

5 Reguły asocjacyjne.

Problem koszyka zakupowego.

Mając bazę danych transakcji, znajdź reguły, które będą przewidywać wystąpienie pozycji na podstawie wystąpienia innych pozycji w transakcji.

Market-Basket transactions

<i>TID</i>	<i>Items</i>
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Example of Association Rules

$\{\text{Diaper}\} \rightarrow \{\text{Beer}\},$
 $\{\text{Milk, Bread}\} \rightarrow \{\text{Eggs, Coke}\},$
 $\{\text{Beer, Bread}\} \rightarrow \{\text{Milk}\}$

Implication here means
co-occurrence, not causality!

Biorąc pod uwagę bazę danych transakcji, gdzie każda transakcja jest zbiorem przedmiotów (zakupionych przez klienta w sklepie), znajdź wszystkie reguły, które korelują obecność jednego zbioru przedmiotów z innym zestawem przedmiotów

Przykład:

30% wszystkich transakcji dotyczących pieluch zawiera również piwo;

5% wszystkich transakcji zawiera te pozycje

- 30%: zaufanie do reguły.

- 5%: nośnik reguły

Jesteśmy zainteresowani znalezieniem wszystkich zasad, zamiast sprawdzać, czy obowiązuje dana reguła.

Pojęcie często występującego zestawu przedmiotów.

- Zestaw przedmiotów

- Zbiór jednego lub więcej elementów

Przykład: {mleko, chleb, pielucha}

- k -zestaw: zestaw przedmiotów zawierający k pozycji

- Liczność nośnika (σ)

- Częstotliwość występowania zestawu przedmiotów

Przykład: $\sigma(\{\text{mleko, chleb, pielucha}\}) = 2$

- Nośnik

- Ułamek transakcji, które zawierają zestaw przedmiotów

Przykład: $s(\{\text{mleko, chleb, pielucha}\}) = 2/5$

- Często występujący zestaw przedmiotów

- Zestaw przedmiotów, którego nośnik jest większy lub równy ustalonemu progowi minimalnemu

Pojęcie reguły asocjacyjnej.

- Reguła asocjacyjna

- Implikacja postaci $X \rightarrow Y$, gdzie X i Y są zestawami przedmiotów

Przykład: $\{\text{Mleko, Pielucha}\} \rightarrow \{\text{Piwo}\}$

- Miara oceny reguł

- Nośnik (s)

Ułamek transakcji, które zawierają zarówno X , jak i Y

- Zaufanie (c)

Mierzy, jak często pozycje w Y pojawiają się w transakcjach zawierających X

<i>TID</i>	<i>Items</i>
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Example:

$\{\text{Milk, Diaper}\} \Rightarrow \text{Beer}$

$$s = \frac{\sigma(\text{Milk, Diaper, Beer})}{|T|} = \frac{2}{5} = 0.4$$

$$c = \frac{\sigma(\text{Milk, Diaper, Beer})}{\sigma(\text{Milk, Diaper})} = \frac{2}{3} = 0.67$$

Eksplorowanie reguł asocjacyjnych.

<i>TID</i>	<i>Items</i>
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Example of Rules:

$\{\text{Milk, Diaper}\} \rightarrow \{\text{Beer}\}$ ($s=0.4, c=0.67$)
 $\{\text{Milk, Beer}\} \rightarrow \{\text{Diaper}\}$ ($s=0.4, c=1.0$)
 $\{\text{Diaper, Beer}\} \rightarrow \{\text{Milk}\}$ ($s=0.4, c=0.67$)
 $\{\text{Beer}\} \rightarrow \{\text{Milk, Diaper}\}$ ($s=0.4, c=0.67$)
 $\{\text{Diaper}\} \rightarrow \{\text{Milk, Beer}\}$ ($s=0.4, c=0.5$)
 $\{\text{Milk}\} \rightarrow \{\text{Diaper, Beer}\}$ ($s=0.4, c=0.5$)

- Wszystkie powyższe reguły są binarnymi partycjami tego samego zestawu przedmiotów: {mleko, pielucha, piwo}
- Reguły pochodzące z tego samego zestawu przedmiotów mają identyczne nośniki, ale mogą mieć różną pewność
- W ten sposób możemy oddzielić wymagania dotyczące nośnika i zaufania

Znajdowanie reguł asocjacyjnych.

