PRACE NAUKOWE UNIWERSYTETU EKONOMICZNEGO WE WROCŁAWIU nr 82
Informatyka Ekonomiczna 14 2009

Paweł Weichbroth
Politechnika Gdańska

ODKRYWANIE REGUŁ ASOCJACYJNYCH Z TRANSAKCYJNYCH BAZ DANYCH

Streszczenie: W metodologii drążenia danych ekstrakcja reguł asocjacyjnych z dużych baz danych jest popularną i wysoko rozwiniętą metodą odkrywania nieznanych związków pomiędzy zmiennymi. Zaprezentowany w niniejszej pracy algorytm Apriori jest przeznaczony do znalezienia powiązań pomiędzy produktami zarejestrowanymi przez systemy transakcyjne w sklepach wielkopowierzchniowych. Posiadanie takiej wiedzy może być z powodzeniem wykorzystane do zarządzania rozmieszczeniem produktów na półkach w sklepie, opracowania pakietów promocyjnych, sugerowania sprzedaży dodatkowej czy badań porównawczych sklepów.

Słowa kluczowe: reguły asocjacyjne, algorytm Apriori, drążenie danych, analiza koszyka zakupów.

1. Wstęp

Nauka wyodrębniania przydatnych informacji z dużych zbiorów danych (baz danych) jest znana jako drążenie danych (*data mining*). Jest to nowa interdyscyplinarna dziedzina nauki, którą można zakwalifikować do nauk matematycznych. Swoje podstawy czerpie z matematyki (statystyki) oraz informatyki (bazy danych, programowanie, przetwarzanie informacji). Wszystkie one związane są z określonymi aspektami analizy danych, jednak każda ma wyraźny kształt, podkreślając szczególne problemy i typy rozwiązań.

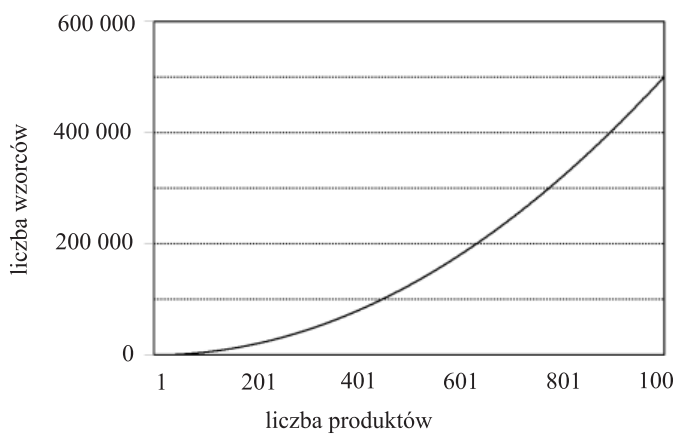
Ponieważ drążenie danych współgra z szerokimi typami tematów nauki o komputerach i statystyce, nie jest możliwe, aby wymienić pełny materiał w tej pracy. Biorąc to pod uwagę, autor skupił się na zagadnieniach fundamentalnych. W celu zrozumienia istoty reguł asocjacyjnych (*association rules*) potrzebna jest znajomość rachunku prawdopodobieństwa, algebry liniowej, optymalizacji i estymacji parametrów.

Dostępność szczegółowych informacji na temat transakcji klienta doprowadziła do rozwoju technik umożliwiających wyszukiwanie asocjacji pomiędzy elementami zapisanymi w bazie danych. Za przykład może posłużyć baza danych sklepu, który korzysta ze skanerów kodów kreskowych. W tym przypadku będzie to tzw. analiza koszykowa (*market basket analysis*), złożona z dużej liczby transakcji. Każda trans-

akcja reprezentuje jednego anonimowego klienta. Kierownicy sklepów, analitycy sprzedaży byliby zainteresowani tym, czy pewne grupy produktów konsekwentnie są nabywane razem. Informacje tego typu mogłyby być z powodzeniem wykorzystane do odpowiedniego rozmieszczenia na półkach sklepowych, organizowania promocji zbiorowych, projektowania katalogów promocyjnych oraz utworzenia profilów klientów.

