GDAŃSK UNIVERSITY OF TECHNOLOGY

# Eksploracja danych

## REGUŁY ASOCJACYJNE


***Wojciech Waloszek***

*wowal@eti.pg.gda.pl*

***Teresa Zawadzka***

*tegra@eti.pg.gda.pl*

*Katedra Inżynierii Oprogramowania*
*Wydział Elektroniki, Telekomunikacji i Informatyki*
*Politechnika Gdańska*


GDAŃSK UNIVERSITY OF TECHNOLOGY

# Budowa reguł asocjacyjnych

- Przy budowie reguł asocjacyjnych nie mamy wyróżnionego atrybutu decyzyjnego,
- Próbujemy uchwycić różne zależności między atrybutami bez wyróżniania żadnego z nich,
- Naiwne podejście mogłoby polegać na wygenerowaniu klasyfikatora decyzyjnego dla każdego z atrybutów – jest ono jednak skrajnie nieefektywne

GDAŃSK UNIVERSITY OF TECHNOLOGY

# Algorytm Apriori

- Algorytm działa jedynie na atrybutach nominalnych,
- Dla generowanych reguł ustalamy progi pokrycia i poprawności ($s_{min}$ i $a_{min}$),
- Algorytm ten buduje reguły tworząc testy złożone,
- Na początek brane pod uwagę jest jedynie pokrycie testu, które musi przekraczać ustalony próg minimalny

GDAŃSK UNIVERSITY OF TECHNOLOGY

# Apriori – przykład

- Próg pokrycia ustalamy na $s_{min}$=2,
- Najpierw bierzemy po uwagę testy proste:

```
S.C.=S
S.C.=M
Wykształcenie=podstawowe
Wykształcenie=średnie
Wykształcenie=wyższe
Sam.=tak
Sam.=nie
Z.K.=tak
Z.K.=nie
```

GDAŃSK UNIVERSITY OF TECHNOLOGY

# Obliczanie pokrycia testu

| S.C. | D.O.R. | Wiek | Wykształcenie | Sam. | Z.K. |
|---|---|---|---|---|---|
| S | 800 | 32 | wyższe | tak | tak |
| S | 1200 | 35 | średnie | tak | tak |
| S | 700 | 26 | podstawowe | nie | nie |
| M | 600 | 45 | wyższe | nie | tak |
| M | 650 | 38 | średnie | tak | tak |
| S | 900 | 28 | wyższe | nie | nie |
| S | 1100 | 65 | średnie | tak | nie |
| M | 500 | 22 | średnie | nie | nie |
| S | 800 | 43 | podstawowe | tak | nie |

**S.C.=S**

**Pokrycie: s = 6**

GDAŃSK UNIVERSITY OF TECHNOLOGY

# Apriori – testy proste

- Otrzymujemy następujące pokrycia:

| | s |
|---|---|
| S.C.=S | 6 |
| S.C.=M | 3 |
| Wykształcenie=podstawowe | 2 |
| Wykształcenie=średnie | 4 |
| Wykształcenie=wyższe | 3 |
| Sam.=tak | 5 |
| Sam.=nie | 4 |
| Z.K.=tak | 4 |
| Z.K.=nie | 5 |

Wszystkie testy proste przekraczają próg pokrycia
i mogą być wykorzystane do budowy testów złożonych

GDAŃSK UNIVERSITY OF TECHNOLOGY

# Apriori – testy złożone

- Testy podwójne budujemy z testów pojedynczych z poprzedniego kroku (oba składowe testy pojedyncze muszą mieć minimalne pokrycie),
- Pary budujemy według zasady każdy-z-każdym (oprócz odrzuconych):

```
S.C.=S and Wykształcenie=podstawowe
S.C.=S and Wykształcenie=średnie
S.C.=S and Wykształcenie=wyższe
S.C.=S and Sam.=tak
...
```

# Apriori – testy złożone (2)

- Następnie badamy pokrycia testów:

```
                                                    s
S.C.=S and Wykształcenie=podstawowe                 2
S.C.=S and Wykształcenie=średnie                    2
S.C.=S and Wykształcenie=wyższe                     2
...
S.C.=M and Wykształcenie=wyższe                     1
...
```

Spośród 30 testów złożonych kryterium pokrycia spełnia 19

GDAŃSK UNIVERSITY OF TECHNOLOGY

# Apriori – generacja reguł

- Otrzymaliśmy 19 testów złożonych spełniających warunek minimalnego pokrycia,
- Z każdego testu możemy spróbować wygenerować reguły; dla testu:

```
S.C.=S and Wykształcenie=podstawowe
```

mogą to być:

```
if S.C.=S then Wykształcenie=podstawowe
if Wykształcenie=podstawowe then S.C.=S
```