Podejście dwuetapowe:

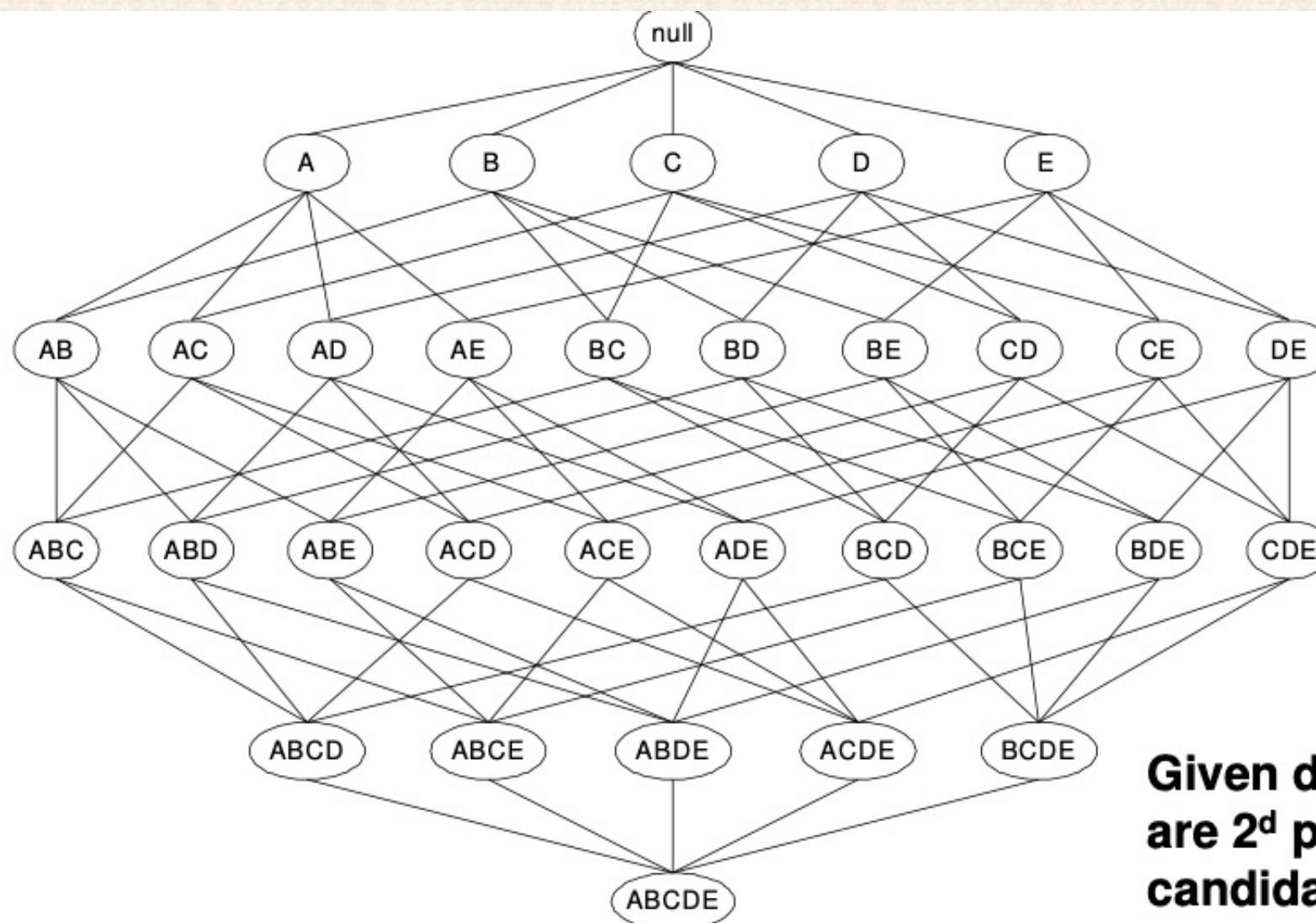
1. Generowanie zestawów częstych przedmiotów

Wygeneruj wszystkie zestawy elementów, których nośnik $\geq$ minsup

2. Generowanie reguł

Generuj reguły wysokiego zaufania z każdego częstego zestawu elementów, gdzie każda reguła jest binarnym podziałem częstego zestawu elementów

