Analiza asocjacji i reguły asocjacyjne w badaniu wyborów zajęć dydaktycznych dokonywanych przez studentów. Zastosowanie algorytmu Apriori

Mirosława Lasek[*], Marek Pęczkowski[*]

Streszczenie

W artykule przedstawiono możliwości i korzyści zastosowania metody analizy asocjacji, zaliczanej do metod eksploracji danych (ang. Data Mining), w zagadnieniach dotyczących wyborów przez studentów nieobowiązkowych zajęć dydaktycznych. Krótko opisano metodę analizy asocjacji i budowy reguł asocjacyjnych, szczególną uwagę poświęcając algorytmowi Apriori, jednemu z najpopularniejszych algorytmów analizy asocjacji i budowy reguł asocjacyjnych, w sposób uporządkowany i logiczny realizującemu potrzebne działania oraz przystępnie, przejrzyście i zrozumiale obrazującemu ideę analizy asocjacji i generowanie reguł asocjacyjnych. Rozważania zilustrowano przykładem wyboru przez studentów zajęć, spośród prowadzonych na Wydziale Nauk Ekonomicznych Uniwersytetu Warszawskiego we współpracy z SAS Institute Polska, w ramach ich cyklu o nazwie „Data Mining Certificate Program". Wskazano, wraz z krótką wzmianką o jego działaniu, oprogramowanie wspomagające przeprowadzanie analizy asocjacji, tworzenie reguł asocjacyjnych i interpretację uzyskiwanych wyników – program SAS Enterprise Miner firmy SAS Institute Inc. z USA, wykorzystywany przez nas w zaprezentowanym w artykule zagadnieniu wyboru zajęć dydaktycznych przez studentów.

Słowa kluczowe: analiza asocjacji, reguły asocjacyjne, eksploracja danych, algorytm Apriori, program Enterprise Miner firmy SAS, wybór zajęć dydaktycznych przez studentów szkół wyższych.
JEL Code: C60.

[*] Katedra Informatyki Gospodarczej i Analiz Ekonomicznych, Wydział Nauk Ekonomicznych, Uniwersytet Warszawski, ul. Długa 44/50, 00-241 Warszawa, e-mail: mlasek@wne.uw.edu.pl; mpeczkowski@wne.uw.edu.pl

Wstęp

Analiza asocjacji wraz z budowaniem reguł asocjacyjnych jest metodą zaliczaną do metod eksploracji danych (ang. Data Mining), pod pojęciem których rozumie się metody statystyczne i metody sztucznej inteligencji umożliwiające odkrywanie nieznanych zależności między danymi w nagromadzonych zbiorach danych [Berry, Linoff, 2004; Lasek, 2004; Lasek, Pęczkowski 2013]. Są to metody, które pozwalają z danych tworzyć wiedzę, tzn. znajdywać zależności, wzorce, trendy „ukryte" w danych.

Odkrywanie asocjacji i budowanie reguł asocjacyjnych służy poszukiwaniu i odnajdywaniu związków między obiektami lub grupami obiektów opisanych przez wiele cech ilościowych lub jakościowych [Larose, 2006].

Celem artykułu jest zbadanie przydatności zastosowania analizy asocjacji i reguł asocjacyjnych do znajdywania prawidłowości i zasad postępowania, jakimi kierują się studenci, wybierając zajęcia, aby brać w nich udział, spośród zajęć (przedmiotów), w których mogą uczestniczyć lub mogą nie uczestniczyć w zależności od ich indywidualnych preferencji.

Analiza asocjacji i reguły asocjacyjne pozwalają wnioskować o tym na ile pewne powtarzające się wybory można uznać nie za przypadkowe, ale przypisać im miano wyróżniających się spośród innych, sygnalizujących pewną przyczynowość i traktować jako bardziej prawdopodobne niż inne. Pozyskanie takich informacji o wyborach zajęć przez studentów może być z pewnością przydatne dla planowania harmonogramów zajęć, a tu chociażby przewidywania liczby grup studentów, w zależności od liczby studentów, którzy zgłoszą akces uczestnictwa w zajęciach danego rodzaju, rezerwacji potrzebnych sal, w których będą odbywać się zajęcia, zagwarantowania dyspozycyjności osób, które będą prowadzić zajęcia.

1. Algorytm Apriori w analizie asocjacji

1.1. Analiza asocjacji. Reguły asocjacyjne

Analiza asocjacji polega na identyfikacji współzależności cech. Umożliwia wykrycie logicznych reguł występujących między obiektami w zbiorze danych. Analiza polega na identyfikacji pozycji, które często występują razem. Dzięki odkryciu tej współzależności możemy stworzyć tzw. reguły asocjacyjne.

Reguły asocjacyjne mają postać implikacji: jeżeli [poprzednik], to [następnik], co możemy zapisać: jeżeli A, to B, gdzie A oznacza poprzednik, a B następnik i oznaczać symbolicznie $A \rightarrow B$ [D.T. Larose, 2006].

Od logicznych implikacji, reguły asocjacyjne różnią się tym, że jeżeli zajdzie zdarzenie A, to zdarzenie B nie musi wystąpić z pewnością, a jedynie z pewnym prawdopodobieństwem (np. 90%). Przykładem reguł asocjacyjnych są stwierdze-

nia: „98% kupujących opony i akcesoria samochodowe oddaje także samochód do serwisu" [Aher, Lobo, 2012], „w 90% transakcji, w których kupiono chleb i masło, kupiono też mleko" [Agrawal, Srikant, 1994].

Wyszukiwanie reguł asocjacyjnych jest jednym z podstawowych metod odkrywania wiedzy. Reguły asocjacyjne znalazły zastosowanie w wielu rozmaitych dziedzinach, np. analizie koszykowej (odkrywanie wzorców zachowań klientów, co pozwala lepiej/efektywniej rozmieścić towary w sklepie, zaprojektować katalogi produktów, zachęcić klientów do zakupu dodatkowego artykułu), diagnozowaniu awarii w sieciach komunikacyjnych, prowadzeniu akcji marketingowych, działalności ubezpieczeniowej, bankowości (por. np. [Berry, Linoff, 2004; Lasek, Pęczkowski, 2013]).

1.2. Podstawowe pojęcia metody analizy asocjacji

W przedstawianych objaśnieniach pojęć przyjęto oznaczenia używane powszechnie w opisach metody analizy asocjacji – także w polskiej literaturze (np. [Larose, 2006; Lasek, Nowak, Pęczkowski, 2008; Lasek, Pęczkowski, 2013]

Niech $I = \{i_1, i_2, \ldots, i_m\}$ oznacza zbiór składający się z *m* elementów. W analizie koszykowej jest to zbiór towarów możliwych do zakupu w supermarkecie.

Każdy podzbiór T_j zbioru I ($T_j \subset I$) nazywamy transakcją. W analizie koszykowej transakcja T_j stanowi zbiór towarów zakupionych przez *j*-tego klienta (tzw. koszyk).

Bazą transakcji jest zbiór par postaci (id_j, T_j), gdzie id_j jest identyfikatorem transakcji, T_j jest transakcją, czyli np. zbiorem towarów zakupionych przez klienta o identyfikatorze id_j. Wówczas regułę asocjacyjną można formalnie zapisać jako implikację $A \rightarrow B$, gdzie $A \subset I$, $B \subset I$ oraz $A \cap B = \emptyset$, tzn. A i B są transakcjami nie zawierającymi wspólnych elementów.