GDAŃSK UNIVERSITY OF TECHNOLOGY

# Apriori – generacja reguł (2)

- Reguły akceptujemy tylko jeżeli przekraczają one próg poprawności (ustalmy $a_{min}$=1):

| | *a* |
|---|---|
| ~~`if S.C.=S then Wykształcenie=podstawowe`~~ | ~~`2/6`~~ |
| `if Wykształcenie=podstawowe then S.C.=S` | `2/2` |

Kryterium poprawności spełnia tylko jedna z wygenerowanych reguł

GDAŃSK UNIVERSITY OF TECHNOLOGY

# Apriori – generacja reguł (3)

- Z 19 testów podwójnych możemy wygenerować tylko 2 reguły spełniające progi poprawności:

```
if Wykształcenie=podstawowe then S.C.=S
if Wykształcenie=podstawowe then Z.K.=nie
```

- Po wygenerowaniu tych reguł możemy przystąpić do generacji testów potrójnych

GDAŃSK UNIVERSITY OF TECHNOLOGY

# Apriori – testy potrójne

- Testy potrójne budujemy z testów podwójnych z poprzedniego kroku (wszystkie składowe testy podwójne muszą mieć minimalne pokrycie):

**`S.C.=S and Sam.=nie and Z.K.=tak`**

na pewno nie spełnia kryterium pokrycia, bo nie spełnia go test podwójny:

**`Sam.=nie and Z.K.=tak`**

GDAŃSK UNIVERSITY OF TECHNOLOGY

# Apriori – testy potrójne (2)

- Kryterium pokrycia spełniają następujące testy potrójne:

```
Wykształcenie=wyższe and Sam.=tak and Z.K.=tak
S.C.=S and Sam.=nie and Z.K.=nie
S.C.=S and Sam.=tak and Z.K.=nie
S.C.=S and Sam.=tak and Z.K.=tak
S.C.=S and Wykształcenie=wyższe and Sam.=tak
S.C.=S and Wykształcenie=podstawowe and Z.K.=tak
```

- Dla każdego testu generowane są reguły, dla których sprawdzane jest kryterium poprawności

GDAŃSK UNIVERSITY OF TECHNOLOGY

# Apriori – przykład

- Dla testu:

```
Wykształcenie=wyższe and Sam.=tak and Z.K.=tak
```

generowane są następujące reguły:

```
if Wykształcenie=wyższe then Sam.=tak and Z.K.=tak
if Sam.=tak then Wykształcenie=wyższe and Z.K.=tak
if Z.K.=tak then Wykształcenie=wyższe and Sam.=tak
if Wykształcenie=wyższe and Sam.=tak then Z.K.=tak
if Wykształcenie=wyższe and Z.K.=tak then Sam.=tak
if Sam.=tak and Z.K.=tak then Wykształcenie=wyższe
```

Tylko jedna spełnia kryterium poprawności

GDAŃSK UNIVERSITY OF TECHNOLOGY

# Apriori – generacja reguł

- Z 6 testów podwójnych generujemy 8 reguł spełniających próg poprawności:

```
if Wykształcenie=wyższe then Sam.=tak and Z.K.=tak
if S.C.=S and Sam.=nie then Z.K.=nie
if Sam.=tak and Z.K.=nie then S.C.=S
if S.C.=S and Z.K.=tak then Sam.=tak
if S.C.=S and Wykształcenie=wyższe then Sam.=tak
if Wykształcenie=postawowe then S.C.=S and Z.K.=nie
if S.C.=S and Wykształcenie=postawowe then Z.K.=nie
if Wykształcenie=postawowe and Z.K.=nie then S.C.=S
```

- Po wygenerowaniu tych reguł możemy przystąpić do generacji testów poczwórnych

GDAŃSK UNIVERSITY OF TECHNOLOGY

# Apriori – testy poczwórne

- Testy poczwórne budujemy z testów potrójnych z poprzedniego kroku (wszystkie składowe testy podwójne muszą mieć minimalne pokrycie),
- W podanym przykładzie nie ma testów poczwórnych spełniających warunek minimalnego pokrycia,
- Oznacza to, że algorytm Apriori kończy pracę