W zbiorze danych opisujących klientów supermarketu wzorcem może być: „10% klientów, którzy kupili wino, kupuje ser", firmy telekomunikacyjnej zaś: „jeżeli alarmy A i B zdarzają się w ciągu 30 sekund dla każdego z nich, to alarm C zdarzy się w ciągu 60 sekund z prawdopodobieństwem 0,5", z kolei w logach odwiedzin strony internetowej wzorzec możemy określić następująco: „jeżeli osoba odwiedzi stronę Onet.pl, jest 60% szansy, iż osoba ta odwiedzi stronę Poczta.onet.pl". W każdym z tych przypadków wzorcem jest potencjalnie interesujący „kawałek" zależności o części ze zbioru danych.

Jakie są sposoby znalezienia wzorców w bazie danych? W wysoko jednorodnych, małych zbiorach danych, o niskiej złożoności informacji, stosowana jest **metoda trywialna** (*trivial method*), polegająca na znalezieniu wszystkich możliwych kombinacji. Wyżej opisane właściwości zbioru danych powodują, iż zastosowanie tej metody jest zadowalające – także dla z góry znanych zależności lub dla małej liczby wzorców. Jednak w miarę wzrostu liczby produktów liczba wzorców wzrasta kwadratowo (rys. 1). Może się to okazać czasochłonne i nieefektywne pod kątem interpretacji ekonomicznej. Na przykład dla bazy danych transakcji 1000 produktów będzie to 499,5 tys. wzorców[1]. W przypadku obrazów czy sekwencji alarmów potencjalnie liczba możliwych wzorców jest nieskończona.



Rys. 1. Efektywność metody trywialnej

Źródło: opracowanie własne.

[1] Policzone na podstawie wzoru (1); wzory znajdują się na końcu artykułu.

Jeżeli wzorce byłyby kompletnie niepowiązane ze sobą, nie byłoby wyboru poza trywialną metodą. Często zbiór wzorców ma dużą liczbę struktur i należy skorzystać z niej w celu zdefiniowania poszukiwań. Zazwyczaj istnieje ogólna relacja pomiędzy wzorcami. Wzorzec α jest bardziej ogólnikowy niż wzorzec β, jeżeli kiedykolwiek w zbiorze danych pojawia się β, występuje także α. Na przykład wzorzec „Co najmniej 10% klientów kupuje wino" jest w większym stopniu ogólny niż wzorzec „Przynajmniej 5% klientów kupuje wino i ser". Użycie takiego uogólnienia relacji pomiędzy wzorcami prowadzi do prostego algorytmu znalezienia wszystkich wzorców określonego typu w bazie danych.

2. Definicja reguły asocjacyjnej

Reguła asocjacyjna dostarcza informacji w formie stwierdzenia „jeżeli – to" (*if – then*), które dzieli się na dwie części: poprzednika „jeżeli" (*antecedent*) oraz następnika (*consequent*) – „to". W przeciwieństwie do reguły logicznej ma cechy probabilistyczne. Reguła jest policzalna w postaci dwóch mierników, wyrażających stopień niepewności o regule.

Pierwszym miernikiem jest **wsparcie** reguły (*support*), które jest liczbą transakcji zawierających jednocześnie poprzednik i następnik. W pracy wykorzystano także **współczynnik wsparcia** (*support ratio*), dany wzorem:

$$\text{support}\{\text{A} \rightarrow \text{B}\} = \frac{N_{A \rightarrow B}}{N}.$$

Drugim miernikiem jest **współczynnik ufności** (*confidence ratio*), będący stosunkiem liczby transakcji zawierających wszystkie transakcje poprzednika i następnika do liczby transakcji zawierających poprzednika, dany wzorem:

$$\text{confidence}\{\text{A} \rightarrow \text{B}\} = \frac{N_{A \rightarrow B}}{N_A} = \frac{\frac{N_{A \rightarrow B}}{N}}{\frac{N_A}{N}} = \frac{\text{support}\{\text{A} \rightarrow \text{B}\}}{\text{support}\{\text{A}\}}.$$

