Curs 8:

Reguli de asociere

Structura

- Motivaţie
  - Problema coşului de cumpărături
- Concepte de bază
  - Suport (support), încredere (confidence)
  - Seturi frecvente (frequent itemset)
- Algoritmul Apriori

Un exemplu

Analiza coşului de cumpărături (market basket analysis):

- Se consideră un set de înregistrări care conţin informaţii despre produsele cumpărate de clienţii unui supermarket
- Fiecare înregistrare corespunde unei tranzacţii şi conţine lista produselor achiziţionate

Exemplu:

T1: {milk, bread, meat, water}
T2: {bread, water}
T3: {bread, butter, meat, water}
T4: {water}

- Scop: identificarea produselor care sunt în mod frecvent achiziţionate împreună cu scopul de a extrage informaţii utile pentru decizii de marketing

Motivaţie

Problema de rezolvat: fiind dat un set de tranzacţii să se găsească regulile care descriu relaţii între apariţiile simultane ale unor produse în listele de tranzacţii

Exemplu: IF “bread AND minced meat” THEN “mustard”

Obs: regulile de asociere exprimă doar relaţii de co-ocurenţă nu şi relaţii de cauzalitate

La modul general o “tranzacţie” poate fi:

- Listă de produse sau servicii achiziţionate de către un client
- Lista de simptome asociate unui pacient
- Lista de cuvinte cheie sau nume de entităţi (named entities), adică nume de persoane, instituţii, locaţii, identificate într-o colecţie de documente
- Liste de actions urmate de un utilizator într-o aplicaţie de gestiune a unei reţele sociale

Concepte de bază

T1: {milk, bread, meat, water}
T2: {bread, water}
T3: {bread, butter, meat, water}
T4: {water}

- Entitate sau produs (item)
  - Element al unei tranzacţii (e.g: “water”)
  - Componentă a unei înregistrări: atribute=valoare (e.g. Vârsta =foarte tînâr)
- Set de entităţi (itemset) = colecţie sau mulţime
  - Exemplu: {bread, butter, meat, water}
- k-itemset = set de k entităţi
  - Exemple de 2-itemset: {bread, water}
- Set frecvent (frequent itemset) = un set care apare în multe tranzacţii
  - Frecvenţa set = nr de tranzacţii care conţin setul
  - Exemplu: 2-itemset-ul {bread,water} apare în 3 dintre cele 4 tranzacţii

Concepte de bază

T1: {milk, bread, meat, water}
T2: {bread, water}
T3: {bread, butter, meat, water}
T4: {water}

- Regulă de asociere = IF antecedent THEN consequent

(regulă ce conţine un itemset atât în partea de antecedent cât şi în cea de concluzie)

- Exemplu: IF {bread,meat} THEN {water}

- Cum poate fi interpretată această regulă?
  - Când se cumpără pâine şi carne există şansă mare să se cumpere şi apă
- Câtă încredere putem avea într-o astfel de regulă? Cât este ea de utilă ? Cum putem măsura calitatea unei reguli ?

Concepte de bază

T1: {milk, bread, meat, water}
T2: {bread, water}
T3: {bread, butter, meat, water}
T4: {water}

- Suport (support)
  - Pt un set: raportul dintre numărul tranzacţii ce conţin setul şi numărul total de tranzacţii
  - Pt o regulă: raportul dintre numărul tranzacţii ce conţin entităţile prezente în regulă (atât în membrul stâng cât şi în cel drept) şi numărul total de tranzacţii: supp(IF A THEN B)=supp({A,B})

Exemple:

- supp({milk,bread})=1/4=0.25
- supp({water})=4/4=1
- supp(IF {milk,bread} THEN {water})=supp({milk,bread,water})=1/4=0.25

Concepte de bază

T1: {milk, bread, meat, water}
T2: {bread, water}
T3: {bread, butter, meat, water}
T4: {water}

- Coeficientul de încredere a unei reguli – confidence (IF A THEN B)
  - Raportul dintre suportul setului {A,B} şi suportul lui {A}: supp({A,B})/supp(A)

