Katedra Sztucznej Inteligencji i Matematyki Stosowanej
WI ZUT Szczecin

Sztuczna inteligencja i uczenie maszynowe w systemach interaktywnych

dr inż. Joanna Kolodziejczyk
pokój nr 27 WI1
konsultacje: czwartki 12:00 - 14:00

October 13, 2021

Plan prezentacji

Indukcja reguł decyzyjnych

- Informacje podstawowe
- Zbiór danych - problemy
- Definicja pokrycia, kompletności i spójności

Algorytmy generowania reguł

- Algorytmy indukcji reguł decyzyjnych
- Algorytm LEM1

Indukcja reguł asocjacyjnych

- Podstawowe miary
- Algorytm Apriori

Indukcja reguł
Istota

Zalety

Indukcja reguł jest jedną z ważniejszych technik maszynowego uczenia się. Regularności ukryte w danych są często i chętnie wyrażane w postaci reguł. Przedstawienie w tej postacie wiedzy jest zrozumiałe dla człowieka i podlega w łatwy sposób intuicyjnej ocenie.

Wady

Wydobycie zbioru reguł z danych charakteryzuje duży koszt obliczeniowy i pamięciowy.

Definicja

Reguła jest przestawiana jako instrukcja warunkowa. Np.:

$$if(attribute_1 = value_1)\&(attribute_2 = value_2)\& \ldots$$

$$\&(attribute_n = value_n)then(decision = value)$$

Założenie

Rozważymy indukcję reguł, która należy do nadzorowanego uczenia się: wszystkie przypadki są preklasyfikowane przez eksperta. Innymi słowy, wartość decyzji jest przypisywana przez eksperta do każdego przypadku. Atrybuty są niezależnymi zmiennymi, a decyzja jest zmienną zależną.

Przykład zbioru danych
Zbiór wyjściowy

Case	Attributes				Decision
	Temperature	Headache	Weakness	Nausea	Flu
1	very_high	yes	yes	no	yes
2	high	yes	no	yes	yes
3	normal	no	no	no	no
4	normal	yes	yes	yes	yes
5	high	no	yes	no	yes
6	high	no	no	no	no
7	normal	no	yes	no	no

Przykład zbioru danych
Błędy w danych

Case	Attributes				Decision
	Temperature	Headache	Weakness	Nausea	Flu
1	very_high	yes	yes	no	yes
2	high	yes	no	yes	yes
3	normal	no	no	no	no
4	normal	yes	yes	yes	yes
5	high	no	yes	no	yes
6	high	no	no	no	no
7	normal	no	42.5	no	no

Przykład zbioru danych
Dane numeryczne

Case	Attributes				Decision
	Temperature	Headache	Weakness	Nausea	Flu
1	41.6	yes	yes	no	yes
2	39.8	yes	no	yes	yes
3	36.8	no	no	no	no
4	37.0	yes	yes	yes	yes
5	38.8	no	yes	no	yes
6	40.2	no	no	no	no
7	36.6	no	yes	no	no

Przykład zbioru danych
Brakujące wartości

Case	Attributes				Decision
	Temperature	Headache	Weakness	Nausea	Flu
1	very_high	yes	yes	no	yes
2	?	yes	no	yes	yes
3	normal	no	?	no	no
4	normal	?	yes	yes	yes
5	high	no	yes	no	yes
6	high	no	no	no	no
7	normal	no	yes	no	no

Przykład zbioru danych

Dane niespójne - konflikt

Case	Attributes				Decision
	Temperature	Headache	Weakness	Nausea	Flu
1	very_high	yes	yes	no	yes
2	high	yes	no	yes	yes
3	normal	no	no	no	no
4	normal	yes	yes	yes	yes
5	high	no	yes	no	yes
6	high	no	no	no	no
7	normal	no	yes	no	no
8	normal	no	yes	no	yes

Pokrycie (cover)

Przypadek *x* jest **pokryty** regułą *r* wtedy i tylko wtedy, gdy każdy **warunek** (para atrybut=wartość) reguły *r* jest spełniony przez odpowiednią wartość atrybutu dla *x*.

Całkowite pokrycie

Konkluzja *C* zdefiniowana przez prawą stronę reguły *r*. Mówimy, że konkluzja *C* jest **całkowicie pokryta** zestawem reguł *R* wtedy i tylko wtedy, gdy dla każdego przypadku *x* z *C* istnieje reguła *r* z *R* tak, że *r* pokrywa *x*.

Kompletność zestawu reguł

Zestaw reguł R jest **kompletny** wtedy i tylko wtedy, gdy każda konkluzja ze zbioru danych jest całkowicie pokryta przez R.

Spójność (consistent) reguły

Reguła r jest **spójna** (ze zbiorem danych) wtedy i tylko wtedy, gdy dla każdego przypadku x pokrytego przez r, x jest elementem konkluzji C wskazanym przez r. Zbiór reguł R jest spójny wtedy i tylko wtedy, gdy każda reguła z R jest spójna ze zbiorem danych.