Na przykład jeżeli baza danych supermarketu ma zapisanych 100 000 transakcji, z których 2000 zawiera produkty A i B oraz 800 z nich posiada także produkt C, reguła asocjacyjna „Jeżeli A i B zostały kupione, to produkt C wystąpi w tej samej transakcji" ma współczynnik wsparcia równy 0,8% (800/10000) i współczynnik zaufania 40% (800/2000). Niski poziom wsparcia (jedna transakcja na 100 tys.) świadczy o tym, że określona reguła jest nieistotna lub w danych występują błędy (zob. [Weichbroth, Korczak 2006]). Współczynnik wsparcia jest prawdopodobieństwem zdarzenia, iż losowo wybrana transakcja z bazy danych będzie zawierać poprzednik i następnik, współczynnik zaufania zaś jest warunkowym prawdopodobieństwem, iż przypadkowo wybrana transakcja będzie zawierała wszystkie z góry określone elementy następnika w stosunku do poprzednika.

3. Generowanie reguł asocjacyjnych ze zbioru danych

Menedżger sklepu ABC chciałby wiedzieć, które produkty są sprzedawane razem i jak silna jest zależność między nimi. Pozwoli mu to na przygotowanie ulotki promocyjnej (*retail promotion leaflet*), gdzie znajdą się produkty o najwyższym powiązaniu. Tabela 1 jest uproszczonym modelem transakcji sprzedaży wykorzystanym do analizy.

Tabela 1. Uproszczony model transakcji sprzedaży sklepu ABC

Id transakcji	Kod produktu			
1	1	2	5	
2	2	4		
3	2	3		
4	1	2	4	
5	1	3		
6	2	3		
7	1	3		
8	1	2	3	5
9	1	2	3	

Źródło: [Patel 2003].

Pokazanych zostało 9 transakcji sprzedaży, z których każda jest zapisem zakupionych produktów (np. w transakcji pierwszej anonimowy klient zakupił produkty o kodach 1, 2, 5). Załóżmy, iż będą brane pod uwagę wyłącznie reguły asocjacyjne, mające wsparcie co najmniej 2 (współczynnik wsparcia 22%, 2/9). W ten sposób przyjęty został p**róg odcięcia** (*cut-off*), który określany jest arbitralnie, ze względu na przedmiot badania i subiektywną ocenę poziomu istotności parametrów badania. Tabela 2 pokazuje wsparcie dla „częstych" zbiorów produktów, policzonych na podstawie danych z tab. 1.

Tabela 2. Wsparcie dla reguł asocjacyjnych

Zbiór	Wsparcie	Zbiór	Wsparcie
{1}	6	{1, 3}	4
{2}	7	{1, 5}	2
{3}	6	{2, 3}	4
{4}	2	{2, 4}	2
{5}	2	{2, 5}	2
{1, 2}	4	{1, 2, 3}	2
{1, 2, 5}	2		

Źródło: opracowanie własne na podstawie [Patel 2003].

Raz utworzona lista wszystkich zbiorów, mających wymagane wsparcie, pozwala wyselekcjonować te reguły, które mają pożądaną wielkość zaufania. Każdy podzbiór ze zbioru musi wystąpić co najmniej tyle razy, ile razy występuje cały zbiór, a także znajdować się na liście. Na przykład zbiór {1, 2, 5} można podzielić na trzy podzbiory {1}, {2} i {5}, które odpowiednio mają wsparcie 6, 7 i 2. Cały zbiór {1, 2, 5} ma wsparcie 2 – tym samym zachowany jest opisany powyżej warunek. Tabela 3 przedstawia współczynniki ufności dla możliwych reguł asocjacyjnych zbioru {1, 2, 5}.

Tabela 3. Reguły asocjacyjne i ich współczynniki zaufania

Reguła	Współczynnik ufności
{1, 2} => {5}	50% (2/4)
{1, 5} => {2}	**100%** (2/2)
{2, 5} => {1}	**100%** (2/2)
{1} => {2, 5}	33% (2/6)
{2} => {1, 5}	29% (2/7)
{5} => {1, 2}	**100%** (2/2)

Źródło: opracowanie własne.