Exemple:

- R1: IF {milk,bread} THEN {water}
  - supp({milk,bread,water})=1/4=0.25
  - supp({milk,bread})=1/4=0.25
  - conf(R1)=supp({milk,bread,water})/supp({milk,bread})=1
  - Interpretare: în toate cazurile în care se cumpără lapte şi pâine se cumpără şi apă.
- R2: IF {bread, water} THEN {meat}
  - conf(R2)=supp({bread,water,meat})/supp({bread,water})=2/3=0.66

Extragerea regulilor de asociere

T1: {milk, bread, meat, water}
T2: {bread, water}
T3: {bread, butter, meat, water}
T4: {water}

- Input: set de tranzacţii
- Output: set de reguli cu suport şi grad de încredere mai mare decât un prag specificat S={R1,R2,….}, adică

Fiecare regulă R: IF A THEN B satisface

supp(R)=supp({A,B})

=nr tranzacţii ce conţin A şi B/ nr total tranzacţii > prag suport (e.g. 0.2)

conf(R)=supp({A,B})/supp(A) > prag încredere(e.g. 0.7)

Obs: pragurile sunt specificate de către utilizator (de regulă pragul pt suport este mai mic decât cel pt încredere)

Extragerea regulilor de asociere

Abordări:

- Forţă brută: se generează toate regulile după care se aplică filtre (first generate then filter):
  - Generează toate regulile pornind de la setul E de entităţi
    - Pt fiecare submulţime A a lui E (considerată ca fiind membru stâng) se selectează fiecare submulţime B a lui (E-A) cu rol de membru drept şi se construieşte regula IF A THEN B
  - Selectează regulile care satisfac restricţiile privind suportul şi coeficientul de încredere
- OBS: o astfel de abordare este ineficientă; dacă N numărul total de entităţi din E atunci numărul de reguli generate este:

$$\sum_{k=1}^{N-1} C_N^k \sum_{i=1}^{N-k-1} C_{N-k}^i$$

Extragerea regulilor de asociere

T1: {milk, bread, meat, water}
T2: {bread, water}
T3: {bread, butter, meat, water}
T4: {water}

Forţa brută – exemplu:

- I={bread, butter, meat, milk, water}, N=5
- A={bread}; sunt 16 submulţimi ale lui E-A={butter, meat, milk, water} care pot fi folosite cu rol de membru drept
- R1: IF {bread} THEN {butter}
- R2: IF {bread} THEN {meat}
- R3: IF {bread} THEN {milk}
- R4: IF {bread} THEN {water}
- R5: IF {bread} THEN {butter,meat}
- R6: IF {bread} THEN {butter, milk)
- …
- R16: IF {bread} THEN {butter, meat, milk, water}
- … R500840 (mai mult de 500000 reguli în cazul unei liste cu 5 entităţi)

Extragerea regulilor de asociere

T1: {milk, bread, meat, water}
T2: {bread, water}
T3: {bread, butter, meat, water}
T4: {water}

Forţa brută – exemplu:

- I={bread, butter, meat, milk, water}, N=5
- A={bread}; sunt 16 submulţimi ale lui E-A={butter, meat, milk, water} care pot fi folosite cu rol de membru drept
- R1: IF {bread} THEN {butter} (supp(R1)=0.25, conf(R1)=0.33)
- R2: IF {bread} THEN {meat} (supp(R2)=0.5, conf(R2)=0.66)
- R3: IF {bread} THEN {milk} (supp(R3)=0.25, conf(R3)=0.33)
- R4: IF {bread} THEN {water} (supp(R4)=0.75, conf(R4)=1)
- R5: IF {bread} THEN {butter,meat} (supp(R5)=0.25, conf(R5)=1)
- R6: IF {bread} THEN {butter, milk) (supp(R6)=0.25, conf(R6)=1)
- …
- R16: IF {bread} THEN {butter, meat, milk, water}
  (supp(R6)=0, conf(R6)=0)