Jakość reguły asocjacyjnej można mierzyć na podstawie danych zawartych w konkretnej bazie transakcji. Zdefiniujemy trzy wskaźniki [Lasek, Nowak, Pęczkowski, 2008]:

(a) wsparcie reguły (ang. support (supp))

$$\text{supp}(A \rightarrow B) = \frac{n(A \cap B)}{N} = P(A \cap B)$$

gdzie N – liczba wszystkich transakcji, $n(A \cap B)$ – liczba transakcji zawierających jednocześnie elementy transakcji A i transakcji B, $P(A \cap B)$ prawdopodobieństwo, że transakcja zawiera jednocześnie A i B.

Jeżeli transakcja pasuje do reguły, tzn. spełnione są warunki poprzednika i następnika, to mówimy, że reguła zawiera określoną transakcję (transakcja wspiera określoną regułę asocjacyjną).

(b) ufność reguły (ang. confidence (conf))

$$\text{conf}(A \rightarrow B) = \frac{n(A \cap B)}{n(A)} = P(B / A)$$

(c) przyrost (ang. lift)

$$lift(A \frac{conf(A \rightarrow B)}{P(B)} = \frac{P(B \mid A)}{P(B)}$$

gdzie $P(B \mid A)$ oznacza prawdopodobieństwo warunkowe

Rozważmy regułę asocjacyjną i jej charakterystyki – wsparcie, ufność, przyrost – [Buty] → [Skarpety] ([Lasek, Pęczkowski, 2013]).

Zawarto N = 1 000 000 transakcji, z czego 200 000 - n(buty), 50 000 – n(skarpety), 20 000 – n(buty & skarpety). Możemy obliczyć:

$$wsparcie = \frac{n(buty \ \& \ skarpety)}{N} = \frac{20000}{1000000} = 2\%$$

$$ufność = \frac{n(buty \ \& \ skarpety)}{n(buty)} = \frac{20000}{200000} = 10\%$$

$$przyrost = \frac{ufność}{P(skarpety)} = \frac{10\%}{5\%} = 2,$$

gdzie

$$P(skarpety) = \frac{n(skarpety)}{N} = \frac{50000}{1000000} = 5\%$$

Wsparcie 2% oznacza, że wśród badanych transakcji poprzednik i następnik występują razem w dwóch procentach, a ufność 10% oznacza, że w 10% występowania poprzednika występuje również następnik.

Interesujące są reguły, w których zarówno wsparcie, jak i ufność przyjmują w miarę duże wartości. Mówimy, że reguła asocjacyjna jest mocna (silna), jeżeli jej wsparcie i ufność są większe niż pewne ustalone wartości minimalne:

supp(A → B) > minSupp,

conf(A → B) > minConf,

gdzie parametry minSupp i minConf są ustalone przez użytkownika programu komputerowego lub eksperta z danej dziedziny, w zależności od rodzaju problemu.

Ważnym pojęciem jest częstość zbioru transakcji. W danej bazie transakcji częstość zbioru transakcji A jest to liczba transakcji zawierających dany zbiór: n(A). Zbiór A jest określany jako częsty, gdy występuje w transakcjach przynaj-

mniej pewną ustaloną minimalną liczbę razy Φ (np. gdy przyjmiemy Φ = 4, oznacza to, że dany zestaw elementów ze zbioru I występuje przynajmniej w czterech transakcjach).

Przyjmijmy, że elementy zbioru I są uporządkowane (można utożsamiać je z kolejnymi liczbami naturalnymi albo uporządkować leksykograficznie). Mamy zatem

$$i_1 < i_2 < \ldots < i_m.$$

To uporządkowanie przenosi się na podzbiory zbioru I, czyli transakcje.

Zajmijmy się teraz algorytmem generowania reguł asocjacyjnych. Liczba możliwych do utworzenia reguł asocjacyjnych nawet dla małolicznego zbioru I jest bardzo duża. Nie można więc po prostu wygenerować wszystkich reguł asocjacyjnych, a dopiero potem wybrać „najlepsze".

1.3. Algorytm Apriori generowania reguł asocjacyjnych

Jednym z najpopularniejszych algorytmów generowania reguł asocjacyjnych jest algorytm Apriori zaproponowany w pracy [Agrawal, Srikant, 1994], choć po raz pierwszy problem odkrywania reguł asocjacyjnych był już rozpatrywany rok wcześniej w [Agrawal, Imieliński, Swami, 1993].

Algorytm Apriori składa się z dwóch podstawowych etapów (po ustaleniu minimalnych wartości wsparcia i ufności, tj. parametrów minSupp i minConf). Te dwa etapy, to [Larose, 2006; Morzy, 2013; Nguyen Sinh Hoa, online, dostęp: 28 grudnia 2013]:

1) generowanie zbiorów częstych na podstawie parametru minSupp,,

2) na podstawie utworzonych zbiorów częstych budowanie reguł o ufności większej niż minConf.

Poważniejszym i trudniejszym niż etap drugi, jest etap pierwszy – wygenerowanie wszystkich zbiorów częstych. W definicji zbioru częstego nie musimy określać, które elementy należą do poprzednika, a które do następnika reguły asocjacyjnej. Mamy po prostu rodzinę pewnych podzbiorów zbioru I.

1.4. Generowanie zbiorów częstych algorytmu Apriori

Kluczową właściwością wykorzystywaną przy generowaniu zbiorów częstych jest tzw. własność Apriori, z której wynika, że [Larose, 2006]:

każdy podzbiór zbioru częstego jest zbiorem częstym, albo inaczej: **jeżeli jakiś zbiór nie jest zbiorem częstym, to jego nadzbiór też nie jest zbiorem częstym**.

Własność Apriori ułatwia przeszukiwanie zbiorów, ponieważ jeżeli jakiś zbiór A nie jest częsty, to możemy pominąć w rozważaniach wszystkie jego nadzbiory, tzn. zbiory X, takie że $A \subset X$. Opisywany fakt ilustruje rysunek 1 przedstawiony w opisie przykładu w niniejszym artykule.

Szukanie zbiorów częstych jest algorytmem iteracyjnym. W każdym kroku zwiększa się liczba elementów zbioru.

Oznaczmy:

C_k – zbiór transakcji *k*-elementowych,

L_k – zbiór transakcji częstych *k*-elementowych.

Algorytm zaczyna się od znalezienia wszystkich zbiorów częstych jednoelementowych L_1. Następnie L_1 jest wykorzystywany do tworzenia zbiorów częstych dwuelementowych L_2 i tak dalej, aż do momentu stwierdzenia, że dla pewnych *k* nie ma już częstych zbiorów *k*-elementowych.

W etapie szukania zbiorów częstych istotne są dwie główne operacje:

1) łączenie (ang. *join*),

2) przycinanie (ang. *prune*).

Operacja łączenia polega na realizacji opisanych poniżej czynności.

Mając ustalony zbiór L_{k-1} do zbioru C_k wstawiamy $A \cup B$ takich par A, B $\in L_{k-1}$, które mają wspólne *k*-2 początkowych elementów.