Analiza typu reguł
Reguła 1

Case	Attributes				Decision
	Temperature	Headache	Weakness	Nausea	Flu
1	very_high	yes	yes	no	yes
2	high	yes	no	yes	yes
3	normal	no	no	no	no
4	normal	yes	yes	yes	yes
5	high	no	yes	no	yes
6	high	no	no	no	no
7	normal	no	yes	no	no

Reguła $r1$

$$(Headache = yes) \Rightarrow (Flu = yes)$$

- Reguła $r1$ pokrywa rekordy $\{1, 2, 4, 5\}$;
- Konkluzja $\{1, 2, 4, 5\}$ nie jest w pełni pokryta zbiorem reguł zawierającym $r1$, gdyż reguła $r1$ pokrywa tylko przypadki 1, 2 i 4;
- Reguła $r1$ jest spójna ze zbiorem danych (dla każdego rekordu z $Headache = yes$ konkluzja jest $Flu = yes$.

Analiza typu reguł
Reguła 2

Case	Attributes				Decision
	Temperature	Headache	Weakness	Nausea	Flu
1	very_high	yes	yes	no	yes
2	high	yes	no	yes	yes
3	normal	no	no	no	no
4	normal	yes	yes	yes	yes
5	high	no	yes	no	yes
6	high	no	no	no	no
7	normal	no	yes	no	no

Reguła *r*2

$$(Headache = no) \Rightarrow (Flu = no)$$

- Reguła *r*2 pokrywa rekordy $\{3, 6, 7\}$;
- Konkluzja $\{3, 6, 7\}$ jest w pełni pokryta regułą *r*2, gdyż reguła *r*2 pokrywa wszystkie przypadki 3, 6 i 7;
- Reguła *r*2 nie jest spójna ze zbiorem danych, gdyż $Headache = no$ dotyczy przypadków 3, 5, 6 i 7.

Analiza typu reguł
Reguły 3 i 4

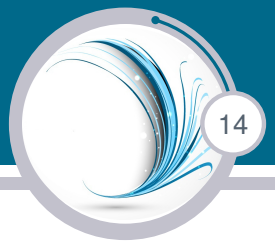
Case	Attributes				Decision
	Temperature	Headache	Weakness	Nausea	Flu
1	very_high	yes	yes	no	yes
2	high	yes	no	yes	yes
3	normal	no	no	no	no
4	normal	yes	yes	yes	yes
5	high	no	yes	no	yes
6	high	no	no	no	no
7	normal	no	yes	no	no

Reguły $r3$ i $r4$

$$(Headache = yes)\&(Weakness = yes) \Rightarrow (Flu = yes)$$

$$(Temperature = High)\&(Headache = yes) \Rightarrow (Flu = yes)$$

- Reguły dotyczą konkluzji $(Flu = yes)$ $\{1, 2, 4, 5\}$;
- Konkluzja $\{1, 2, 4, 5\}$ nie jest w pełni pokryta regułami $r3$ i $r4$, gdyż pokrywają one tylko przypadki 1,2 i 4 (próbka 5 nie jest pokryta przez żadną regułę);
- Reguły $r3$ i $r4$ są spójne ze zbiorem danych.



Zbiór reguł dyskryminujący

Najczęstszym zadaniem indukcji reguł jest wyłonienie zestawu reguł R, który jest spójny i kompletny. Taki zestaw reguł R nazywany jest **dyskryminującym** [Michalski, 1983].

Zbiór dyskryminujący dla przykładu z grupą

$$(Headache = yes) \Rightarrow (Flu = yes)$$

$$(Temperature = High)\&(Weakness = yes) \Rightarrow (Flu = yes)$$

$$(Temperature = Normal)\&(Headache = no) \Rightarrow (Flu = no)$$

$$(Headache = no)\&(Weakness = no) \Rightarrow (Flu = no)$$

Inne podejścia

Silne reguły

Istnieje wiele innych rodzajów zasad, które są stosowane w indukcji reguł.
Na przykład, niektóre systemy indukują zestawy reguł składające się z **silnych** reguł, tj. takich, w których każda reguła obejmuje wiele przypadków.

Reguły asocjacyjne

Innym zadaniem jest indukowanie reguł asocjacyjnych, w których po obu stronach reguły, po lewej i prawej stronie, znajdują się atrybuty.
np.:

$$(Nuesea = yes) = (Headache = yes)$$

Algorytmy generowania zbioru reguł

Metoda zupełna “Dziel i rządź”.

Zarys metody

Dany jest zbiór przykładów uczących T.

1. Znajdź jedna regułę, która dobrze (według zadanego kryterium) pasuje do aktualnych danych uczących.
2. Usuń z T wszystkie przykłady pokrywane (pasujące do poprzednika (IF)) przez skonstruowaną regułę.
3. Jeżeli nadal są jakieś przykłady do pokrycia zacznij procedurę od początku.

Warunek zatrzymania jest słabszy (niż zbiór pusty), aby zapobiec generowaniu reguł pokrywających tylko mały podzbiór rekordów. Technikę „dziel i rządź” nazywa się także metodą pokryciową generowania reguł decyzyjnych.

Problemy

Dyskusja

W algorytmie jednym z problemów jest wykonanie kroku 1 (generowanie reguły). Powstało wiele praktycznych i skutecznych algorytmów generowania reguł.

Zagadnienie wygenerowania reguły z całej przestrzeni rozwiązań jest zadaniem przeszukiwania przestrzeni rozwiązań.