Extragerea regulilor de asociere

T1: {milk, bread, meat, water}
T2: {bread, water}
T3: {bread, butter, meat, water}
T4: {water}

Obs:

- Suportul regulii IF A THEN B este mai mare decât pragul doar dacă suportul lui {A,B} este mai mare decât pragul

- Idee: ar fi util să se identifice prima dată seturi cu un suport mai mare decât pragul şi apoi să se construiască reguli prin separarea setului între membrul stâng şi membrul drept
- De exemplu, nu are sens să se caute reguli pt care {A,B}={bread, butter, meat, milk, water}, întrucât suportul acestui set este 0

(in această abordare ar fi 2^N-2 reguli care implică toate entităţile – toate combinaţiile posibile în a le distribui astfel încât nici unul dintre cei doi membri ai regulii să nu fie nul)

Extragerea regulilor de asociere

T1: {milk, bread, meat, water}
T2: {bread, water}
T3: {bread, butter, meat, water}
T4: {water}

Abordări mai eficiente:

- Apriori :
  - Pas 1: Se determină toate seturile cu suportul mai mare decât pragul specificat (e.g. 0.2) – acestea sunt seturile frecvente (frequent itemsets)
  - Pas 2: Pt fiecare set se generează toate toate regulile posibiles (distribuind elementele setului între membrul stâng şi membrul drept) şi se selectează cele care au coeficientul de încredere mai mare decât pragul (e.g. 0.7)

- Obs: problema principală este generarea seturile frecvente fără o analiză exhaustivă a subseturilor (cum se face în abordarea bazată pe forţa brută)

Extragerea regulilor de asociere

T1: {milk, bread, meat, water}
T2: {bread, water}
T3: {bread, butter, meat, water}
T4: {water}

Întrebare: Cum s-ar putea identifica seturile frecvente fără a genera toate subseturile posibile?

Obs: orice subset al unui set frecvent trebuie să fie şi el frecvent (să aibă suportul mai mare decât pragul)

Exemplu: supp({bread, water, meat})=0.5 =>
supp({bread})=0.66>0.5, supp({water})=1>0.5, supp({meat})=0.5
supp({bread,water})=0.66>0.5, supp({bread,meat})=0.5
supp({water,meat})=0.5

Idee: se construiesc seturile frecvente incremental pornind de la seturi constituite dintr-un singur element

Extragerea regulilor de asociere

T1: {milk, bread, meat, water}
T2: {bread, water}
T3: {bread, butter, meat, water}
T4: {water}

Construirea seturilor frecvente
(prag pt suport: 0.3)
1-itemsets

{bread} supp({bread})=0.75
{butter} supp({butter})=0.25
{meat} supp({meat})=0.5
{milk} supp({milk})=0.25
{water} supp({water})=1

Extragerea regulilor de asociere

T1: {milk, bread, meat, water}
T2: {bread, water}
T3: {bread, butter, meat, water}
T4: {water}

Construirea seturilor frecvente
(prag pt suport: 0.3)
1-itemset-uri frecvente

{bread} supp({bread})=0.75
{butter} supp({butter})=0.25
{meat} supp({meat})=0.5
{milk} supp({milk})=0.25
{water} supp({water})=1

Extragerea regulilor de asociere

T1: {milk, bread, meat, water}
T2: {bread, water}
T3: {bread, butter, meat, water}
T4: {water}

Cosntruirea seturilor frecvente
(prag pt suport: 0.3)

1-itemset-uri

{bread} supp({bread})=0.75
{meat} supp({meat})=0.5
{water} supp({water})=1

2-itemset-uri

{bread,meat} supp({bread, meat})=0.5
{bread,water} supp({meat,water})=0.75
{meat,water} supp({water})=0.5

Extragerea regulilor de asociere

T1: {milk, bread, meat, water}
T2: {bread, water}
T3: {bread, butter, meat, water}
T4: {water}

Cosntruirea seturilor frecvente
(prag pt suport: 0.3)