Niech

$A = \{i_{A_1}, \ldots, i_{A_{k-2}}, i_{A_{k-1}}\}$,

$B = \{i_{B_1}, \ldots, i_{B_{k-2}}, i_{B_{k-1}}\}$,

wówczas warunkiem dołączenia $A \cup B$ do zbioru C_k jest, aby

$$(i_{A_1}=i_{B_1}) \,\&\, \ldots \,\&\, (i_{A_{k-2}}=i_{B_{k-2}}) \,\&\, (i_{A_{k-1}} < i_{B_{k-1}}).$$

Warunek $i_{A_{k-1}} < i_{B_{k-1}}$ został wprowadzony, aby zapobiec występowaniu powtarzających się elementów w zbiorze C_k.

$A = \{i_{A_1}, \ldots, i_{A_{k-2}}, i_{A_{k-1}}\}$

takie same elementy

$B = \{i_{B_1}, \ldots, i_{B_{k-2}}, i_{B_{k-1}}\}$

zbiór (k-1) - elementowy

$\Rightarrow A \cup B = \{i_{A_1}, \ldots, i_{A_{k-2}}, i_{A_{k-1}}, i_{B_{k-1}}\}$

zbiór k-elementowy

Druga z wymienionych operacji etapu szukania zbiorów częstych to operacja przycinania.

Powstający w wyniku łączenia transakcji zbiór C_k nie musi składać się z samych zbiorów częstych, ale wszystkie k-elementowe zbiory częste należą do C_k,

czyli $L_k \subseteq C_k$.

Celem operacji przycinania jest usunięcie ze zbioru C_k transakcji, które nie są częste. Korzystamy tutaj z własności Apriori. Usuwamy z C_k zbiory, których nie wszystkie podzbiory (k-1) – elementowe należą do L_{k-1}. Z własności Apriori

wynika, że jeżeli jakiś podzbiór takiego zbioru nie należy do L_{k-1}, to taki zbiór nie może należeć do L_k.

Przedstawmy w postaci pseudokodu algorytm generowania zbiorów częstych, przyjmując oznaczenia: L_k – rodzina zbiorów częstych k-elementowych, C_k – rodzina kandydatów na zbiory częste k-elementowe.

Pseudokod ilustrujący algorytm generowania zbiorów częstych będzie obejmował następujące kroki postępowania:

1. Ustal minimalne wsparcie minSupp
2. Oblicz wsparcie dla wszystkich transakcji jednoelementowych tworzących zbiór $C_1 = \{\{i_1\}, \{i_2\}, \ldots, \{i_m\}\}$
3. Wybierz te transakcje jednoelementowe, które spełniają warunek minimalnego wsparcia i utwórz z nich zbiór L_1, taki że $L_1 = \{x: supp(x) > minSupp\}$
4. k = 1

5. Dopóki $\bar{\bar{L}}_k > k+1$, gdzie $\bar{\bar{L}}_k$ jest mocą zbioru, czyli liczbą elementów w zbiorze powtarzaj
   - 5.1. Utwórz zbiór kandydatów $C_{k+1} = L_k \times L_k$ (łączenie)
   - 5.2. Usuń z C_{k+1} zbiory, które zawierają nieczęsty podzbiór o rozmiarze k, tzn. zbiory $x \in L_k$
   - 5.3. Oblicz wsparcie dla pozostałych zbiorów $x \in C_{k+1}$ (nieusuniętych w pkt. 5.2)
   - 5.4. Usuń z C_{k+1} zbiory, które nie spełniają warunku minimalnego wsparcia
   - 5.5. Z pozostałych elementów zbioru C_{k+1} utwórz zbiór L_{k+1}
   - 5.6. Zwiększ k, tak że k:=k+1
6. Jeżeli $L_k = \emptyset$, to k:=k-1

7. Zachowaj wszystkie zbiory częste $\bigcup_{i=1}^{k} L_i$

1.5. Budowanie reguł na podstawie zbiorów częstych

Tworzenie reguł na podstawie zbiorów częstych polega na zamianie zbioru częstego na regułę (wydzielenie poprzednika i następnika) i sprawdzeniu, czy tak określona reguła ma ufność większą niż przyjęta minimalna wartość minConf. Definicja zbioru częstego określa, które elementy występują w regule, ale nie określa poprzednika i następnika. Budowanie reguł polega na przerzucaniu po kolei elementów z poprzednika do następnika i sprawdzaniu, czy w ten sposób utworzona reguła $X \rightarrow Y$ spełnia warunek $conf(X \rightarrow Y) > minConf$ przyjęte w zadaniu.

Mając np. zbiór częsty {A, B, C, D} możemy utworzyć z niego następujących 14 kandydatów na regułę:
A&B&C → D, A&B&D → C, A&C&D → B, B&C&D → A, (następnik 1-elementowy)
A&B → C&D, A&C → B&D, A&D → B&C, B&C → A&D, B&D → A&C, C&D → A&B, (następnik 2-elementowy)
A → B&C&D, B → A&C&D, C → A&B&D, D → A&B&C, (następnik 3-elementowy).

Najpierw tworzymy wszystkie reguły zawierające jeden element w następniku. Usuwamy reguły, które nie spełniają warunku minimalnej ufności. W następnym kroku staramy się dla każdej z nieodrzuconych reguł przerzucić jeden element z poprzednika do następnika. Odrzucamy te reguły, które nie spełniają warunku minimalnej ufności. W algorytmie korzystamy z faktu, że jeżeli reguła AB → CD spełnia warunek minimalnej ufności, to ABC → D i ABD → C też spełniają. Staramy się otrzymać reguły, mające mały poprzednik i duży następnik.

Algorytm Apriori jest dosyć efektywny w przypadku bazy transakcji, w której transakcje zawierają niedużo elementów. Algorytm przestaje być efektywny, gdy liczba transakcji jest duża, sięga milionów pozycji (rekordów). Poszukiwano modyfikacji algorytmu, w których redukuje się liczbę przeglądań całej bazy transakcji (algorytm AprioriTid, Apriori Hybryd [Agrawal, Srikant, 1994]).