Algorytmy

Możliwe podejścia:

- general-to-specific search np. algorytmy CN2 i PRISM.
- directional general-to-specific search np. rodzina algorytmów AQ.
- pruned search (przycinanie) np. RIPPER.
- Metody redukcyjne (reduct based) LEM2
- Metody przeszukiwania ewolucyjnego takie jak algorytmy genetyczne, mrówkowe, rojowe.

Warunki początkowe

Zbiory danych wejściowych (uczący) są:

- wolne od błędów,
- atrybuty numeryczne zostały już zdyskredytowane,
- nie ma brakujących wartości
- są spójne.

Algorytmy indukcyjne reguł mogą być kategoryzowane jako

- globalne - przestrzeń wyszukiwania jest zbiorem wszystkich wartości atrybutów,
- lokalne - przestrzeń wyszukiwania jest zbiorem par atrybutów i wartości.

Algorytm LEM1
Założenia

Cechy

- LEM1 - Learning from Examples Module version 1;
- indukuje zbiory reguł dyskryminacyjnych;
- globalny algorytm indukcyjny;
- wykorzystuje zbiory przybliżone Pawlaka 1982.

Cechy

- Niech B będzie niepustym podzbiorem zbioru A wszystkich atrybutów;
- Niech U oznacza zbiór wszystkich przypadków (rekordów) trenujących;
- Zależność nierozróżnialności (indiscernibility) $IND(B)$ jest zależnością na zbiorze U określoną dla $x, y \in U$ jako $(x, y) \in IND(B)$ wtedy i tylko wtedy, gdy dla x i y wartości wszystkich atrybutów z B są identyczne;

Algorytm LEM1
Założenia

Case	Attributes				Decision
	Temperature	Headache	Weakness	Nausea	Flu
1	very_high	yes	yes	no	yes
2	high	yes	no	yes	yes
3	normal	no	no	no	no
4	normal	yes	yes	yes	yes
5	high	no	yes	no	yes
6	high	no	no	no	no
7	normal	no	yes	no	no

Cechy

- Zależność nierozróżnialności $IND(B)$ nazywane są elementarnymi zbiorami B.

Na przykład, dla $B = \{Temperature, Headache\}$, zbiory elementarne $IND(B)$ to $\{1\}, \{2\}, \{3, 7\}, \{4\}, \{5, 6\}$.

Rodzina wszystkich zbiorów elementów B będzie oznaczona jako B^*, $\{Temperature, Headache\}^* = \{\{1\}, \{2\}, \{3, 7\}, \{4\}, \{5, 6\}\}$.

Algorytm LEM1
Założenia

Dla decyzji/konkluzji d mówi się, że $\{d\}$ zależy od B wtedy i tylko wtedy, gdy

$$B^* \leq \{d\}^*$$

Globalne pokrycie $\{d\}$ jest podzbiorem B z A takim, że $\{d\}$ jest zależny od B i B jest minimalny w A.

Tak więc, globalne pokrycie $\{d\}$ jest obliczane przez porównanie B^* z $\{d\}^*$.

Algorytm LEM1
Krok 1 – obliczenia pojedynczego globalnego pokrycia

Algorithm to compute a single global covering
(**input**: the set $A$of all attributes, partition $\{d\}^*$ on U;
output: a single global covering R);
begin
compute partition A^*;
$P := A$;
$R := \emptyset$;
if $A^* \leq \{d\}^*$
then
begin
for each attribute a in A **do**
begin
$Q := P - \{a\}$;
compute partition Q^*;
if $Q^* \leq \{d\}^*$ **then** $P := Q$
end {for}
$R := P$
end {then}
end {algorithm}.

Algorytm LEM1
Krok 2 – wytworzenie reguł przez dropping conditions

W oparciu o globalne pokrycia wytwarzane są reguły.
Reguły są obliczane z wykorzystaniem techniki dropping conditions [Michalski, 1983]:
Zakładamy, że reguła ma postać:

$$W_1 \& W_2 \& \ldots \& W_n \Rightarrow D$$

Technika ta skanuje listę wszystkich warunków (W_i), od lewej do prawej, z próbą odrzucenia każdego warunku i sprawdzanie z tabelą, czy uproszczona reguła nie narusza spójności reguły dyskryminującej.

Algorytm LEM1
Przykład krok 1 – Wyszukanie globalnego pokrycia

Case	Attributes				Decision
	Temperature	Headache	Weakness	Nausea	Flu
1	very_high	yes	yes	no	yes
2	high	yes	no	yes	yes
3	normal	no	no	no	no
4	normal	yes	yes	yes	yes
5	high	no	yes	no	yes
6	high	no	no	no	no
7	normal	no	yes	no	no

Obliczenie A^*

$$\{Temperature, Headache, Weakness, Nausea\}^* =$$

$$= \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}\}$$

Obliczenie $\{d\}^*$

$$\{Flu\}^* = \{\{1, 2, 4, 5\}, \{3, 6, 7\}\}$$

Sprawdzenie:

$$\{Temperature, Headache, Weakness, Nausea\}^* \leq \{Flu\}^*$$

Algorytm LEM1
Przykład krok 1 – Wyszukanie globalnego pokrycia

Case	Attributes				Decision
	Temperature	Headache	Weakness	Nausea	Flu
1	very_high	yes	yes	no	yes
2	high	yes	no	yes	yes
3	normal	no	no	no	no
4	normal	yes	yes	yes	yes
5	high	no	yes	no	yes
6	high	no	no	no	no
7	normal	no	yes	no	no