1-itemset-uri

{bread} supp({bread})=0.75
{meat} supp({meat})=0.5
{water} supp({water})=1

2-itemset-uri frecvente

{bread,meat} supp({bread, meat})=0.5
{bread,water} supp({breadt,water})=0.75
{meat,water} supp({meat, water})=0.5

3-itemset-uri

{bread,meat,water} supp({bread, meat, water})=0.5

Extragerea regulilor de asociere

T1: {milk, bread, meat, water}
T2: {bread, water}
T3: {bread, butter, meat, water}
T4: {water}

Toate seturile frecvente cu cel puţin 2 entităţi
(prag pt suport: 0.3)

{bread,meat} supp({bread, meat})=0.5
{bread,water} supp({bread,water})=0.75
{meat,water} supp({meat,water})=0.5
{bread,meat,water} supp({bread, meat, water})=0.5

Reguli

R1: IF {bread} THEN {meat} conf(R1)=1
R2: IF {meat} THEN {bread} conf(R2)=0.66
R3: IF {bread} THEN {water} conf(R3)=1
R4: IF {water} THEN {bread} conf(R4)=0.75
R5: IF {meat} THEN {water} conf(R5)=1
R6: IF {water} THEN {meat} conf(R6)=0.5

Extragerea regulilor de asociere

T1: {milk, bread, meat, water}
T2: {bread, water}
T3: {bread, butter, meat, water}
T4: {water}

Toate seturile frecvente cu cel puţin 2 entităţi
(prag pt suport: 0.3)

{bread,meat} supp({bread, meat})=0.5
{bread,water} supp({bread,water})=0.75
{meat,water} supp({meat,water})=0.5
{bread,meat,water} supp({bread, meat, water})=0.5

Rules

R7: IF {bread} THEN {meat, water} conf(R7)=0.66
R8: IF {meat} THEN {bread, water} conf(R8)=1
R9: IF {water} THEN {bread, meat} conf(R9)=0.5
R10: IF {bread,meat} THEN {water} conf(R10)=1
R11: IF {bread,water} THEN {meat} conf(R11)=0.66
R12: IF {meat,water} THEN {bread} conf(R12)=1

Extragerea regulilor de asociere

T1: {milk, bread, meat, water}
T2: {bread, water}
T3: {bread, butter, meat, water}
T4: {water}

Toate regulile cu nivel de încredere ridicat
(prag pt nivelul de încredere: 0.75)

R1: IF {bread} THEN {meat} conf(R1)=1
R3: IF {bread} THEN {water} conf(R3)=1
R4: IF {water} THEN {bread} conf(R4)=0.75
R5: IF {meat} THEN {water} conf(R5)=1
R8: IF {meat} THEN {bread, water} conf(R8)=1
R10: IF {bread,meat} THEN {water} conf(R10)=1
R12: IF {meat,water} THEN {bread} conf(R12)=1

Obs: doar 12 din cele mai mult de 500000 de reguli posibile sunt generate; dintre acestea se selectează 7 reguli cu nivel ridicat de încredere

Extragerea regulilor de asociere

T1: {milk, bread, meat, water}
T2: {bread, water}
T3: {bread, butter, meat, water}
T4: {water}

Întrebare: sunt toate regulile cu nivel ridicat de încredere şi interesante? (o regulă interesantă furnizează informaţie ne-trivială, nouă sau neaşteptată)

Exemplu: regula IF {bread} THEN {water} are coeficientul de încredere 1; furnizează informaţie nouă?

Cum poate fi măsurat gradul de interes (noutate) a unei reguli?

Există diferite abordări. O variantă simplă este bazată pe Piatesky-Shapiro argument care afirmă că antecedentul şi concluzia unei reguli nu ar trebui să fie independente (în a sens statistic)

O regulă IF A THEN B este considerată interesantă dacă raportul (denumit “lift” sau “interest”)

supp({A,B})/(supp(A)*supp(B)) nu este apropiat de 1

Extragerea regulilor de asociere

T1: {milk, bread, meat, water}
T2: {bread, water}
T3: {bread, butter, meat, water}
T4: {water}

Eliminarea regulilor cu nivel mic al interesului
(cele pt care supp({A,B})=supp({A})*supp({B})