W przypadku progu odcięcia 80% dla współczynnika ufności tylko druga, trzecia i ostatnia reguła miałyby znaczenie.

Przy z góry przyjętych założeniach co do wsparcia i zaufania dla reguł asocjacyjnych problem znalezienia takich reguł został podzielony na dwa etapy. W pierwszym zostały znalezione wszystkie zbiory z wymaganym wsparciem (określane jako „częste" lub „duże"), w drugim zaś kroku zostały wyodrębnione reguły asocjacyjne, które mają założoną wielkość współczynnika zaufania.

4. Algorytm AprioriAll

Algorytm AprioriAll przedstawiony przez Agrawala i Srikanta (zob. [Agrawal, Srikant 1994]) jest rozwiązaniem problemu poszukiwania sekwencyjnych wzorców w dużych bazach danych. **Wzorzec** (sekwencja) rozumiany jest jako uporządkowany zbiór transakcji każdego z klientów.

Głównym założeniem leżącym u podstaw algorytmu jest znalezienie wszystkich wspólnych wzorów transakcji klientów zapisanych w bazie danych. Opisany problem jest podobny do problemu poszukiwania rzędu słów, które odpowiadają danemu regularnemu wyrażeniu (jak w przypadku linuksowego narzędzia *grep*). Trudnością jest wybór wzorców i efektywny proces wyszukiwania, które znajdują się w sekwencjach transakcji klientów.

Algorytm ma zastosowanie w bazach danych liczących miliony rekordów (transakcji sprzedaży). Składa się z pięciu faz, tj.: sortowania, tworzenia zbiorów, transfor-

macji, szukania sekwencji i wydzielania maksimów. Pierwsza faza polega na posortowaniu bazy danych $Đ$ na podstawie klucza głównego *id-klienta* oraz klucza *czas-transakcji*. W tej fazie dokonuje się konwersja pierwotnej transakcyjnej bazy danych w bazę danych sekwencji transakcji klientów. Faza druga obejmuje znalezienie zbioru L wszystkich długich sekwencji transakcji $l_i \{<l> \mid l \epsilon L\}$, o zdefiniowanej z góry częstości wystąpień. W kolejnej fazie każda sekwencja klienta podlega transformacji w alternatywną reprezentację. W zmienionej sekwencji klienta każda transakcja zastępowana jest przez zbiór l_i występujący w tej transakcji. Jeżeli transakcja nie ma żadnego zbioru l_i, nie pozostaje dalej w zmienionej sekwencji, również jeżeli sekwencja klienta nie zawiera żadnego zbioru l_i, jest usuwana z bazy danych. Po tej transformacji sekwencja klienta jest reprezentowana przez zbiór l_i, bazę danych zaś oznaczymy przez $Đ_T$. Celem czwartej fazy jest wyszukanie pożądanych sekwencji w zbiorze bazy danych $Đ_T$. Algorytm wielokrotnie przeszukuje dane i w każdym przejściu generuje potencjalnie nową dużą sekwencję, nazywaną sekwencją kandydującą. Dla każdej sekwencji liczona jest liczba wystąpień, w ostatnim zaś przejściu pozostają tylko sekwencje o największej liczbie wystąpień. To te sekwencje kandydatów są punktem wyjścia dla następnej iteracji przeszukiwania bazy danych. Ostatnim etapem jest znalezienie wszystkich kandydatów o największej częstości.

W konstrukcji algorytmu dwa zagadnienia pozostają do rozwiązania: jak kandydaci są formowani oraz jak zostanie policzona liczba wystąpień sekwencji w danych. W celu utworzenia kandydatów z kolekcji zbiorów L_i należy znaleźć wszystkie zbiory Y o wielkości $i+1$ przez utworzenie wszystkich par $\{U, V\}$ w zbiorach należących do L_i, takich, dla których połączenie U i V ma rozmiar co najmniej rzędu $i+1$. Następnie testowane jest, czy połączenie U i V jest potencjalnym kandydatem w zbiorze danych.