Następnie sprawdzamy, czy po odrzuceniu atrybutu *Temperature* spełnione jest:

$$\{Headache, Weakness, Nausea\}^* \leq \{Flu\}^*$$

Niestety warunek nie jest spełniony, gdyż:

$$\{Headache, Weakness, Nausea\}^* = \{\{1\}, \{2\}, \{3, 6\}, \{4\}, \{5, 7\}\}$$

Następnie sprawdzamy, czy odrzucenie atrybutu *Headache* zachowuje zasadę:

$$\{Temperature, Weakness, Nausea\}^* \leq \{Flu\}^*$$

Warunek jest spełniony, gdyż:

$$\{Temperature, Weakness, Nausea\}^* = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}\}$$

Algorytm LEM1
Przykład krok 1 – Wyszukanie globalnego pokrycia

Case	Attributes				Decision
	Temperature	Headache	Weakness	Nausea	Flu
1	very_high	yes	yes	no	yes
2	high	yes	no	yes	yes
3	normal	no	no	no	no
4	normal	yes	yes	yes	yes
5	high	no	yes	no	yes
6	high	no	no	no	no
7	normal	no	yes	no	no

Dalsza redukcja sprawdza kolejną redukcję:

$$\{\mathit{Temperature}, \mathit{Nausea}\}^* = \{\{1\}, \{2\}, \{3, 7\}, \{4\}, \{5, 6\}\}$$

gdzie 5 i 6 nie dają tej samej decyzji, zatem zbiór atrybutów $\{\mathit{Temperature}, \mathit{Nausea}\}$ nie jest globalnym pokryciem.
Zatem sprawdzamy:

$$\{\mathit{Temperature}, \mathit{Weakness}\}^* = \{\{1\}, \{2, 6\}, \{3\}, \{4, 7\}, \{5\}\}$$

i również uzyskujemy:

$$\{\mathit{Temperature}, \mathit{Weakness}\}^* \not\leq \{\mathit{Flu}\}^*$$

Algorytym LEM1
Przykład krok 1 – Wyszukanie globalnego pokrycia

Case	Attributes				Decision
	Temperature	Headache	Weakness	Nausea	Flu
1	very_high	yes	yes	no	yes
2	high	yes	no	yes	yes
3	normal	no	no	no	no
4	normal	yes	yes	yes	yes
5	high	no	yes	no	yes
6	high	no	no	no	no
7	normal	no	yes	no	no

Globalne pokrycie

Pierwsze znalezione globalne pokrycie to następujący zbiór atrybutów:

$$\{Temperature, Weakness, Nausea\}$$

Algorytm LEM1
Przykład krok 2 – Wytworzenie reguł

Case	Attributes				Decision
	Temperature	Headache	Weakness	Nausea	Flu
1	very_high	yes	yes	no	yes
2	high	yes	no	yes	yes
3	normal	no	no	no	no
4	normal	yes	yes	yes	yes
5	high	no	yes	no	yes
6	high	no	no	no	no
7	normal	no	yes	no	no

Pierwszy przypadek z tabeli zakłada następującą regułę wstępną:

$$(Temperature = very\_high)\&(Weakness = yes)\&(Nausea = no)$$

$$\Rightarrow (Flu = yes)$$

Reguła opisuje tylko pierwszy rekord. Pierwszy warunek $(Temperature = very\_high)$ nie może zostać usunięty, gdyż wówczas reguła pokrywa rekordy 1 i 7, które mają różne konkluzje (decyzje).

Algorytm LEM1
Przykład krok 2 – Wytworzenie reguł

Case	Attributes				Decision
	Temperature	Headache	Weakness	Nausea	Flu
1	very_high	yes	yes	no	yes
2	high	yes	no	yes	yes
3	normal	no	no	no	no
4	normal	yes	yes	yes	yes
5	high	no	yes	no	yes
6	high	no	no	no	no
7	normal	no	yes	no	no

Jednakże, próba zrezygnowania z kolejnego warunku, $(Weakness = yes)$ jest skuteczna, ponieważ reguła

$$(Temperature = very\_high)\&(Nausea = no) \Rightarrow (Flu = yes)$$

opisuje tylko pierwszy rekord. Następna możliwość, aby zrezygnować z ostatniego warunku $(Nausea = no)$ jest również skuteczna, ponieważ wynikająca z tego reguła:

$$(Temperature = very\_high) \Rightarrow (Flu = yes)$$

nadal opisuje tylko 1-szy rekord.

Algorytm LEM1
Wynik

W podobny sposób indukowane są pozostałe reguły, aż do uzyskania spójności.

Ostateczny zbiór reguł

1. $(Temperature = very\_high) \Rightarrow (Flu = yes)$
2. $(Nausea = yes) \Rightarrow (Flu = yes)$
3. $(Temperature = high)\&(Weakness = yes) \Rightarrow (Flu = yes)$
4. $(Weakness = no)\&(Nausea = no) \Rightarrow (Flu = no)$
5. $(Temperature = normal)\&(Nausea = no) \Rightarrow (Flu = no)$

Istnieje też drugie globalne pokrycie z atrybutów:

$$\{Temperature, Headache, Weakness\}$$

Wymagania

Eksploracja danych zajmuje się też wskazywaniem reguł asocjacyjnych (skojarzeniowych).