R1: IF {bread} THEN {meat} supp(R1)=0.75, supp({bread})*supp({meat})=0.37
R3: IF {bread} THEN {water} supp(R3)=0.75, supp({bread})*supp({water})=0.75
R4: IF {water} THEN {bread} supp(R4)=0.75, supp({bread})*supp({water})=0.75
R5: IF {meat} THEN {water} supp(R5)=0.5, supp({meat})*supp({water})=0.5

supp(R8)=supp(R10)=supp(R12)=0.5

R8: IF {meat} THEN {bread, water} supp({meat})*supp({bread, water})=0.37
R10: IF {bread,meat} THEN {water} supp({bread,meat})*supp({water})=0.5
R12: IF {meat,water} THEN {bread} supp({meat, water})*supp({bread})=0.37

Algoritmul Apriori

Structura generală:

Pas 1: Se generează lista de seturi frecvente incremental pornind de la seturi cu un element şi folosind proprietatea de anti-monotonie a măsurii suport

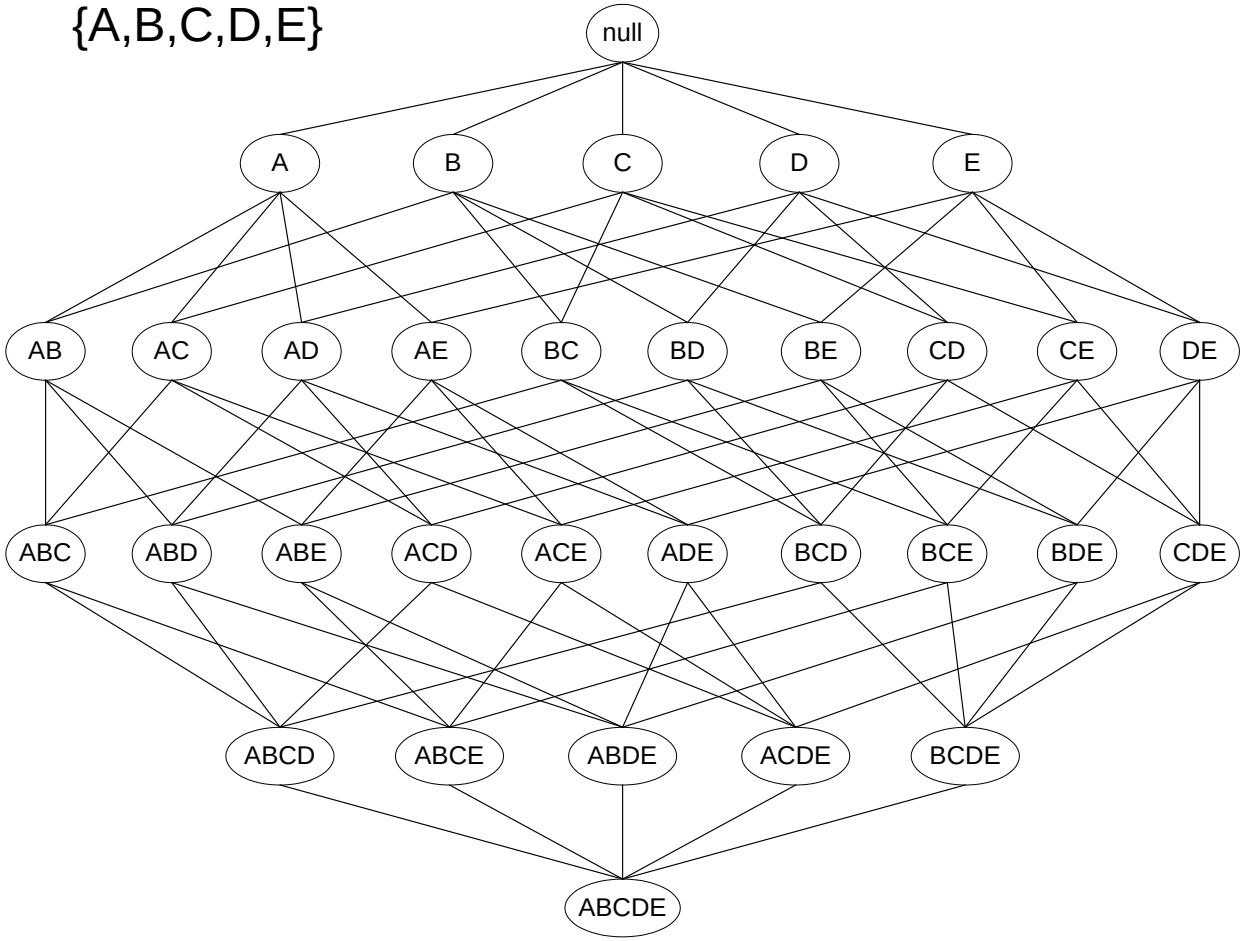
Pt orice submulţime B a setului A: supp(B)>=supp(A)