Wygenerowanie „częstych" jednoelementowych zbiorów nie jest zadaniem trudnym, gdyż wystarczy przeliczyć wszystkie transakcje znajdujące się w bazie danych. Otrzymane liczby to wsparcie dla tych właśnie zbiorów. Dla założonego progu wsparcia (*minimum support*) odrzucamy te, które go nie spełniają – tym sposobem mamy pierwszą listę kandydatów. Na ich podstawie algorytm generuje zbiory kandydujące (*candidate itemsets*) przez wykonanie operacji łączenia dwóch zbiorów jednoelementowych w jeden dwuelementowy, które potencjalnie mogą być częste. Dla każdej pary zostaje policzone wsparcie w bazie danych D – jeżeli wynosi co najmniej założony próg, para taka jest dołączana do zbiorów częstych i w kolejnym kroku zostanie wykorzystana do wygenerowania zbiorów kandydujących trzyelementowych. Kolejne kroki są iteracyjne, tj. „częste" zbiory trzyelementowe zostaną wykorzystane do stworzenia zbiorów kandydujących czteroelementowych, „częste" zbiory czteroelementowe do utworzenia zbiorów kandydujących pięcioelementowych itd. Każdy kolejny krok działania algorytmu generuje zbiory kandydujące o rozmiarze większym o 1 w stosunku do poprzedniego. Skutkiem tego są kolejne odczyty z bazy danych w celu obliczenia wsparcia dla kolejnych zbiorów. Algorytm kończy działanie w przypadku braku możliwości wygenerowania kolejnych zbiorów

kandydujących. Końcowym wynikiem jest suma *k*-elementowych „częstych" zbiorów ($k = 1, 2, 3, \ldots$).

Z każdego zbioru L_i można wygenerować mniej niż $|L_i|^2$ par zbiorów. Obecność każdej pary w zbiorze jest weryfikowana, co stanowi duże obciążenie czasu pracy algorytmu. W praktyce przedstawiona metoda zwykle działa w liniowym czasie, w zależności od rozmiaru L_i, z kolei formowanie kandydatów jest niezależne od liczby rekordów w bazie danych. Poniżej został zamieszczony pseudokod opisanego powyżej algorytmu AprioriAll.

```
/*
Zmienne
Ck - zbiór wszystkich k-sekwencji kandydatów
Lk - zbór wszystkich długich k-sekwencji
*/
insert into Ck
select p.sekwencja1, ... , p.sekwencjak-1, q.sekwencjak-1
from Lk-1p, Lk-1q
where  p.sekwencja1 = q.sekwnecja1, ...
       p.sekwencjak-2 = q.sekwencjak-2;
if sekwencjak-1 є c is not in Lk-1
delete wszystkie sekwencje c є Ck
for ( k = 2; Lk-1 ≠ 0; k++) do
       begin
       Ck = nowi kandydaci wygenerowani z Lk-1
       for each sekwencji-klienta c w bazie danych do
       zwiększ licznik wszystkich kandydatów w Ck є c
       Lk = kandydaci w Ck z najmniejszą liczbą wystąpień
       end
/*
```

Wynikiem działania algorytmu są **największe sekwencje** znalezione w **zbiorze $\mathbf{L_k}$**.

Przedstawiony algorytm AprioriAll generuje wszystkie możliwe reguły asocjacyjne ze zbioru danych, które spełniają trzy warunki (zob. [Maloof 2006]): 1) **liczby zmiennych** wykorzystanych do utworzenia reguły, 2) **poziomu wparcia** i 3) **poziomu ufności** dla reguły. W swojej konstrukcji nie obsługuje ograniczeń czasowych, „ruchomych okien" oraz taksonomii. W zastosowaniach algorytmu można spotkać się z dwoma problemami. Pierwszy z nich dotyczy natury zasobów obliczeniowych w procesie dokonania transformacji danych, jaka ma miejsce w trakcie każdego przejścia algorytmu po bazie danych. Alternatywnie można przechowywać zmienioną bazę danych, jednak często mogłoby się okazać, iż jest to niewykonalne dla aplikacji, ze względu na dwukrotnie wyższy rozmiar danych i niższą wydajność bazy danych. Po drugie, kiedy stałoby się możliwe, aby rozszerzyć algorytm do korzystania z ograniczeń czasowych i taksonomii, nie dałoby się włączyć „ruchomych okien".