Pseudokod ilustrujący algorytm generowania reguł można przedstawić jako obejmujący następujące kroki postępowania:

1. k=2
2. Dla każdego $x \in L_k$
   2.1. Dla każdego $y \subset x$ (gdzie $y \neq 0$, $y \neq x$)
      2.1.1. Zbuduj regułę y → x\y (tzn. poprzednik y, a następnik zawiera te elementy z x, które nie należą do y)
      2.1.2. Oblicz conf(y → x\y)
      2.1.3. Jeżeli conf(y → x\y) > minConf, to zapamietaj tę regułę
3. Zwiększ k: k:=k+1
4. Jeżeli $L_k = Ø$, to przejdź do punktu 2.
5. Zwróć zapamiętane reguły

2. Przykład zastosowania algorytmu Apriori

2.1. Opis przykładu – wybór zajęć dokonywanych przez studentów

Na Wydziale Nauk Ekonomicznych Uniwersytetu Warszawskiego prowadzone są zajęcia dotyczące metod ilościowych analizy danych. Zajęcia prowadzone są we współpracy z SAS Institute Polska i tworzą cykl zajęć o nazwie „Data Mining Certificate Program". W ramach cyklu realizowanych jest 7 rodzajów zajęć (przedmiotów trzydziestogodzinnych). Studenci mogą wybierać poszczególne zajęcia podczas kolejnych lat studiów, począwszy od III roku. Jeżeli student wybierze i zaliczy 210 godzin zajęć cyklu (7 przedmiotów po 30 godzin) uzyskuje Certyfikat który jest przyznawany wspólnie przez Wydział Nauk Ekonomicznych Uniwersytetu Warszawskiego i SAS Institute Polska, bez potrzeby zdawania dodatkowych egzaminów. Do przedmiotów wchodzących w skład cyklu należą (w nawiasach podano przyjęte przez nas w artykule nazwy zajęć): 1) „Przetwarzanie i wizualizacja danych" (WIZ), 2) „Statystyczna analiza danych z pakietem SAS" (ST), 3) „Ekonometryczna analiza danych z pakietem SAS" (ETS), 4) „SAS –probabilistyczne i deterministyczne modele optymalizacji decyzji" (OR), 5) „Bazy danych oraz hurtownie danych" (HD), 6) „Zastosowanie metod eksploracji danych (Data Mining) w badaniach ekonomicznych" (EM), 7) „Analiza danych nieustrukturyzowanych" (ND). W artykule zajęliśmy się analizą wyboru przedmiotów przez studentów trzeciego roku. Interesowało nas, jakie przedmioty wybierali studenci trzeciego roku, a więc rozpoczynając realizację zajęć cyklu i liczba wybieranych przedmiotów przez „pojedynczego" studenta.

Wśród studentów III roku WNE w roku akademickim 2012/2013 60 osób brało udział w zajęciach Data Mining Certificate Program. Uczestniczyli oni w od 1 do 6 (spośród 7) przedmiotach ścieżki. Łącznie zanotowano 183 uczestnictwa w zajęciach. Mamy zatem 60 transakcji (studentów), których ponumerowaliśmy od 1 do 60. Transakcja jest zbiór przedmiotów wybranych przez studenta id_j. Bazę transakcji przedstawiono w tab. 1.

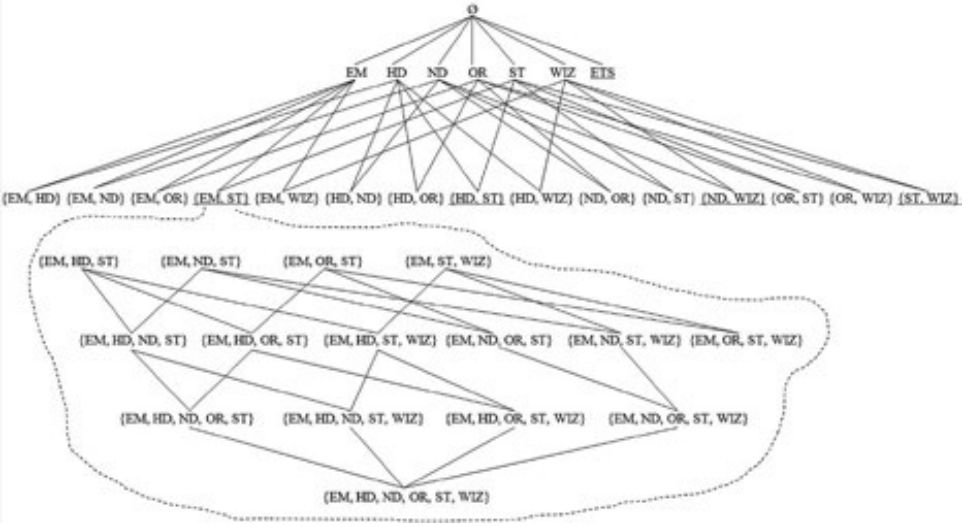
Tab. 1. Lista transakcji

Numer transakcji	Transakcja	Numer transakcji	Transakcja	Numer transakcji	Transakcja
1	{ND}	21	{EM, HD, OR, WIZ, ND}	41	{EM, HD, OR}
2	{EM, ND}	22	{EM}	42	{ST, OR}
3	{EM, HD, WIZ, ND, OR}	23	{EM, OR, ND}	43	{ND}
4	{EM, HD, OR}	24	{ST}	44	{HD, ST, OR, ND}
5	{EM, OR, ND}	25	{EM, HD, OR, WIZ}	45	{OR, ND}
6	{ETS, ST, ND}	26	{EM, OR,WIZ}	46	{OR, ND}
7	{EM, ND, OR}	27	{HD, ND, OR}	47	{ST}

Numer transakcji	Transakcja	Numer transakcji	Transakcja	Numer transakcji	Transakcja
8	{EM, HD, OR, ND}	28	{EM, HD, WIZ, ND, OR}	48	{ST, OR}
9	{ST}	29	{HD}	49	{HD, OR, ND}
10	{HD}	30	{OR}	50	{EM, ST, OR}
11	{OR, WIZ}	31	{EM, HD, OR}	51	{EM, ST, ND, OR}
12	{HD, OR, ND}	32	{EM, HD, OR}	52	
13	{EM, OR}	33	{EM, HD, ST, OR, WIZ, ND}	53	{ST, WIZ, OR}
14	{EM, HD, OR, WIZ, ND}	34	{EM, HD, OR, ND}	54	{HD, ST, OR}
15	{EM, ST, OR, ND}	35	{EM, HD, ST, OR, WIZ, ND}	55	{HD, ST, OR, ND}
16	{EM, HD, OR, WIZ}	36	{EM, OR}	56	{HD, ST, OR, ND}
17	{EM, HD, OR, WIZ}	37	{EM, HD, ND, OR}	57	{EM, HD, ST, OR, ND}
18	{EM, HD, OR, WIZ}	38	{HD, ST}	58	{EM, ND}
19	{EM, HD, OR}	39	{HD, ND, OR}	59	{EM, ST, OR}
20	{EM, HD, OR, ND}	40	{EM, HD, ND, OR}	60	{EM, OR, ND}

Źródło: Opracowanie własne.

Generowanie reguł przeprowadzamy na podstawie zbiorów częstych.



Rys. 1. Przykład pomijania nieczęstych nadzbiorów
Źródło: Opracowanie własne.

Na rysunku 1 podkreślono zbiory, które nie są częste. Jednym z nich jest zbiór {EM, ST}. Dla tego zbioru ukazano jego nadzbiory, które zgodnie z zasadą przyjętą w metodzie Apriori nie są częste i zostają odrzucone w procesie tworzenia reguł.

2.2. Poszukiwanie zbiorów częstych dla przykładu dotyczącego wyboru zajęć przez studentów

Porządkujemy przedmioty alfabetycznie (leksykograficznie): EM, ETS, HD, ND, OR, ST, WIZ. Obliczamy w ilu transakcjach wystąpiły:

EM w 36 transakcjach, ETS w jednej, HD w 32, ND – 33, OR – 48, ST – 20, WIZ – 13.