(principala implicaţie a acestei proprietăţi este că la construirea unui k-itemset se folosesc doar seturi mai mici care au o valoarea a suportului mai mare decât pragul)

Pas 2: Se construieşte lista de reguli analizând toate subseturile seturilor frecvente

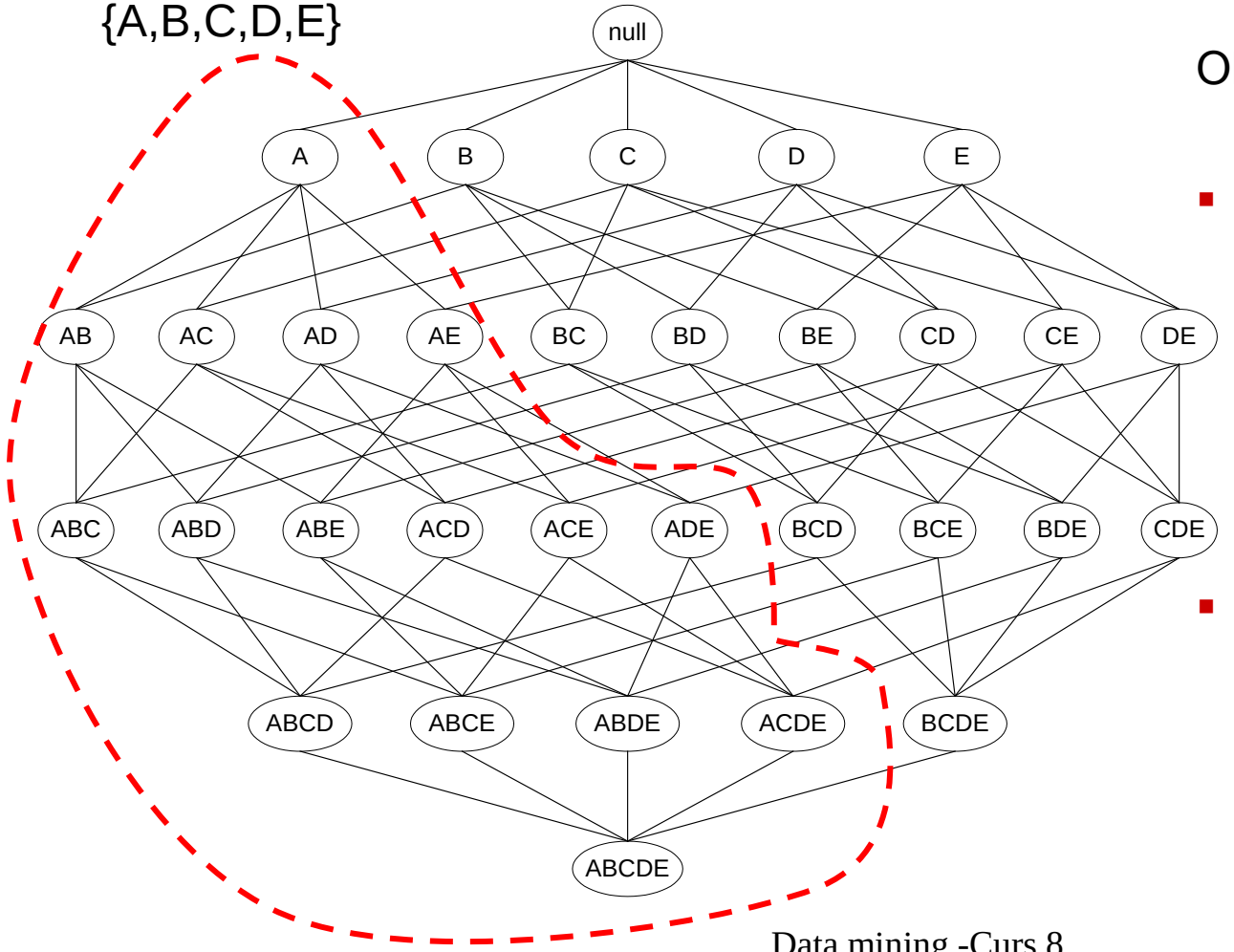
Algoritmul Apriori

Exemplu: construirea incrementala a submulţimilor unei mulţimi cu 5 elemente {A,B,C,D,E}



Algoritmul Apriori

Exemplu: construirea incrementala a submulţimilor unei mulţimi cu 5 elemente {A,B,C,D,E}



Observaţii

- Dacă {A} are un suport mic atunci spaţiul de căutare poate fi redus prin eliminarea tuturor seturilor care îl includ pe A
- Pt a construi un (k+1)-itemset este suficient să se reunească 2 k-itemset-uri frecvente care au (k-1) elemente comune

Algoritmul Apriori

Algoritm pentru generarea seturilor frecvente:

- k=1
- Se generează seturile frecvente cu 1 element

- **Repeat**
  - Generează (k+1) – itemset-uri candidat reunind k-itemset-uri care au k-1 elemente comune
  - Determină suportul item-set-urilor candidat (necesită parcurgerea setului de tranzacţii)
  - Elimină k-itemset-urile candidat care nu sunt frecvente

  Until nu se mai identifica seturi frecvente noi

Algoritmul Apriori

Algoritm pt generarea regulilor pornind de la lista L de itemset-uri frecvente:

- Initializează lista LR de reguli (lista vidă)
- FOR fiecare itemset IS din L
  - FOR fiecare submulţime A a lui IS se construieşte regula

    R(A,IS): IF A THEN IS-A
  - Calculează nivelul de încredere al regulii R(A,IS) şi dacă e mai mare decât pragul se adaugă R(A,IS) la LR