5. Podsumowanie

W przedmiocie drążenia danych reguły asocjacyjne są popularną metodą odkrywania interesujących relacji pomiędzy zmiennymi w bazach danych. Ich eksploracja ze zbioru danych jest rozwiązaniem problemu analizy koszyka zakupów, powszechnie stosowanego przez sklepy wielkopowierzchniowe. Przedstawiony w pracy algorytm ApriorіAll jest kamieniem milowym w rozwoju nauki o eksploracji danych – był pierwszym, który wykorzystał własność monotoniczności miary wsparcia do ograniczenia przestrzeni poszukiwań „częstych" zbiorów. Znalezione reguły, mające sens logiczny, mogą znacznie przyczynić się do zwiększenia efektywności akcji promocyjnych i co za tym idzie – wielkości sprzedaży.

Wykorzystane wzory

1) Kombinacjami z n elementów po k nazywamy zbiory k-elementowe, które można utworzyć, wybierając k dowolnych przedmiotów spośród n danych przedmiotów, przy czym uporządkowanie nie odgrywa roli. Kombinacje te mogą różnić się tylko elementami. Liczba wszystkich kombinacji z n różnych elementów po k wyraża się wzorem:

$$C_n^k = \binom{n}{k} = \frac{n!}{k!(n-k)!}. \tag{1}$$

2) **Współczynnik wsparcia (*support ratio*)**: niech $N_{A\to B}$ będzie liczbą sekwencji w postaci A→B, a N to ogólna liczba transakcji. Współczynnik wsparcia jest ilorazem liczby sekwencji do liczby transakcji ogółem: $\text{support}\{\text{A} \to \text{B}\} = \frac{N_{A\to B}}{N}$.

3) **Współczynnik ufności (*confidence ratio*):** niech A→B będzie regułą typu „jeżeli A, to B", wówczas współczynnik ufności jest ilorazem wsparcia reguły A→B do wsparcia dla zmiennej A:

$$\text{confidence}\{\text{A} \to \text{B}\} = \frac{N_{A\to B}}{N_A} = \frac{\frac{N_{A\to B}}{N}}{\frac{N_A}{N}} = \frac{\text{support}\{\text{A} \to \text{B}\}}{\text{support}\{\text{A}\}}.$$

Literatura

Agrawal R., Srikant R., *Fast Algorithms for Mining Association Rules*, [w:] *Proceedings of the Twentieth International Conference on Very Large Data Bases*, Morgan Kaufmann, San Francisco, CA 1994 , s. 487-499.

Agrawal R., Srikant R., *Mining Sequential Patterns*, [w:] *Proceedings of the 11th ICDE Conference*, red. P.S. Yu, L.P. Arbee Chen, IEEE Computer Society, Taiwan 1995.

Bronsztejn I.N., Siemiediajew K.A., *Matematyka. Poradnik encyklopedyczny*, Wydawnictwo Naukowe PWN, Warszawa 2003, s. 206.

Maloof M., *Some Basic Concepts of Machine Learning and Data Mining*, [w:] *Machine learning and Data Mining for Computer Security*, red. Marcus A. Maloof, Springer, London 2006, s. 32.

Patel N., *15.062 Data Mining, Spring 2003*, http://ocw.mit.edu/OcwWeb/Sloan-School-of-Management/15-062Data-MiningSpring2003/CourseHome/index.htm.

Weichbroth P., Korczak J., *Data mining – drążenie danych*, [w:] *Informatyka ekonomiczna. Cześć I*, *Propedeutyka informatyki. Technologie informacyjne*, red. Jerzy Korczak, Wrocław 2006, s. 265.

DISCOVERING ASSOCIATION RULES IN TRANSACTION DATABASES

Summary: In the data mining methodology, extracting association rules from large databases is a popular and well researched method for discovering interesting relations between variables. The presented AprioriAll algorithm discovers regularities between products in large scale transaction databases, recorded by point of sales systems in supermarkets. Such information might be useful and be used as the basis for decisions regarding marketing activities, such as retail promotion pricing, leaflets or product placement.