- Atrybuty muszą być dyskretne (najłatwiej binarne).
- Związane z danymi transakcyjnymi, takimi jak dokumentacja biznesowa, handlowa, usługowa lub medyczna, w której występowanie cech ($item_1$) utożsamia się z występowaniem produktu ($item_2$).
- Nazywana analizą koszykową (ang. Market Basket Analysis). Jest to technika używana przez dużych detalistów do odkrywania związków pomiędzy kupowanymi przedmiotami.
- Związki wykrywa się poprzez szukanie kombinacji przedmiotów, które często występują razem w transakcjach. Uzyskujemy w ten sposób wiedzę o zachowaniach klientów.

Podejście
Dyskusja

W praktycznych zastosowaniach nie wykorzystuje się skomplikowanych metod znajdowania reguł. Zgrubne oszacowanie pokazuje, ze dla n atrybutów może istnieć $O(3^n)$ reguł. Realne oszacowanie $O(n \cdot 2^{n-1})$ też jest pesymistyczne, bo z założenia zajmujemy się dużymi zbiorami danych.

Cechy

W większości praktycznych algorytmów znajdowania reguł asocjacyjnych proces ten składa się z dwóch kroków:

1. Znajdź zbiór częstych wzorców (ang. frequent item-sets) dla danych (transakcji). (frequent item-set to reguła asocjacyjna bez następnika, tj. koniunkcja warunków na występowanie atrybutów (items))
2. Na podstawie zbioru częstych wzorców wyznacz zbiór „dobrych” reguł asocjacyjnych.

Przykład

Przykład zbioru danych:

Example

Table: Transakcje spożywcze

$t1$	$\{pomidor, ogorek, feta\}$
$t2$	$\{pomidor, szynka\}$
$t3$	$\{chleb, szynka\}$
$t4$	$\{pomidor, ogorek, szynka\}$
$t5$	$\{pomidor, ogorek, cebula, szynka, feta\}$
$t6$	$\{ogorek, cebula, feta\}$
$t7$	$\{ogorek, feta, cebula\}$

Jeśli popatrzymy na zbiór danych, to można zauważyć wzorce:

1. Większość osób, które kupują pomidory kupuje szynkę.
2. Większość ludzi, którzy kupują ogórka, kupuje również ????.
3. Większość ludzi, którzy kupują ???? kupuje również ????.

Zbiór przedmiotów *I*
Definicja

W tabeli widzimy siedem transakcji ze sklepu spożywczego. Każda transakcja pokazuje przedmioty zakupione w tej transakcji.

Definicja zbioru przedmiotów

Możemy reprezentować przedmioty jako zestaw *items* w następujący sposób:

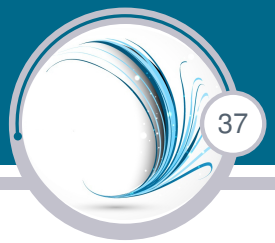
$$I = \{i_1, i_2, \ldots, i_k\}$$

gdzie k - liczba przedmiotów

co odpowiada:

Example

$$I = \{pomidor, ogorek, feta, szynka, chleb, cebula\}$$



Reguła asocjacyjna

Reguła asocjacyjna jest zdefiniowana jako implikacja:

$$X \Rightarrow Y,$$

where $X \subset I$, $Y \subset I$ and $X \cap Y = 0$,

Example

$$\{pomidor, ogorek\} \Rightarrow \{feta\}$$

Wsparcie - (support)
Definicja

Support - definicja

Wsparcie wskazuje, jak często dany zestaw danych pojawia się w zbiorze transakcji (uczącym).

$$\text{supp}(X \Rightarrow Y) = \frac{|X \cup Y|}{n} \tag{1}$$

Innymi słowy, jest to liczba transakcji zawierających X i Y podzielona przez łączną liczbę transakcji. Reguły nie są użyteczne jeżeli wartość wsparcia jest niska.

Example

$t1$	$\{pomidor, ogorek, feta\}$
$t2$	$\{pomidor, szynka\}$
$t3$	$\{chleb, szynka\}$
$t4$	$\{pomidor, ogorek, szynka\}$
$t5$	$\{pomidor, ogorek, cebula, szynka, feta\}$
$t6$	$\{ogorek, cebula, feta\}$
$t7$	$\{ogorek, feta, cebula\}$

1. $\text{supp}(\{pomidor\} \Rightarrow \{ogorek\}) = \frac{3}{7} = 43\%$
2. $\text{supp}(\{ogorek\} \Rightarrow \{feta\}) = \frac{4}{7} = 57\%$
3. $\text{supp}(\{pomidor, ogorek\} \Rightarrow \{feta\}) = \frac{2}{7} = 28\%$

Wsparcie - (support)
Dyskusja

Wsparcie jest ważną miarą, ponieważ reguła, która ma bardzo niskie wsparcie, może pojawić się po prostu przez przypadek. Co więcej, reguła o niskim wsparciu jest również mniej interesująca z perspektywy biznesowej dlatego, że promowanie przedmiotów, które klienci rzadko kupują razem, może nie być opłacalne.