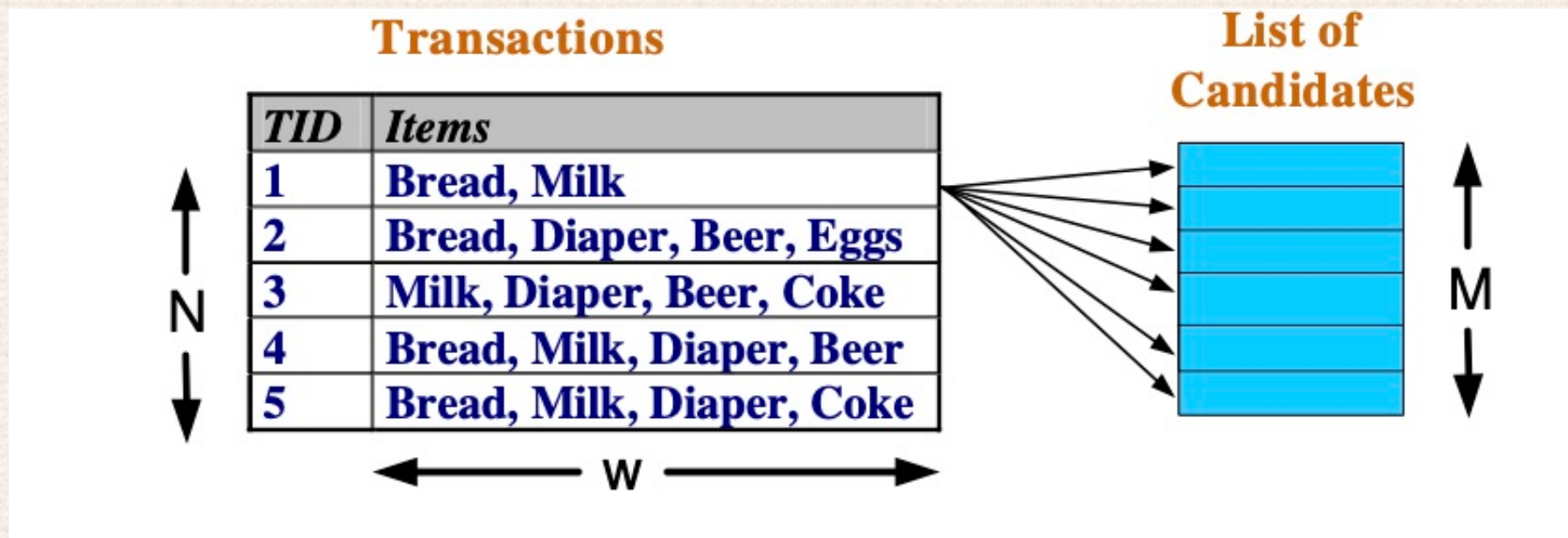
Generowanie zestawów częstych przedmiotów jest nadal kosztowne obliczeniowo.



Given d items, there are 2^d possible candidate itemsets

Generowanie zbioru częstych przedmiotów.

- Podejście brutalne:
 - Każdy zestaw elementów w kratce jest odpowiednim, częstym zestawem elementów
 - Policz poparcie każdego kandydata, skanując bazę danych



Strategie generowania zbioru częstych przedmiotów.

