjątkiem korzenia, związana jest etykieta reprezentująca jednoelementowy zbiór częsty oraz licznik transakcji, reprezentujący liczbę transakcji wspierających dany zbiór (rysunek 1).



Rys. 1. Przykładowe *FP-Drzewo* z tablicą nagłówkową

Źródło: opracowanie własne na podstawie danych Delikatesów Alma.

Procedura FP-Growth (rysunek 2) przedstawia dwa podstawowe kroki algorytmu. Algorytm ma dwa początkowe parametry: *Tree* = *FP-Drzewo* oraz α = *null*. W przypadku gdy *FP-Drzewo* posiada tylko pojedynczą ścieżkę *p*, wtedy dla każdej kombinacji β wierzchołków ścieżki *p* tworzony jest zbiór $\beta \cup \alpha$ o wsparciu równym minimalnemu wsparciu elementów należących do zbioru β. Jeśli *FP-Drzewo* zawiera więcej niż jedną ścieżkę, to na każdy element α_i należący do tablicy nagłówków *Tree* tworzony zbiór $\beta = \alpha_i + \alpha$ o wsparciu odpowiadającym wsparciu elementów α_i. Następnie generowana jest warunkowa baza wzorca β i warunkowe *FP-Drzewo* wzorca β, oznaczone *Tree-*β. Po tym kroku sprawdzane jest, czy *Tree-*β jest niepuste; jeśli jest puste, przerywany jest algorytm, w przeciwnym wypadku ponownie uruchamiana jest procedura FP-Growth z parametrami *Tree* = *Tree-*β i $\alpha = \beta$.

2. Baza danych transakcyjnych sklepu internetowego Alma24

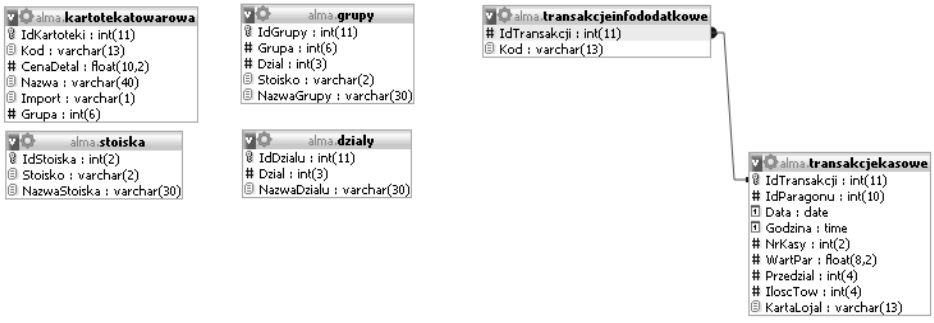
Przeprowadzanie badań wymagało zaprojektowania platformy umożliwiającej połączenie aktualnych systemów transakcyjnych z nowymi funkcjonalnościami ekstrakcji reguł. W platformie eksperymentalnej wyróżniono trzy komponenty funkcjonalne: bazę danych transakcyjnych sklepu z informacjami kartotek towarowych, oprogramowanie do wymiany informacji między systemem magazynowym i kasowym oraz aplikację do wyszukiwania reguł asocjacyjnych. Baza danych, zarządzana przez serwer MySQL, składa się z następujących tabel: *KartotekaTowa-*

rowa, Grupy, Działy, Stoiska, TransakcjeKasowe oraz TransInfoDodatkowe (rysunek 3).

```
PROCEDURE FP-GROWTH (TREE, A)
                                        IF TREE ZAWIERA POJEDYNCZĄ ŚCIEŻKĘ
                                THEN FOR EACH KOMBINACJI B WIERZCHOŁKÓW ŚCIEŻKI P I
                                                                        END I
  ELSE FOR EACH A-I NALEŻĄCEGO DO TABLICY NAGŁÓWKÓW ELEMENTÓW TREE I
                                        IF TREE-B ≠ Ø THEN FP-GROWTH(TREE-B, B)
END PROCEDURE;
```

Rys. 2. Algorytm – procedura FP-Growth

Źródło: http://wazniak.mimuw.edu.pl/images/3/3f/ED-4.2-m03-1.0-kolor.pdf



Rys. 3. Schemat bazy Alma

Źródło: opracowanie własne.