Zaufanie (confidence)
Definicja

confidence - definicja

Dla reguły $X \Rightarrow Y$ zaufanie to odsetek w jakim Y jest kupowany z X. Wskazuje jak często reguła jest prawdziwa.

$$\mathrm{conf}(X \Rightarrow Y) = \frac{\mathrm{supp}(X \cup Y)}{\mathrm{supp}(X)} \tag{2}$$

Example

Reguła $\mathrm{conf}(\{pomidor\} \Rightarrow \{ogorek\}) = \frac{3/7}{4/7} = 3/4$ ma zaufanie na poziomie 3/4, co oznacza, że dla 75% transakcji zawierających pomidory reguła jest prawidłowa (75% razy klient kupując pomidory, kupi również ogórek).

Zaufanie (confidence)
Obliczenia

Trzy przykłady:

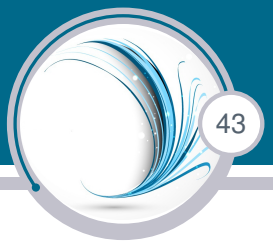
Example

$t1$	$\{pomidor, ogorek, feta\}$
$t2$	$\{pomidor, szynka\}$
$t3$	$\{chleb, szynka\}$
$t4$	$\{pomidor, ogorek, szynka\}$
$t5$	$\{pomidor, ogorek, cebula, szynka, feta\}$
$t6$	$\{ogorek, cebula, feta\}$
$t7$	$\{ogorek, feta, cebula\}$

1. $\mathrm{conf}(\{ogorek\} \Rightarrow \{feta\}) = \frac{4/7}{5/7} = \frac{4}{5} = (80\%)$
2. $\mathrm{conf}(\{pomidor\} \Rightarrow \{feta\}) = \frac{2/7}{4/7} = 50\%$
3. $\mathrm{conf}(\{pomidor, ogorek\} \Rightarrow \{feta\}) = \frac{2/7}{3/7} = 66\%$

Zaufanie (confidence)
Obliczenia

Zaufanie mierzy wiarygodność wnioskowania przedstawionego przez regułę. Wysoki poziomem ufności w przypadku reguły $\{A\} \Rightarrow \{B\}$ wskazuje, że pojawienie się B jest bardziej prawdopodobne razem z A.



Lift - definicja

Wzrost w regule jest stosunkiem obserwowanego do oczekiwanego wsparcia, jeśli X i Y były niezależne, i jest definiowane jako:

$$\text{lift}(X \Rightarrow Y) = \frac{\text{supp}(X \cup Y)}{\text{supp}(X) \cdot \text{supp}(Y)} \tag{3}$$

Większe wartości wzrostu wskazują na silniejsze kojarzenie.

Wzrost (lift)
Obliczenia

Oto kilka przykładów:

Example

$t1$	$\{pomidor, ogorek, feta\}$
$t2$	$\{pomidor, szynka\}$
$t3$	$\{chleb, szynka\}$
$t4$	$\{pomidor, ogorek, szynka\}$
$t5$	$\{pomidor, ogorek, cebula, szynka, feta\}$
$t6$	$\{ogorek, cebula, feta\}$
$t7$	$\{ogorek, feta, cebula\}$

1. $\text{lift}(\{pomidor\} \Rightarrow \{ogorek\}) = \frac{3/7}{(4/7)(5/7)} = 1.05$
2. $\text{lift}(\{ogorek\} \Rightarrow \{feta\}) = \frac{4/7}{(5/7)(4/7)} = 1.4$
3. $\text{lift}(\{pomidor\} \Rightarrow \{feta\}) = \frac{2/7}{(4/7)(4/7)} = 0.875$
4. $\text{lift}(\{pomidor, ogorek\} \Rightarrow \{feta\}) = \frac{3/7}{(4/7)(5/7)} = 1.17$

Wzrost (lift)
Interpretacja

1. Jeśli $\text{lift} = 1$, oznacza to, że możliwość wystąpienia Poprzednika i Następnika nie są od siebie zależne.
2. Jeśli $\text{lift} < 1$, oznacza to, że wystąpienie Poprzednika ma negatywny wpływ na wystąpienie Następnika i vice versa.
3. Jeśli $\text{lift} > 1$, oznacza to, że te dwa zdarzenia są od siebie zależne i te reguły są bardzo przydatne do określenia Następnika. Daje nam to również informację o tym, w jakim stopniu zdarzenia te są od siebie wzajemnie

Pewność (conviction)
Definicja

Conviction

Definicja
Pewność o regule definiujemy jako:

$$\mathrm{conv}(X \Rightarrow Y) = \frac{1 - \mathrm{supp}(Y)}{1 - \mathrm{conf}(X \Rightarrow Y)} \tag{4}$$

Pewność można interpretować jako stosunek oczekiwanej częstotliwości występowania X bez Y jeżeli X i Y były niezależne, podzielony przez obserwowaną częstość występowania błędnych prognoz (predykcji). Wysoka wartość współczynnika oznacza, że konsekwencja (then) zależy w dużym stopniu od przesłanki (if).