Przedmioty tworzą zbiór C_1 – transakcji 1-elementowych. Ustalamy minimalne wsparcie minSupp = 15%. Ponieważ mamy 60 transakcji, więc 15% oznacza 9 transakcji. Jednoelementowe zbiory częste (L_1) są to takie zbiory przedmiotów (jednoelementowe !), które wystąpiły przynajmniej w 9 transakcjach. Są nimi uwzględniane przedmioty oprócz ETS.

C_1	Liczba wystąpień
{EM}	36
{ETS}	1
{HD}	32
{ND}	33
{OR}	48
{ST}	20
{WIZ}	13

→

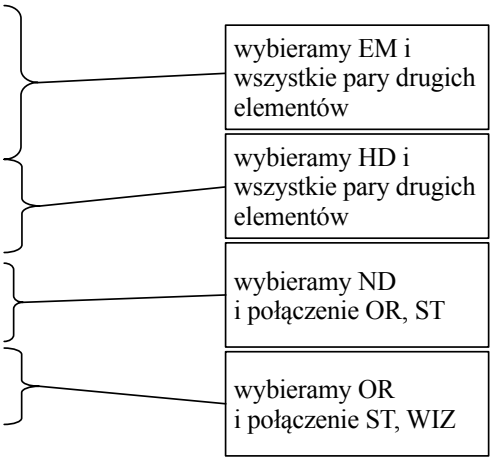
L1	Liczba wystąpień
{EM}	36
{HD}	32
{ND}	33
{OR}	48
{ST}	20
{WIZ}	13

Utworzyliśmy 1-elementowe zbiory częste L_1. Teraz ze zbioru L_1 tworzymy zbiory 2-elementowe (łączenie). Jest tych zbiorów 15. Tworzą one zbiór C_2. Aby zbiór należący do C_2 był częsty, musi wystąpić w przynajmniej 9 transakcjach.

C_2	Liczba wystąpień
{EM, HD}	21
{EM, ND}	21
{EM, OR}	33
{EM, ST}	8
{EM, WIZ}	11
{HD, ND}	19
{HD, OR}	29
{HD, ST}	8
{HD, WIZ}	9
{ND, OR}	28
{ND, ST}	10
{ND, WIZ}	6
{OR, ST}	15
{OR, WIZ}	13
{ST, WIZ}	3

Z danych przedstawionych powyżej wynika, że 2-elementowe zbiory {EM, ST}, {HD, ST}, {ND, WIZ} oraz {ST, WIZ} występują w mniejszej liczbie niż w 9 transakcjach. Nie są więc częste. Poniżej zestawiono zbiory częste 2-elementowe (L_2).

L_2	Liczba wystąpień
{EM, HD}	21
{EM, ND}	21
{EM, OR}	33
{EM, WIZ}	11
{HD, ND}	19
{HD, OR}	29
{HD, WIZ}	9
{ND, OR}	28
{ND, ST}	10
{OR, ST}	15
{OR, WIZ}	13



Przeprowadzamy dalej operację łączenia. Wybieramy EM i wszystkie pary drugich elementów, wybieramy HD i wszystkie pary drugich elementów, wybieramy ND i połączenie OR, ST, wybieramy OR i połączenie ST, WIZ.

Operacja łączenia – wyniki:

C_3	Liczba wystąpień
{EM, HD, ND}	12
{EM, HD, OR}	21
{EM, HD,WIZ}	10
{EM, ND, OR}	19
{EM, ND,WIZ}	6
{EM, OR,WIZ}	11
{HD, ND, OR}	19
{HD, ND,WIZ}	6
{HD, OR,WIZ}	10
{ND, OR, ST}	9
{OR, ST, WIZ}	3

Przeprowadzamy przycinanie C_3. Usuwamy {EM, ND, WIZ}, bo ma tylko 6 wystąpień. Usuwamy {HD, ND, WIZ}, bo podzbiór {ND, WIZ} nie jest częsty ($\notin L_2$) - przycinanie C_3. Podobnie usuwamy (przycinanie C_3) zbiór {OR, ST, WIZ}, bo {ST, WIZ} nie jest częsty ($\notin L_2$).

L_3	Liczba wystąpień
{EM, HD, ND}	12
{EM, HD, OR}	21
{EM, HD,WIZ}	10
{EM, ND, OR}	19
{EM,OR,WIZ}	11
{HD, ND, OR}	19
{HD, OR,WIZ}	10
{ND, OR, ST}	9

wybieramy, żeby 2 pierwsze elementy były takie same – otrzymamy 3 zbiory 4-elementowe

Nie da się wybrać, żeby 2 pierwsze elementy były takie same

C_4	Liczba wystąpień
{EM, HD, ND, OR}	12
{EM, HD, ND, WIZ}	
{EM, HD, OR, WIZ}	10

Podzbiór {HD, ND, WIZ} nie należy do L_3 (przycinanie C_4)

{EM, HD, ND, WIZ} zostaje usunięte, bo {HD, ND, WIZ} nie jest częsty ($\notin L_3$)

L_4	Liczba wystąpień
{EM, HD, ND, OR}	12
{EM, HD, OR, WIZ}	10

C_5	Liczba wystąpień
{EM, HD, ND, OR, WIZ}	6

Zawiera podzbiór {EM, HD, ND, WIZ}, który nie jest częsty

C_5 zawiera jeden zbiór, który ma mniej niż 9 elementów (nie jest częsty), zatem L_5 jest zbiorem pustym.

2.3. Tworzenie reguł na podstawie zbiorów częstych dla przykładu dotyczącego wyboru zajęć przez studentów

Prześledźmy dla naszego przykładu jak odbywa się, zgodnie z algorytmem Apriori, generowanie reguł na podstawie zbiorów częstych. Zaczynamy od zbioru L_2, obejmującego zbiory częste dwuelementowe. W naszym przypadku L_2 zawiera 11 elementów, a więc można utworzyć 22 kandydatów na reguły, jak zilustrowano

to poniżej.

Lp.	Zbiór częsty dwuelementowy	Nr reguły kandydata	Reguła kandydat	Nr reguły kandydata	Reguła kandydat
1.	{EM, HD}	1	EM → HD	12	HD → EM
2.	{EM, ND}	2	EM → ND	13	ND → EM
3.	{EM, OR}	3	EM → OR	14	OR → EM
4.	{EM, WIZ}	4	EM → WIZ	15	WIZ → EM
5.	{HD, ND}	5	HD → ND	16	ND → HD
6.	{HD, OR}	6	HD → OR	17	OR → HD
7.	{HD, WIZ}	7	HD → WIZ	18	WIZ → HD
8.	{ND, OR}	8	ND → OR	19	OR → ND
9.	{ND, ST}	9	ND → ST	20	ST → ND
10.	{OR, ST}	10	OR → ST	21	ST → OR
11.	{OR, WIZ}	11	OR → WIZ	22	WIZ → OR

Zbiór L_3 zawiera 8 elementów postaci {A, B, C}. Z każdego zbioru należącego do L_3 można utworzyć 3 kandydatów na reguły: A & B → C, A & C → B, B & C → A oraz 3 kandydatów na reguły: A → B & C, B → A & C, C → A & B.