GDAŃSK UNIVERSITY OF TECHNOLOGY

# Apriori – wynik

- Ostatecznie wygenerowano 10 reguł:

```
if Wykształcenie=wyższe then Sam.=tak and Z.K.=tak
if S.C.=S and Sam.=nie then Z.K.=nie
if Sam.=tak and Z.K.=nie then S.C.=S
if S.C.=S and Z.K.=tak then Sam.=tak
if S.C.=S and Wykształcenie=wyższe then Sam.=tak
if Wykształcenie=postawowe then S.C.=S and Z.K.=nie
if S.C.=S and Wykształcenie=postawowe then Z.K.=nie
if Wykształcenie=postawowe and Z.K.=nie then S.C.=S
if Wykształcenie=podstawowe then S.C.=S
if Wykształcenie=podstawowe then Z.K.=nie
```

Wszystkie spełniają kryteria poprawności i pokrycia

GDAŃSK UNIVERSITY OF TECHNOLOGY

# Algorytm Apriori

**Wejście:** $s_{min}$ – próg pokrycia,
$a_{min}$ – próg poprawności,
$P$ – zbiór przykładów
**Wyjście:** $R$ – zbiór reguł

1. $n := 1$, $R := \{\}$
2. Generuj zbiór testów pojedynczych $T_1$
3. Eliminuj testy o pokryciu mniejszym od $s_{min}$
4. Dopóki $T_n$ jest niepusty wykonuj:
5. $n := n + 1$
6. Generuj zbiór testów $n$-krotnych $T_n$
7. Eliminuj z $T_n$ testy o pokryciu mniejszym od $s_{min}$
8. Dla każdego testu $t$ z $T_n$:
9. Generuj zbiór $R_t$ reguł dla testu $t$
10. Eliminuj z $R_t$ reguły o poprawności mniejszej od $a_{min}$
11. $R := R \cup R_t$
12. Koniec pętli wewnętrznej
13. Koniec pętli zewnętrznej

GDAŃSK UNIVERSITY OF TECHNOLOGY

# Apriori – drugi przykład

- Prześledźmy jeszcze działanie algorytmu dla $s_{min} = 4$ i $a_{min} = 0,8$

# Apriori – drugi przykład (2)

- Dla testów prostych otrzymujemy następujące pokrycia:

| | s |
|---|---|
| S.C.=S | 6 |
| S.C.=M | 3 |
| Wykształcenie=podstawowe | 2 |
| Wykształcenie=średnie | 4 |
| Wykształcenie=wyższe | 3 |
| Sam.=tak | 5 |
| Sam.=nie | 4 |
| Z.K.=tak | 4 |
| Z.K.=nie | 5 |

Testy niespełniające kryterium pokrycia możemy wyeliminować.

GDAŃSK UNIVERSITY OF TECHNOLOGY

# Apriori – drugi przykład (3)

- Testy podwójne budujemy z testów pojedynczych z poprzedniego kroku (oba składowe testy pojedyncze muszą mieć minimalne pokrycie),
- Po odrzuceniu testów niespełniających kryterium pokrycia otrzymujemy dwa testy podwójne:

```
S.C.=S and Sam.=tak
S.C.=S and Z.K.=nie
```

GDAŃSK UNIVERSITY OF TECHNOLOGY

# Apriori – drugi przykład (4)

- Z testów podwójnych możemy wygenerować 2 reguły spełniające progi poprawności:

```
if Z.K.=nie then S.C.=S

if Sam.=tak then S.C.=S
```

- Po wygenerowaniu tych reguł możemy przystąpić do generacji testów potrójnych

GDAŃSK UNIVERSITY OF TECHNOLOGY

# Apriori – drugi przykład (5)

- Żaden z testów potrójnych nie spełnia kryterium minimalnego pokrycia, zatem ostateczny wynik pracy algorytmu to:

```
if Z.K.=nie then S.C.=S
if Sam.=tak then S.C.=S
```

GDAŃSK UNIVERSITY OF TECHNOLOGY

# Apriori – komentarz

- Apriori w większości przypadków pozwala stosunkowo szybko (mimo teoretycznie wykładniczej złożoności) wygenerować reguły asocjacyjne,
- Liczba generowanych reguł jest zazwyczaj bardzo duża, zatem zalecane jest rozpoczęcie pracy do wysokiego progu pokrycia i redukowanie go w trakcie analizy

GDAŃSK UNIVERSITY OF TECHNOLOGY

# Dziękujemy za uwagę

Zapraszamy na wykład:

**ANALIZA SZEREGÓW CZASOWYCH**