Pewność (conviction)
Przykłady

Oto kilka przykładów:

Example

$t1$	$\{pomidor, ogorek, feta\}$
$t2$	$\{pomidor, szynka\}$
$t3$	$\{chleb, szynka\}$
$t4$	$\{pomidor, ogorek, szynka\}$
$t5$	$\{pomidor, ogorek, cebula, szynka, feta\}$
$t6$	$\{ogorek, cebula, feta\}$
$t7$	$\{ogorek, feta, cebula\}$

1. $\text{conv}(\{pomidor\} \Rightarrow \{ogorek\}) = \frac{1-5/7}{1-3/4} = 1.14$
2. $\text{conv}(\{ogorek\} \Rightarrow \{feta\}) = \frac{1-4/7}{1-4/5} = 2.14$
3. $\text{conv}(\{pomidor\} \Rightarrow \{feta\}) = \frac{1-4/7}{1-1/2} = 0.86$
4. $\text{conv}(\{pomidor, ogorek\} \Rightarrow \{feta\}) = \frac{1-4/7}{1-2/3} = 1.28$

Przykład
omówienie

$$Rule:\ X \Rightarrow Y$$

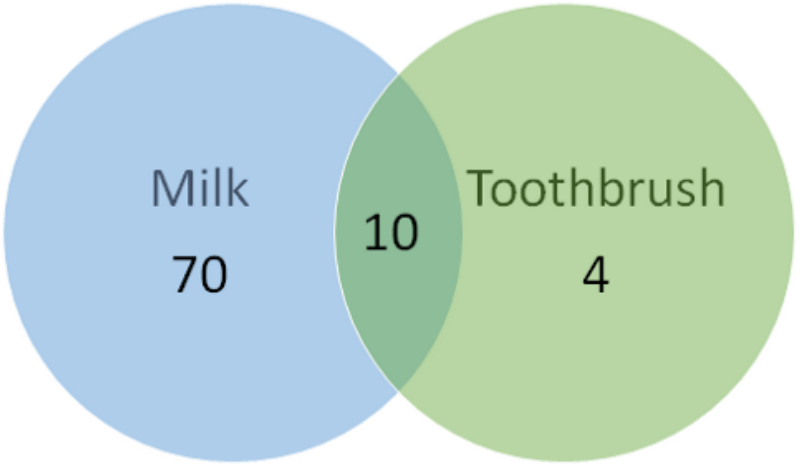
$$Support = \frac{frq(X,Y)}{N}$$

$$Confidence = \frac{frq(X,Y)}{frq(X)}$$

$$Lift = \frac{Support}{Supp(X) \times Supp(Y)}$$

Rule	Support	Confidence	Lift
$A \Rightarrow D$	2/5	2/3	10/9
$C \Rightarrow A$	2/5	2/4	5/6
$A \Rightarrow C$	2/5	2/3	5/6
$B \& C \Rightarrow D$	1/5	1/3	5/9

Przykład
omówienie Confidence i Lift

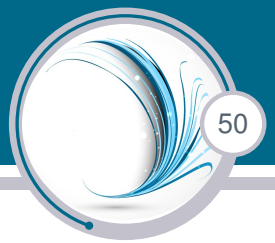


$$\mathrm{conf}\{\textit{toothbrush}\} \Rightarrow \{\textit{Milk}\} = 10/(10+4) = 0,7$$

Zaufanie jest duże, ale intuicyjnie czujemy, że te dwa produkty słabo się kojarzą. Aby ująć tę liczbę w odpowiedniej perspektywie, weź pod uwagę prawdopodobieństwo obecności mleka w koszyku bez wiedzy o szczoteczce $80/100 = 0,8$.

$$\mathrm{lift}\{\textit{toothbrush}\} \Rightarrow \{\textit{Milk}\} = 0.7/0.8 = 0.87$$

Liczby pokazują, że posiadanie szczoteczki do zębów w rzeczywistości zmniejsza prawdopodobieństwo posiadania mleka do 0,7 z 0,8! Będzie to wzrost o 0,7/0,8 = 0,87.



Własność Apriori

zbudowana na prostym przekonaniu, że wszystkie podzbiory częstego zbioru przedmiotów muszą być również częste.

Na przykład zbiór $\{pomidor, ogrek, feta\}$, może być częsty, wtedy i tylko wtedy, gdy on sam i wszystkie jego podzbiory pojedynczych elementów, par i trójek występują często.

Algorytm Apriori jest przeznaczony do znajdowania wzorców w dużych zbiorach danych. Jeśli jakiś wzór zdarza się często, jest on uważany za „interesujący”.

Algorytm Apriori
Omówienie

W przypadku dużych zbiorów danych, w setkach tysięcy transakcji mogą znajdować się setki przedmiotów. Algorytm Apriori próbuje wyodrębnić reguły dla każdej możliwej kombinacji przedmiotów. Na przykład dla i_1 i i_2, i_1 i i_3, i_1 i i_4, a następnie i_2 i i_3, i_2 i i_4, a następnie kombinacji przedmiotów np. i_1, i_2 i i_3; podobnie i_1, i_2 i i_4 itd.

Jak widać z powyższego przykładu, proces ten może być bardzo powolny ze względu na liczbę kombinacji.