W naszym przypadku mamy następujące zbiory częste trzyelementowe: {EM, HD, ND}, {EM, HD, OR}, {EM, HD, WIZ}, {EM, ND, OR}, {EM, OR, WIZ}, {HD, ND, OR}, {HD, OR, WIZ}, {ND, OR, ST}. Ponieważ L_3 zawiera 8 elementów, a więc łącznie można utworzyć 48 kandydatów na reguły, jak przedstawiono poniżej. Wydzielono reguły kandydujące, tworzone z kolejnych zbiorów częstych trzyelementowych: z {EM, HD, ND} – reguły kandydaci o numerach od 1 do 6, z {EM, HD, OR} – o numerach od 7 do 12, {EM, HD, WIZ} – o numerach od 13 do 18, z {EM, ND, OR} – o numerach od 19 do 24, z {EM, OR, WIZ} – o numerach od 25 do 30, z {HD, ND, OR} – o numerach od 31 do 36, z {HD, OR, WIZ} – o numerach od 37 do 42, z {ND, OR, ST} – o numerach od 43 do 48.

Nr reguły kandydata	Reguła kandydat	Nr reguły kandydata	Reguła kandydat	Nr reguły kandydata	Reguła kandydat
1	EM & HD → ND	17	HD → EM & WIZ	33	ND & OR → HD
2	EM & ND → HD	18	EM → HD & WIZ	34	OR → HD & ND
3	HD & ND → EM	19	EM & ND → OR	35	ND → HD & OR
4	ND → EM & HD	20	EM & OR → ND	36	HD → ND & OR
5	HD → EM & ND	21	ND & OR → EM	37	HD & OR → WIZ

6	EM → HD & ND	22	OR → EM & ND	38	HD & WIZ → OR
7	EM & HD → OR	23	ND → EM & OR	39	OR & WIZ → HD
8	EM & OR → HD	24	EM → ND & OR	40	WIZ → HD & OR
9	HD & OR → EM	25	EM & OR → WIZ	41	OR → HD & WIZ
10	OR → EM & HD	26	EM &WIZ → OR	42	HD → OR & WIZ
11	HD → EM & OR	27	OR &WIZ → EM	43	ND & OR → ST
12	EM → HD & OR	28	WIZ → EM & OR	44	ND & ST → OR
13	EM & HD → WIZ	29	OR → EM & WIZ	45	OR & ST → ND
14	EM & WIZ → HD	30	EM → OR & WIZ	46	ST → ND & OR
15	HD & WIZ → EM	31	HD & ND → OR	47	OR → ND & ST
16	WIZ → EM & HD	32	HD & OR → ND	48	ND → OR & ST

Zbiór L_4 zawiera 2 elementy: {EM, HD, ND, OR} i {EM, HD, OR, WIZ}. Z każdego z nich można utworzyć 14 reguł – kandydatów, a więc łącznie 28. Z {EM, HD, ND, OR} będą to:

EM & HD & ND → OR	EM & HD & OR → ND	
EM & ND & OR → HD	HD & ND & OR → EM	
EM & HD → ND & OR	EM & ND → HD & OR	EM & OR → HD & ND
ND & OR → EM & HD	HD & OR → EM & ND	HD & ND → EM & OR
EM → HD & ND & OR	HD → EM & ND & OR	
ND → EM & HD & OR	OR → EM & HD & ND	

Uwzględniając {EM, HD, OR, WIZ} otrzymamy:

EM & HD & OR → WIZ	EM & HD & WIZ → OR	
EM & OR & WIZ → HD	HD & OR & WIZ → EM	
EM & HD → OR & WIZ	EM & OR → HD & WIZ	EM & WIZ → HD & OR
OR & WIZ → EM & HD	HD & WIZ → EM & OR	HD & OR → EM & WIZ
EM → HD & OR & WIZ	HD → EM & OR & WIZ	
OR → EM & HD & WIZ	WIZ → EM & HD & OR	

Z powyższych rozważań, dotyczących zbiorów częstych L_2, L_3, L_4 wynika, że mamy (22 +48+28) 98 kandydatów reguł na opracowanie akceptowanych reguł.

Można sprawdzić, że spełniają one warunek minimalnej ufności, a więc otrzymaliśmy 98 reguł asocjacyjnych.

3. Analiza asocjacji i tworzenie reguł wyboru zajęć przez studentów za pomocą programu SAS Enterprise Miner

Przeprowadzenie analizy asocjacji i wygenerowanie reguł asocjacji za pomocą programu SAS Enterprise Miner [Lasek, Pęczkowski, 2013; SAS Enterprise Miner 12.1 Reference Help, 2011] wymaga przygotowania danych wejściowych w tzw. formacie transakcyjnym, w którym niezbędne są dwie kolumny: (i) identyfikator transakcji, w naszym przypadku identyfikator studenta – na rys. 2. zawartość kolumny *student* oraz (ii) pozycja, w naszym przypadku nazwa wybranego przedmiotu (użyliśmy przyjętych przez nas w tym artykule nazw zajęć) – na rys. 2. zawartość kolumny *Przedmiot*. Ten sam identyfikator transakcji (numer studenta) może występować wiele razy. W kolumnie *Przedmiot* każdy wiersz zawiera tylko jedną pozycję.

Rys. 2. Dane wejściowe (fragment)

	Przedmiot	student
1	ND	1
2	ND	2
3	EM	2
4	EM	3
5	ND	3
6	WIZ	3
7	OR	3
8	HD	3
9	OR	4
10	HD	4
11	EM	4
12	EM	5
13	OR	5
14	ND	5
15	ETS	6
16	ND	6
17	ST	6
18	EM	7
19	OR	7
20	ND	7

Źródło: Opracowanie własne przy użyciu programu SAS Enterprise Miner.

Przeprowadzanie analizy danych za pomocą programu SAS Enterprise Miner wymaga zbudowania diagramu, złożonego z tzw. węzłów diagramu ilustrujących

realizowane, poszczególne procedury i liniami ze strzałką skierowaną w kierunku przetwarzania, od węzła do węzła diagramu. W naszym przypadku, jak ilustruje to rysunek 3, diagram składa się jedynie z dwóch węzłów połączonych linią ze strzałką: od danych wejściowych (węzeł *Dane wejściowe*) do węzła realizującego analizę asocjacji i tworzenie reguł asocjacyjnych (węzeł *Association*).

Rys. 3. Diagram dla przeprowadzenia analizy asocjacji

Źródło: Opracowanie własne przy użyciu programu SAS Enterprise Miner.

Możemy ustalać parametry, określające przetwarzanie danych w każdym z węzłów. W węźle *Association*, jak ilustruje to rysunek 4 przyjęliśmy, że będą brane pod uwagę reguły, które zawierają co najwyżej 4 zajęcia, czy też przedmioty (*Associations Maximum Items 4*), o minimalnej ufności wynoszącej 20% (*Association Minimum Confidence Level 20*) i o minimalnym wsparciu – 15% (*Association Support Percentage 15*).