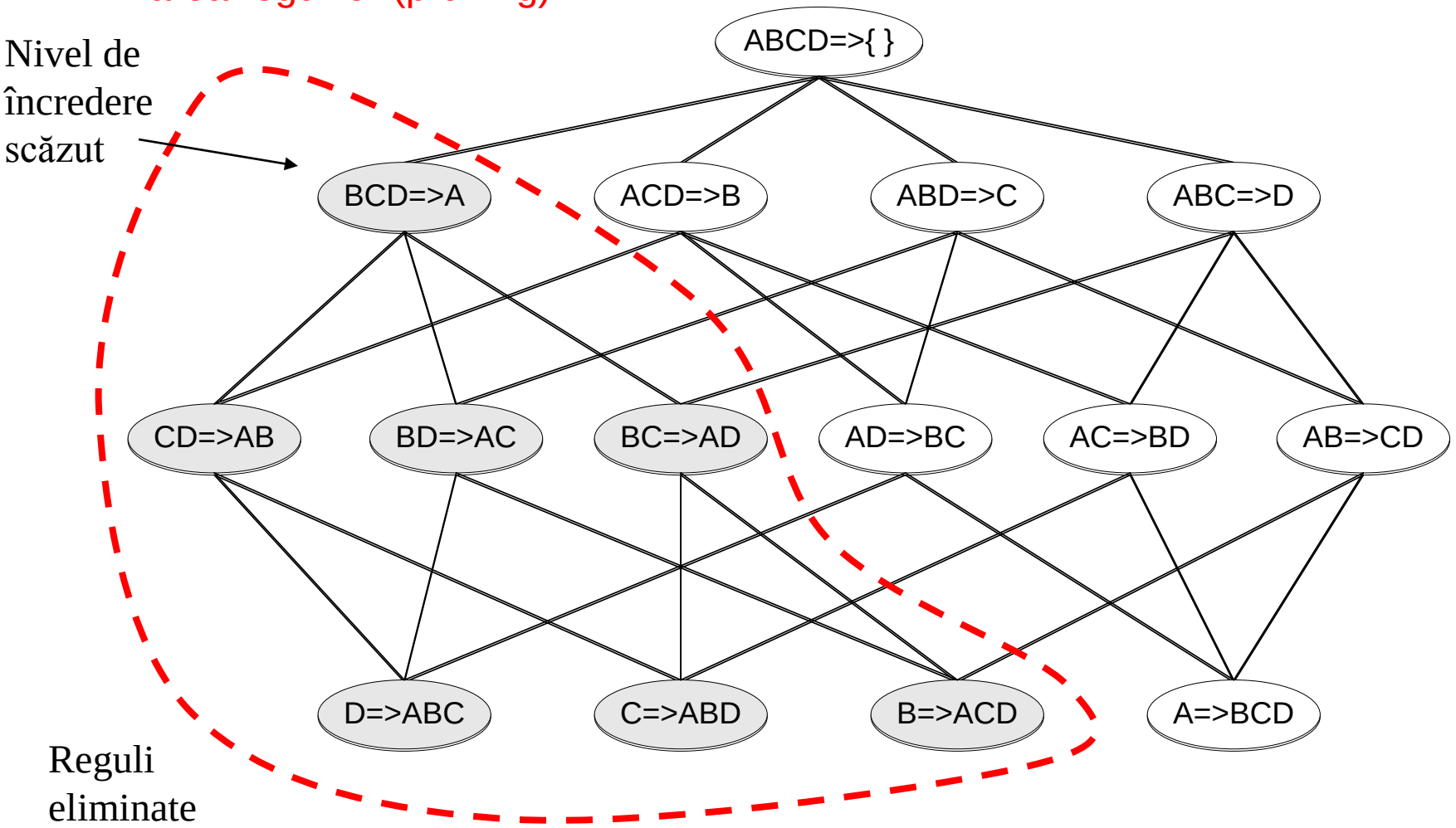
Obs:

- Pt fiecare k-itemset pot fi generate 2^k-2 rules (regulile cu membru stâng sau drept vid se ignoră)
- Pt a limita nr de reguli pt care se calculează nivelul de încredere se poate folosi proprietatea: nivelul de încredere este mai mare dacă cardinalitatea antecedentului este mai, i.e

$$conf(\{A,B,C\} \rightarrow D) \geq conf(\{A,B\} \rightarrow \{C,D\}) \geq conf(\{A\} \rightarrow \{B,C,D\})$$

Algoritmul Apriori

Eliminarea regulilor (pruning)



Algoritmul Apriori

Alte idei pentru a reduce volumul de calcule in procesul de generare a regulilor pornind de la seturi frecvente:

- Este mai eficient dacă se porneşte cu itemset-urile mari
- Noi reguli pot fi construite prin reunirea unor reguli existente

Exemplu:

- join(IF {C,D} THEN {A,B}, IF {B,D} THEN {A,C}) conduce la regula
  IF {D} THEN {A,B,C}

- Dacă regula IF {A,D} THEN {B,C} are nivelul de încredere mai mic decât pragul atunci regula reunită ar trebui eliminată (va avea nivelul de încredere mai mic)

Algoritmul Apriori

Influenţa pragurilor:

- Dacă pragul pt suport este prea mare atunci se pot pierde itemset-uri care includ entităţi rare (de exemplu produse scumpe, sau simptome rare)

- Dacă pragul pt suport este prea mic atunci numărul de itemset-uri generate este mare şi costul de calcul e mare