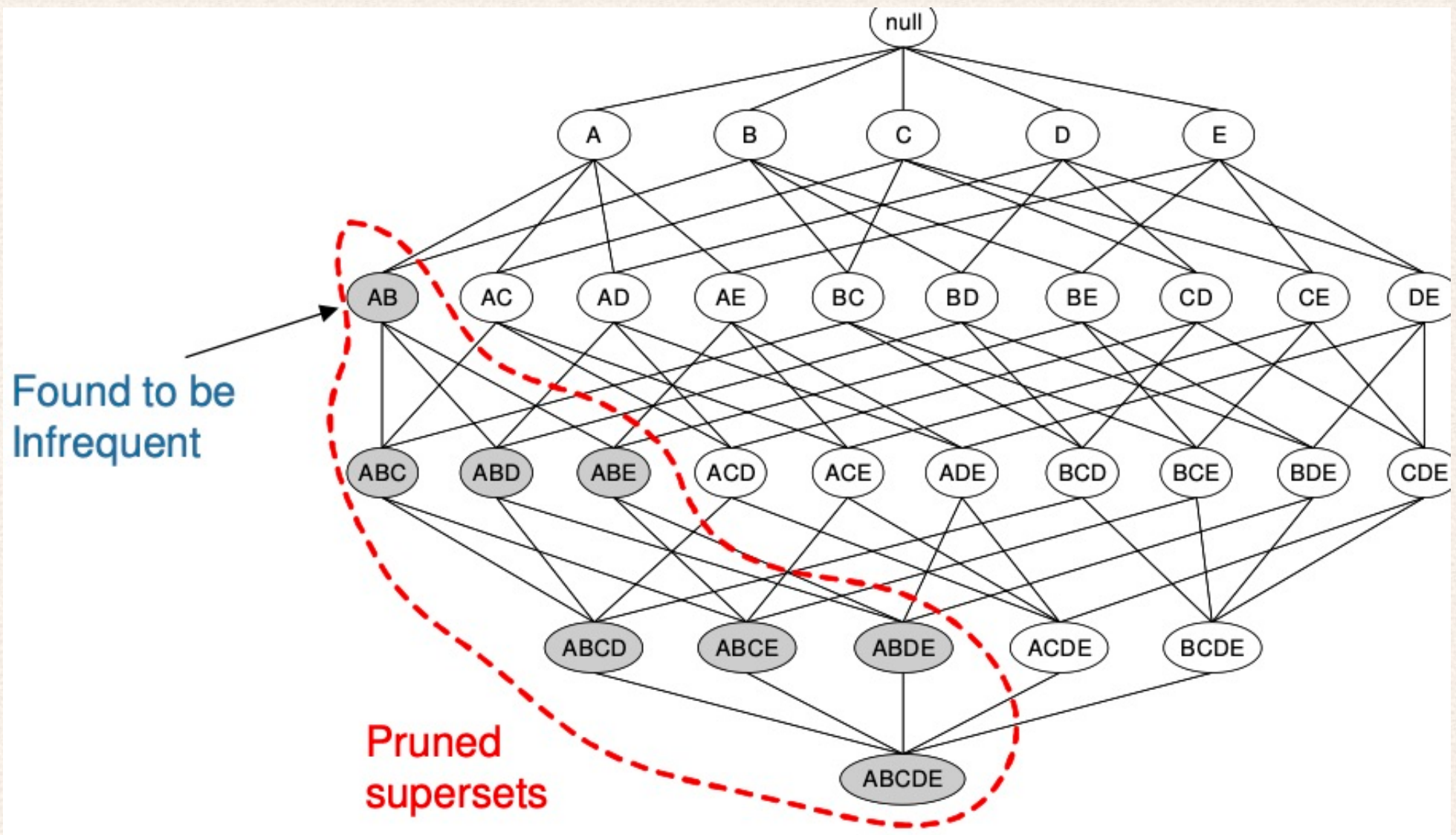
- Zmniejsz liczbę kandydatów (M)
 - Pełne wyszukiwanie: $M = 2^d$
 - Użyj technik przycinania, aby zmniejszyć M .
- Zmniejsz liczbę transakcji (N)
 - Zmniejsz rozmiar N wraz ze wzrostem wielkości zbioru częstych przedmiotów
 - Użyj DHP i algorytmów pionowej eksploracji
- Zmniejsz liczbę porównań (NM)
 - Korzystaj z wydajnych struktur danych do przechowywania kandydatów lub transakcji
 - Brak konieczności ponownego dopasowywania każdego kandydata do każdej transakcji

Redukowanie liczby kandydatów.

- Zasada Apriori:
 - Jeśli zestaw elementów występuje często, wszystkie jego podzbiory muszą być również często występujące
- Zasada Apriori obowiązuje ze względu na następującą właściwość miary nośnika:

$$\forall X, Y: (X \subseteq Y) \Rightarrow s(X) \geq s(Y)$$

- Nośnik zestawu przedmiotów nigdy nie przekracza wsparcia dla jego podzbiorów
- Jest to znane jako anty-monotoniczna właściwość nośnika



Item	Count
Bread	4
Coke	2
Milk	4
Beer	3
Diaper	4
Eggs	1

Items (1-itemsets)



Itemset	Count
{Bread,Milk}	3
{Bread,Beer}	2
{Bread,Diaper}	3
{Milk,Beer}	2
{Milk,Diaper}	3
{Beer,Diaper}	3

Pairs (2-itemsets)

(No need to generate candidates involving Coke or Eggs)

Minimum Support = 3

If every subset is considered,
 ${}^6C_1 + {}^6C_2 + {}^6C_3 = 41$
 With support-based pruning,
 $6 + 6 + 1 = 13$



Triplets (3-itemsets)

Item set	Count
{Bread,Milk,Diaper}	3



The Apriori Algorithm

- **Join Step:** C_k is generated by joining L_{k-1} with itself
- **Prune Step:** Any $(k-1)$ -itemset that is not frequent cannot be a subset of a frequent k -itemset
- Pseudo-code:

C_k : Candidate itemset of size k

L_k : frequent itemset of size k

$L_1 = \{\text{frequent items}\};$

for ($k = 1; L_k \neq \emptyset; k++$) **do begin**

C_{k+1} = candidates generated from L_k ;

for each transaction t in database **do**

 increment the count of all candidates in C_{k+1}
 that are contained in t

L_{k+1} = candidates in C_{k+1} with min_support

end

return $\bigcup_k L_k$

Dwa różne grupowania K-means