Rys. 4. Menu właściwości węzła *Association* z przyjętymi parametrami

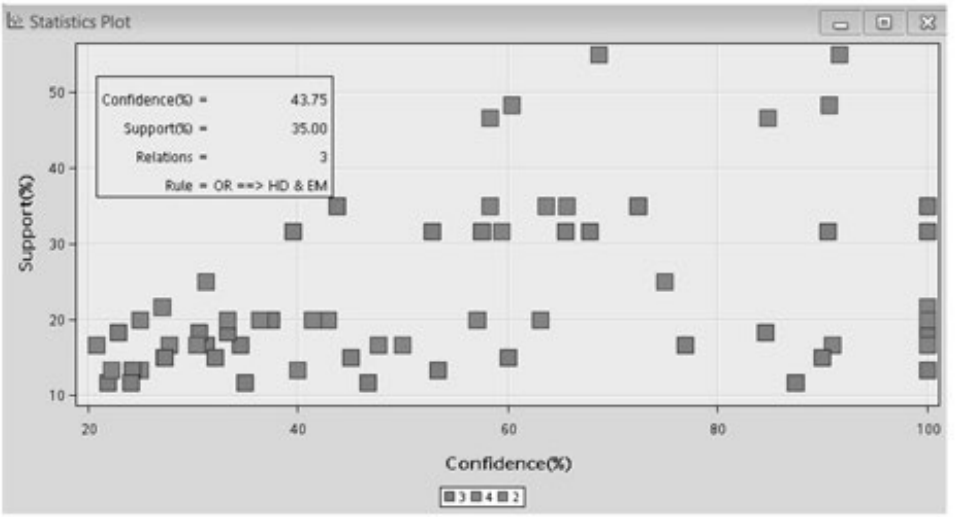
.. Property	Value
General	
Node ID	Assoc
Imported Data	
Exported Data	
Notes	
Train	
Variables	
Maximum Number of Items to Process	100000
Rules	
Association	
Maximum Items	4
Minimum Confidence Level	20
Support Type	Percent
Support Count	.
Support Percentage	15.0

Źródło: Opracowanie własne przy użyciu programu SAS Enterprise Miner.

Po uruchomieniu i zakończeniu przetwarzania w węźle *Association* możemy oglądać wyniki przetwarzania przedstawiane w różny sposób. Na rysunku 5 przedstawiamy wykres ilustrujący dla reguł: ufność i wsparcie. Kierując wskaźnik myszy na daną pozycję reprezentującą daną regułę asocjacji (kwadrat na wykre-

sie), możemy postać reguły oraz jej wartości ufności i wsparcia. Na rysunku 5 przedstawiliśmy przykładowo regułę OR → HD & EM, która ma ufność 43,75%, a wsparcie 35%. Występują też reguły, które mają ufność 100%, ale małe wsparcie por. rys. 5).

Rys. 5. Wykres *Statistics Plot*



Źródło: Opracowanie własne przy użyciu programu SAS Enterprise Miner.

Na rysunku 6 przedstawiamy fragment tabeli, przedstawiającą zestawienie reguł posortowanych malejąco według wskaźnika *Lift* – przyrostu, wskazującego, jak rośnie wiarygodność następnika reguły przy wzroście wiarygodności poprzednika (pożądane są wartości *Lift* większe od 1). Na rysunku 6 przedstawiane są także w oddzielnych kolumnach: *Relations* – liczba pozycji, jaka występuje w regule (wierszu kolumny *Rule*), *Confidence* – ufność, częstość występowania następnika, gdy występuje poprzednik (w %), *Support* – wsparcie, częstość występowania jednocześnie poprzednika i następnika (w %), *Transaction Count* – liczba transakcji (w naszym przykładzie: liczba studentów, którzy wybrali przedmioty (zajęcia) występujące w regule).

Rys.6. Tabela *Rules Tables* (fragment)

Rules Table

Relations	Confidence(%)	Support(%)	Lift	Transaction Count	Rule
3	76.92	16.67	2.20	10.00	WIZ ==> HD & EM
4	76.92	16.67	2.20	10.00	WIZ & OR ==> HD & EM
4	76.92	16.67	2.20	10.00	WIZ ==> OR & HD & EM
3	47.62	16.67	2.20	10.00	HD & EM ==> WIZ
4	47.62	16.67	2.20	10.00	OR & HD & EM ==> WIZ
4	47.62	16.67	2.20	10.00	HD & EM ==> WIZ & OR
4	90.91	16.67	1.88	10.00	WIZ & EM ==> OR & HD
4	34.48	16.67	1.88	10.00	OR & HD ==> WIZ & EM
4	100.00	16.67	1.82	10.00	WIZ & HD ==> OR & EM
4	30.30	16.67	1.82	10.00	OR & EM ==> WIZ & HD
3	90.91	16.67	1.70	10.00	WIZ & EM ==> HD
4	90.91	16.67	1.70	10.00	WIZ & OR & EM ==> HD
3	31.25	16.67	1.70	10.00	HD ==> WIZ & EM
4	31.25	16.67	1.70	10.00	HD ==> WIZ & OR & EM
3	100.00	16.67	1.67	10.00	WIZ & HD ==> EM
4	100.00	16.67	1.67	10.00	WIZ & OR & HD ==> EM
3	27.78	16.67	1.67	10.00	EM ==> WIZ & HD
4	27.78	16.67	1.67	10.00	EM ==> WIZ & OR & HD
3	76.92	16.67	1.59	10.00	WIZ ==> OR & HD
3	34.48	16.67	1.59	10.00	OR & HD ==> WIZ
3	84.62	18.33	1.54	11.00	WIZ ==> OR & EM
3	33.33	18.33	1.54	11.00	OR & EM ==> WIZ
2	76.92	16.67	1.44	10.00	WIZ ==> HD
3	76.92	16.67	1.44	10.00	WIZ & OR ==> HD
2	31.25	16.67	1.44	10.00	HD ==> WIZ
3	31.25	16.67	1.44	10.00	HD ==> WIZ & OR
2	84.62	18.33	1.41	11.00	WIZ ==> EM
3	84.62	18.33	1.41	11.00	WIZ & OR ==> EM
2	30.56	18.33	1.41	11.00	EM ==> WIZ
3	30.56	18.33	1.41	11.00	EM ==> WIZ & OR
3	67.86	31.67	1.27	19.00	OR & ND ==> HD
3	59.38	31.67	1.27	19.00	HD ==> OR & ND
3	43.75	35.00	1.25	21.00	OR ==> HD & EM
3	39.58	31.67	1.25	19.00	OR ==> ND & HD
4	25.00	20.00	1.25	12.00	OR ==> ND & HD & EM
2	100.00	21.67	1.25	13.00	WIZ ==> OR
3	100.00	35.00	1.25	21.00	HD & EM ==> OR
3	100.00	31.67	1.25	19.00	ND & HD ==> OR
3	100.00	18.33	1.25	11.00	WIZ & EM ==> OR
3	100.00	16.67	1.25	10.00	WIZ & HD ==> OR
3	100.00	13.33	1.25	8.00	ST & EM ==> OR
4	100.00	20.00	1.25	12.00	ND & HD & EM ==> OR
4	100.00	16.67	1.25	10.00	WIZ & HD & EM ==> OR
2	27.08	21.67	1.25	13.00	OR ==> WIZ
3	22.92	18.33	1.25	11.00	OR ==> WIZ & EM

Źródło: Opracowanie własne przy użyciu programu SAS Enterprise Miner.

Zakończenie

W artykule przedstawiliśmy metodę analizy asocjacji i tworzenia reguł asocjacyjnych za pomocą algorytmu Apriori, aby następnie ukazać jej przydatność w odkrywaniu prawidłowości dokonywania wyboru zajęć dydaktycznych przez studentów.