Do wymiany informacji między systemami i bazą danych wykorzystany zostanie program DM Cafe. Jest on niewielkim programem przygotowanym na potrzeby eksploracji danych w Almie24. Jego zadaniem jest zapis danych z plików systemu magazynowego i kasowego sklepu, którego dane są wykorzystywane

w badaniach. Dane do badań odczytywane są bezpośrednio z bazy danych za pomocą odpowiedniej kwerendy SQL w programie *Rapid Miner*.

Baza danych zawiera łącznie ponad 470 tys. rekordów w sześciu tabelach. Najwięcej danych zgromadzonych jest w tabeli *TransakcjeInfoDodatkowe*, ponieważ znajdują się tam kody przypisane do identyfikatora transakcji. Ich liczba wynosi ponad 370 tys. rekordów dla całego sklepu oraz ponad 25 tys. rekordów dla Alma24. W tabeli *KartotekaTowarowa* zapisane zostały informacje o ponad 63 tys. kodów towarów Delikatesów Alma. W badanym okresie sklep zarejestrował ponad 39 tys. transakcji, z czego ponad 1000 należało do Almy24.

3. Proces ekstrakcji reguł asocjacyjnych

W programie Rapid Miner jakakolwiek analiza rozpoczyna się od utworzenia lub wczytania przygotowanego wcześniej odpowiedniego procesu[13]. Proces składa się z operatorów (tzw. klocków), które są dostępne w oknie *Operators*. Większość operatorów posiada indywidualne parametry, dzięki którym można kontrolować ich zachowanie w procesie. Pilotażowe badania przedstawione w pracy[14] wykazały, że proces poszukiwania reguł asocjacyjnych zużywa duże ilości pamięci operacyjnej oraz jest bardzo czasochłonny. Podstawowymi parametrami, które mają wpływ na wynik eksploracji, są: minimalna wartość wsparcia i minimalna wartość ufności. Oprócz tego najwięcej czasu zużywa wczytywanie i przetwarzanie danych z plików CSV. Dane pobrane należało dodatkowo przekształcić do postaci macierzy, która przekazywana była algorytmowi FP-Growth. Pierwszy problem rozwiązano, implementując bezpośrednio w programie *Rapid Miner* odczyt transakcji z bazy danych za pomocą kwerend SQL. Następnie dane przetwarza się do postaci macierzy użytej do wyszukiwania reguł przez algorytm FP-Growth. Dzięki takiemu zabiegowy proces działa prawie sześciokrotnie krócej. Dodatkowo użyte zostały operatory *Materialize Data* (zapisuje dane z pamięci operacyjnej) oraz *Free Memory* (oczyszcza pamięć operacyjną), które zmniejszyły zużycie pamięci operacyjnej. Na rysunku 4 przedstawiono proces utworzony na potrzeby poszukiwania reguł w danych transakcyjnych sklepu internetowego Alma24.

Po przetworzeniu i wyszukaniu pozycji częstych, tworzone są reguły asocjacyjne. Gdy proces zakończy swoje działanie, przełącza się automatycznie na widok rezultatów. Wynikiem są reguły asocjacyjne w postaci tabeli, którą można sortować według dostępnych kolumn, na przykład poziomu ufności. Oprócz tego możliwe jest wyświetlanie w zależności od kryterium (*min criterion*). Dostępny jest również