Algorytm Apriori
Założenia

Aby przyspieszyć ten proces, musimy wykonać następujące kroki:

1. Ustalić minimalną wartość supp i conf. Oznacza to, że interesujące jest tylko znalezienie reguł dla przedmiotów, które mają pewien domyślny poziom istnienia (wsparcie) i mają minimalną wartość dla współwystępowania z innymi przedmiotami (zaufanie).
2. Wyodrębnić wszystkie podzbiory, które mają wyższą wartość wsparcia niż minimalny próg.
3. Wybrać wszystkie reguły z podzbiorów o wartości zaufania wyższej niż minimalny próg.
4. Uporządkować reguły w kolejności malejącej po parametrze lift.

Dwie części algorytmu
Wsparcie i zaufanie

Znajdowanie częstych wzorców *min_supp*

Interesujące jest znalezienie zbioru częstych wzorców, czyli takich które mają wsparcie (support) powyżej ustalonego progu *min_supp*.

Znajdowanie odpowiednich reguł *min_conf*

W generowaniu reguł wymagane jest by poziom zaufania *confidence* dla tworzonej reguły był powyżej założonego progu *min_conf*.

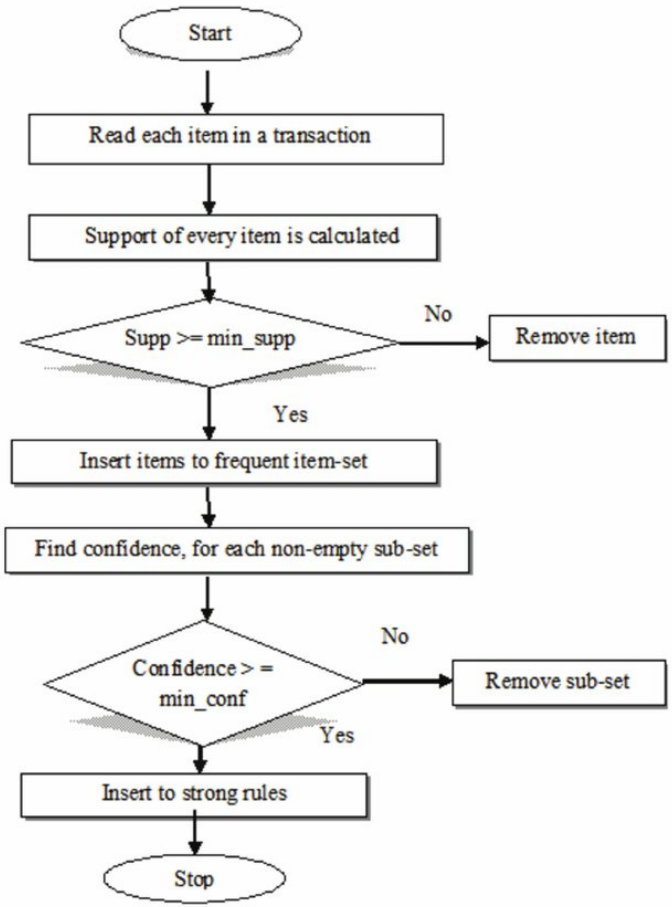
Algorytm Apriori
Krok 1

1. Odfiltrować wszystkie zbiory z minimalnym progiem wsparcia. Odbywa się to iteracyjnie poprzez zwiększenie wielkości zbiorów.
2. W pierwszej iteracji obliczamy wsparcie pojedynczych *i*. W następnej iteracji obliczamy wsparcie dla par i tak dalej.
3. Zbiory przechodzące iterację mogą być uznane za kandydatów do następnej iteracji. Na przykład: jeśli {A}, {B}, {C} są częste, ale {D} nie jest częste w pierwszej iteracji, wtedy w drugiej iteracji rozważamy tylko wsparcie par {A, B}, {A,C}, {B,C}, ignorując wszystkie pary zawierające {D} .
4. W trzeciej iteracji, jeżeli {A,C}, i {B,C} występują często, ale {A,B} nie występuje, to algorytm może zakończyć się, ponieważ wsparcie {A,B,C} jest trywialne (nie przekracza progu wsparcia), biorąc pod uwagę, że {A,B} nie było wystarczająco częste.

Algorytm Apriori
Krok 2

1. Używając zbiorów wybranych w Kroku 1, wygeneruj nowe reguły z zaufaniem większym niż ustalony wcześniej minimalny próg ufności.
2. Kandydackie zbiory atrybutów, które przeszły Krok 1, będą zawierały wszystkie często występujące zbiory atrybutów.
3. Na przykład dla zbioru items {A, C} z wysokim support, obliczylibyśmy confidence dla $\{A\} \Rightarrow \{C\}$ i $\{C\} \Rightarrow \{A\}$ i porównalibyśmy je z minimalnym progiem zaufania.
4. Reguły, które przetrwały, to te z poziomem zaufania przekraczającym minimalny próg.

Algorytm Apriori
Schemat



Źródła

- Jerzy W. Grzymala-Busse, „Rule Induction”, Data Mining and Knowledge Discovery Handbook pp 249-265, 2010
- Ivo D. Dinov. „Data Science and Predictive Analytics. Biomedical and Health Applications using R”, 2018, Springer
- Mandeep Mittal ‚Efficient Ordering Policy for Imperfect Quality Items Using Association Rule Mining”
- Tan, Steinbach, Karpatne, Kumar, „Introduction to Data Mining” `https://www-users.cs.umn.edu/~kumar001/dmbook/ch5_association_analysis.pdf`

Dziękuję za uwagę
Czas na pytania ????