Zaproponowany przez nas sposób analizy wyborów zajęć przez studentów za pomocą badań asocjacyjnych i budowaniu reguł asocjacyjnych ukazał szereg zalet, pozwalających pozyskać możliwie wiarygodną wiedzę o wyborach studentów, która może stanowić cenną wskazówkę przy planowaniu oferty zajęć do wyboru przez studentów w następnych cyklach dydaktycznych.

Każdą z pozyskanych reguł asocjacyjnych możemy ocenić posługując się opisanymi w artykule wskaźnikami: wsparcia reguły (ang. support), ufności (ang. confidence), przyrostu (ang. lift). Wskaźniki pozwalają oszacować i wnioskować, z jakim stopniem pewności (prawdopodobieństwa) możemy polegać na informacji, czy też wiedzy przedstawianej przez regułę.

Metoda analizy asocjacji dostarczyła nam przydatnej dla organizacji zajęć dydaktycznych, informacji o tym , jakie przedmioty są wybierane w największej liczbie przez studentów (przez największą liczbę studentów) „w pierwszej kolejności", gdy mogą dokonywać wyborów w kolejnych latach. Przydatne byłoby zbadanie, czym kierują się studenci, decydując o kolejności wyboru zajęć, czy jest to posiadana już wiedza z innych przedmiotów, spodziewana przydatność wiedzy dla uczestniczenia w innych przedmiotach w najbliższym czasie, obciążenie innymi zajęciami w terminie, na który dokonywany jest wybór, ocena osób prowadzących zajęcia do wyboru w danym terminie, gdy w innych terminach te same zajęcia będą prowadzić osoby lepiej oceniane przez studentów, czy też inne przyczyny.

W przedstawianym przez nas w artykule zagadnieniu wyboru przez studentów zajęć, należących do cyklu zajęć o nazwie „Data Mining Certificate Program", 60 studentów dokonało wyboru zajęć należących do tego cyklu. Możemy więc wskazać 60 zrealizowanych transakcji (transakcji rozumianych zgodnie z terminologią używaną w ramach metody analizy asocjacji), obejmujących 183 przedmioty uwzględnione łącznie we wszystkich transakcjach. W różnych transakcjach przedmioty powtarzają się, stąd ich liczba (183), pomimo, że studenci wybierali przedmioty z zaledwie 7 przedmiotów, pozostających do ich dyspozycji.

Wiele cennych informacji dla planowania organizacji zajęć może dostarczyć analiza zbudowanych reguł asocjacyjnych. W szczególności przydatna wydaje się wiedza, o tym, jakie przedmioty wybierają studenci łącznie w tym samym okresie (semestrze) studiów, co zostaje wskazane w regułach asocjacyjnych. Ocenę wiarygodności (prawdopodobieństwa) takich łącznych wyborów przedmiotów (reguł asocjacyjnych z powtarzającymi się, takimi samymi pozycjami) umożliwia wykorzystanie wskaźników reguł: wsparcia reguły (ang. support), ufności (ang. confidence), przyrostu (ang. lift).

W końcowej części artykułu wskazaliśmy jeden spośród programów umożliwiających komputerowe wspomaganie przeprowadzania analizy asocjacji, tworzenie reguł asocjacyjnych i pomoc w interpretacji uzyskiwanych wyników, mianowicie SAS Enterprise Miner firmy SAS Institute Inc. z USA. Program ten wykorzystywaliśmy w naszej analizie wyboru zajęć dydaktycznych dokonywanych przez studentów. Jest to bardzo wygodne narzędzie do prowadzenia analiz asocjacyjnych. Przy większej liczbie danych, bez posiadania odpowiedniego oprogramowania, jakiekolwiek analizy asocjacji nie wydają się możliwe do zrealizowania.

Bibliografia

Agrawal R., Imieliński T., Swami A. (1993) Mininig association rules between sets of items in large databases, Proc. ACM SIGMOD Conference on Management of Data, s. 207-216.

Agrawal R., Srikant R. (1994) Fast algorithms for mining association rules, Proceedings of the 20th International Conference on Very Large Data Bases (VLDB'94), Santiago, Chile, s. 478-499.

Aher S. B., Lobo L. M. R. J. (2012) A Comparative Study of Association Rule Algorithms for Course Recommender System in E-learning, International Journal of Computer Applications, t. 39, no. 1.

Berry M. J. A., Linoff G. S. (2004) Data Mining Techniques. For Marketing, Sales, and Customer Relationship Management, Wiley Publishing Inc.

Larose D. T. (2006) Odkrywanie wiedzy z danych. Wprowadzenie do eksploracji danych, Wydawnictwo Naukowe PWN, Warszawa.

Lasek M., Nowak E., Pęczkowski M. (2008) Association and Sequence Rules of Events in an Investment Analysis of Agrotourism Farms/Zastosowanie reguł asocjacji i sekwencji zdarzeń do analizy działalności inwestycyjnej gospodarstw agroturystycznych (artykuły w dwóch wersjach językowych: angielskim i polskim), Turyzm, 18/2.

Lasek M. (2004) Od danych do wiedzy. Metody i techniki „Data Mining", Optimum, nr 2(22).

Lasek M., Pęczkowski M. (2013) Enterprise Miner. Wykorzystywanie narzędzi Data Mining w systemie SAS, Wydawnictwa Uniwersytetu Warszawskiego, Warszawa.

Morzy T. (2013) Eksploracja danych. Metody i algorytmy, Wydawnictwo Naukowe PWN, Warszawa.

Nguyen Sinh Hoa (2013) Reguły asocjacyjne, algorytm Apriori, http://edu.pjwstk.edu.pl/wyklady/adn/scb/wyklad12/w12.htm (dostęp w dniu 28 grudnia 2013).

SAS Enterprise Miner 12.1 Reference Help, SAS Institute Inc., Cary, NC, USA, 2011.

Association analysis and association rules in exploring the choices made by students from elective courses. Application of Apriori algorithm

Abstract

This article discusses the possibilities and advantages of the association analysis method, belonging to Data Mining, in problems relating to choices made by students from elective courses. Such choices are possible when students can freely choose several courses from separate groups of elective courses. The association analysis method and the construction of association rules are briefly described. Particular attention was paid to the Apriori algorithm. It constitutes one of the most popular algorithms for the association analysis and the construction of association rules. The Apriori algorithm in an orderly and logical manner performs necessary actions as well as accessibly, transparently and understandably reflects the concept of the association analysis and the construction of association rules. The considerations are illustrated using an example of choices made by students from elective courses conducted within the educational path called "Data Mining Certificate Program" at the Faculty of Economic Sciences, Univerity of Warsaw, in cooperation with SAS Institute Polska. The discussed issue of elective courses selection was explored using SAS Enterprise Miner software from SAS Institute Inc. U.S. – the article provides a very short presentation of its functionality, possibilities of performing the association analysis and the construction of association rules, as well as interpretation of the analysis results.

Keywords: association analysis, association rules, Data Mining, Apriori algorithm, SAS Enterprise Miner software from SAS Institute Inc. U.S., analysis of courses selection by students