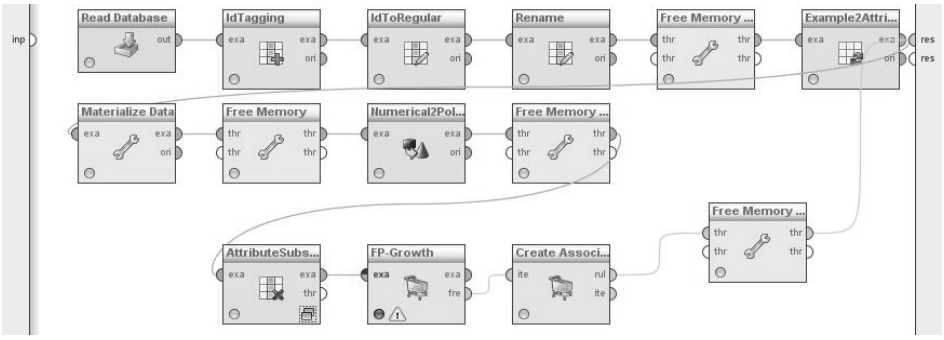
[13] M. Bereta: *Data Mining…*, *op. cit*.; *RapidMiner 4.3. User Guide. Operator Reference. Developer Tutorial* http://docs.huihoo.com/rapidminer/rapidminer-4.3-tutorial.pdf (czerwiec 2010).

[14] P. Skrzypczak: *Modelowanie wzorców…*, *op. cit*.

widok reguł w postaci grafu (opcja *Graph View*), jak również w postaci tekstu identycznego jak ten zapisywany do pliku przez operator *Write as Text*.

W badaniach przeprowadzonych w Almie24 wykorzystywano widok reguł w postaci tabeli uporządkowanej według kolumny *Confidence*. Po tej czynności część wyniku lub cały wynik można było wyeksportować do różnych formatów zewnętrznych, na przykład PDF.

Dane do analizy obejmują zakupy klientów Almy24 z okresu sierpień–wrzesień 2009. Transakcje klientów zawierają dane o zakupionych produktach, to jest kod towaru, ilość oraz cena detaliczna, wartość zakupów, informacje o posiadaniu karty Klubu Konesera (karty lojalnościowej), a także sposobie płatności (gotówka, karta, bon towarowy, przelew). Badania zostały przeprowadzone na poziomie kodu towaru dla klientów Almy24, których wartość koszyka zakupów jest większa niż 200 zł.



Rys. 4. Widok całego procesu

Źródło: opracowanie własne.

Minimalna wartość wsparcia została ustalona na poziomie 2%, natomiast minimalną wartość ufności ustalono na poziomie 80%. Wyniki badań przedstawiono częściowo na rysunku 5 i w tabeli 1. Zostało odnalezionych 31 reguł przy dwuprocentowym minimalnym wsparciu. Wśród interesujących reguł asocjacyjnych należy wyróżnić reguły z prawdopodobieństwem wystąpienia 100% (*confidence = 1*), między innymi:

Co najmniej 2% klientów kupujących pomidory na kg oraz ćwiartkę kurczaka zawsze kupuje również ziemniaki.

Co najmniej 2% klientów kupujących pomidory na kg oraz pomidory cirio w kawałkach zawsze kupuje również ziemniaki.

Premises	Conclusion	Confidence
produkt_pomidor kg, produkt_ćwiartka z kurczaka świeża	produkt_ziemniak młody kg	1
produkt_pomidor kg, produkt_pomidory cirio w kawalkach 400g	produkt_ziemniak młody kg	1
produkt_pomidor kg, produkt_mąka lubella 1kg poznańska pszenna	produkt_ziemniak młody kg	1
produkt_papryka czerwona kg, produkt_masło hajnówka 200g extra	produkt_pomidor kg	1
produkt_winogrono białe kg, produkt_arbuz kg	produkt_pomidor kg	1
produkt_banan chiquita kg, produkt_winogrono białe kg	produkt_ziemniak młody kg	1
produkt_ziemniak młody kg, produkt_mleko grajewo 3.2% 0.5l łaciate	produkt_poduzie z kurczaka	1
produkt_poduzie z kurczaka, produkt_mleko grajewo 3.2% 0.5l łaciate	produkt_ziemniak młody kg	1
produkt_jabłko kg, produkt_cebula czerwona kg	produkt_papryka czerwona kg	1
produkt_pomidor kg, produkt_banan chiquita kg, produkt_winogrono białe kg	produkt_ziemniak młody kg	1

Rys. 5. Widok części znalezionych reguł asocjacyjnych

Źródło: opracowanie własne.

Badania przeprowadzone zostały na transakcjach z okresu sierpień 2009 – styczeń 2010. Z uwagi na fakt, że dane transakcyjne dotyczyły dłuższego okresu niż poprzednio, zmniejszono wartość minimalnego wsparcia, tak aby możliwe było odnalezienie użytecznych dla sprzedaży reguł asocjacyjnych. W tabeli 1 przedstawiono wynik eksperymentu dla klientów z koszykiem zakupów większym niż 200 zł.

Tabela 1

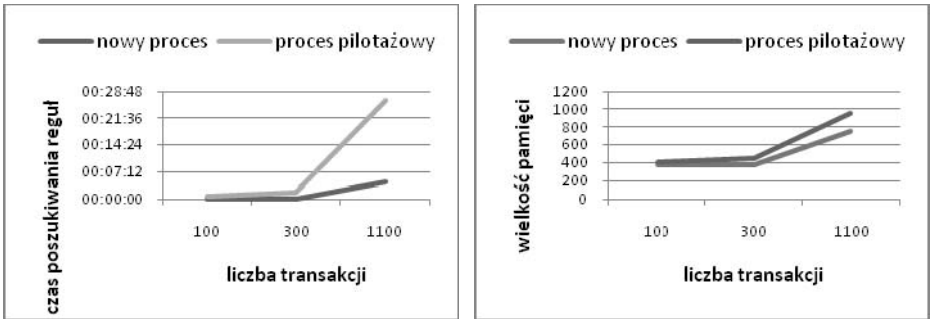
Wynik analizy kodów dla klientów Almy24 z koszykiem powyżej 200 zł

Liczba zbiorów kandydujących: 503	Liczba reguł asocjacyjnych: 7	
Minimalne wsparcie: 2%	Minimalna ufność: 80%	
Reguła		Ufność
marchew kg, seler kg	pietruszka kg	0,882
mandarynka kg, pietruszka kg	marchew kg	0,871
cebula kg, seler kg	pietruszka kg	0,861
pomidor kg, seler kg	pietruszka kg	0,846
cytryna kg, pietruszka kg	marchew kg	0,811
banan chiquita kg, seler kg	pietruszka kg	0,806
cebula kg, pietruszka kg	marchew kg	0,8

Źródło: opracowanie własne.

Wprawdzie badanie na większym zbiorze danych zmniejszyły liczbę odnalezionych reguł, jednak można było odnotować, że występują w obydwu przypadkach podobne reguły asocjacyjne, różniące się poziomem ufności. Liczba transakcji wpłynęła również na czas, jaki jest potrzebny na uzyskanie wyników (odpowiednio 16 s dla danych wykorzystanych w artykule oraz 87 s dla danych z półrocznych badań pilotażowych).

Przeprowadzany eksperyment miał również za zadanie sprawdzenie wydajności stworzonego procesu w odniesieniu do prototypu platformy. Rysunek 6 przedstawia wpływ całkowitej przebudowy procesu na wielkość pamięci oraz czas wykonywanego procesu.



Rys. 6. Wykres czasu poszukiwania reguł w zależności od liczby transakcji oraz wielkości zajmowanej pamięci RAM

Źródło: opracowanie własne.

Podsumowanie

W artykule przedstawiony został proces ekstrakcji reguł asocjacyjnych w transakcjach Delikatesów Internetowych Alma24. Początkowo do badań wykorzystywany był proces przygotowany i wykorzystany w badaniach pilotażowych. Po przeprowadzeniu kilkunastu eksperymentów okazało się, że proces jest mało efektywny oraz wykorzystuje duże ilości pamięci operacyjnej. Poza tym dane pobierane były z plików CSV, co skutkowało dodatkowym zużyciem pamięci oraz wydłużało cały eksperyment. Proces został zbudowany od nowa, dane importowane są za pomocą zapytań SQL bezpośrednio w programie *Rapid Miner*, w którym następnie są przetwarzane do postaci macierzy, wyszukiwane są pozycje częste, z których tworzone są reguły asocjacyjne. Dzięki temu cały proces trwa znacznie krócej, pomimo zastosowania tego samego algorytmu do wyszukiwania pozycji częstych i budowy reguł. Nowy proces zużywa również mniej pamięci, dzięki czemu możliwe jest przeprowadzenie eksperymentów na dużo większych zbiorach danych przy identycznej konfiguracji sprzętowej.

W artykule zaprezentowano, że nie tylko sam algorytm poszukiwania zbiorów częstych i reguł asocjacyjnych ma wpływ na ilość zużywanej pamięci oraz czas oczekiwania na wynik końcowy. Oczywiście ważne jest, aby w procesie eksploracji danych korzystać z wydajnych algorytmów poszukiwania reguł, jednak należy pamiętać również o procesie czyszczenia, konsolidacji i transformacji danych do postaci wykorzystywanej przez algorytm FP-Growth w programie *Rapid Miner*.

Podziękowanie: Autorzy dziękują Zarządowi spółki Delikatesy Alma we Wrocławiu za dostęp do danych firmowych i możliwość wykorzystania ich na potrzeby artykułu.

Literatura

1. Bereta M.: *Data Mining z wykorzystaniem programu RapidMiner*, http://michalbereta.pl/dydaktyka/ZSI/Lab%20Data%20Mining%201.pdf (czerwiec 2010); http://michalbereta.pl/dydaktyka/ZSI/Lab%20Data%20Mining%202.pdf (czerwiec 2010).
2. Bonchi F., Giannotti F., Mazzanti A., Pedreschi D.: *Exante: Anticipated data reduction in constrained pattern mining*, In Proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD '03), Cavtat–Dubrovnik 2003.
3. Borgelt C.: *Keeping Things Simple: Finding Frequent Item Sets by Recursive Elimination*, Workshop Open Source Data Mining Software (OSDM '05, Chicago, IL), ACM Press 2005.
4. Gyorödi C., Gyorödi R., Cofeey T., Holban S.: *Mining association rules using Dynamic FP-trees*, In proceedings of The Irish Signal and Systems Conference, University of Limerick 2003.
5. Han J., Pei J, Yin Y.: *Mining Frequent Patterns without Candidature Generation*, w: Proc. of the 200 ACM SIGMOD Int. Conf. on Management of Data, Dallas 2000.
6. Hand D., Mannila H., Smyth P.: *Eksploracja danych*, Wydawnictwo Naukowo-Techniczne, Warszawa 2005.
7. Morzy T.: *Eksploracja danych*, http://www.portalwiedzy.pan.pl/images/stories/pliki/publikacje/nauka/2007/03/N_307_06_Morzy.pdf (czerwiec 2010).
8. Pasztyła A.: *Analiza koszykowa danych transakcyjnych – cele i metody*, http://www.statsoft.pl/pdf/artykuly/basket.pdf (czerwiec 2010).
9. Rácz B.: *Nonordfp: An FP-Growth Variation without Rebuilding the FP-Tree*, 2nd Int'l Workshop on Frequent Itemset Mining Implementations FIMI 2004.
10. *RapidMiner 4.3. User Guide. Operator Reference. Developer Tutorial*, http://docs.huihoo.com/rapidminer/rapidminer-4.3-tutorial.pdf (czerwiec 2010).
11. Skrzypczak P.: *Modelowanie wzorców zachowań klientów Delikatesów Alma przy wykorzystaniu reguł asocjacyjnych*, Uniwersytet Ekonomiczny, Wrocław 2010.
12. Zaki M., Parthasarathy S., Ogihara M., Li W.: *New Algorithms for Fast Discovery of Association Rules. Proc.*, 3rd Int. Conf. on Knowledge Discovery and Data Mining (KDD '97), AAAI Press 1997.

FP-GROWTH ALGORITHM IN DISCOVERY OF CUSTOMER PURCHASING PATTERNS

Summary

In the paper, an algorithm FP-Growth and its variants are discussed. A new process of finding association rules using the Rapid Miner package has been proposed. The process was optimized in terms of memory usage and performance of the rule discovery process. The impact of proposed modifications made to the whole process of finding association rules has been evaluated. The experiments have been carried out on the internet database containing the customer transactions of the Delicatessen Alma24.

Translated by Jerzy Korczak