

Curs 9:

Reguli de asociere

Structura

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

Un exemplu

Analiza coșului de cumpărături (market basket analysis):

- Se consideră un set de înregistrări care conțin informații despre produsele cumpărate de clienții unui supermarket
- Fiecare înregistrare corespunde unei tranzacții și conține lista produselor achiziționate

Exemplu:

T1: {milk, bread, meat, water}

T2: {bread, water}

T3: {bread, butter, meat, water}

T4: {water}

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

Motivație

Problema de rezolvat: fiind dat un set de tranzacții să se găsească regulile care descriu relații între aparițiile simultane ale unor produse în listele de tranzacții

Exemplu: IF “bread AND meat” THEN “water”

Obs: regulile de asociere exprimă doar relații de **co-ocurență** nu și relații de cauzalitate

La modul general o “tranzacție” poate fi:

- Listă de produse sau servicii achiziționate de către un client → înțelegerea comportamentului utilizatorilor / identificare tipare de comportament
- Lista de simptome asociate unui pacient → identificare asocieri între diverse patologii
- Lista de cuvinte cheie sau nume de entități (named entities), adică nume de persoane, instituții, locații, identificate într-o colecție de documente
- Liste de acțiuni urmate de un utilizator într-o aplicație web → identificare acțiuni frauduloase

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 tranzacții (e.g: “water”)
 - Componentă a unei înregistrări: atribut=valoare (e.g. Vârsta =foarte tânăr)
- **Set de entități (itemset)** = colecție sau mulțime
 - Exemplu: {bread, butter, meat, water}
- **k-itemset** = set de k entități
 - Exemple de 2-itemset: {bread, water}
- **Set frecvent (frequent itemset)** = un set care apare în multe tranzacții
 - Frecvența set = nr de tranzacții care conțin setul
 - Exemplu: 2-itemset-ul {bread,water} apare în 3 dintre cele 4 tranzacții

Concepte de bază

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

- **Regulă de asociere** = IF antecedent THEN consecință
(regulă ce conține un itemset atât în partea de antecedent cât și în cea de concluzie)
- **Exemplu:** IF {bread, meat} THEN {water}
- Cum poate fi interpretată această regulă?
 - Când se cumpără pâine și carne există șansă mare să se cumpere și 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 tranzacții ce conțin setul și numărul total de tranzacții
 - Pt o regulă: raportul dintre numărul tranzacții ce conțin entitățile prezente în regulă (atât în membrul stâng cât și în cel drept) și numărul total de tranzacții: $\text{supp}(\text{IF } A \text{ THEN } B) = \text{supp}(\{A, B\})$

Exemple:

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

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} și suportul lui {A}:
 $\text{supp}(\{A,B\})/\text{supp}(A)$

Exemple:

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

Extragerea regulilor de asociere

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

- **Input:** set de tranzacții
- **Output:** set de reguli cu suport și grad de încredere mai mare decât un prag specificat $S=\{R1, R2, \dots\}$, adică

Fiecare regulă R: **IF A THEN B** satisface

$\text{supp}(R) = \text{supp}(\{A, B\})$

=nr tranzacții ce conțin A și B / nr total tranzacții > **prag suport** (e.g. 0.2)

$\text{conf}(R) = \text{supp}(\{A, B\}) / \text{supp}(A) > \text{prag încredere}$ (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:

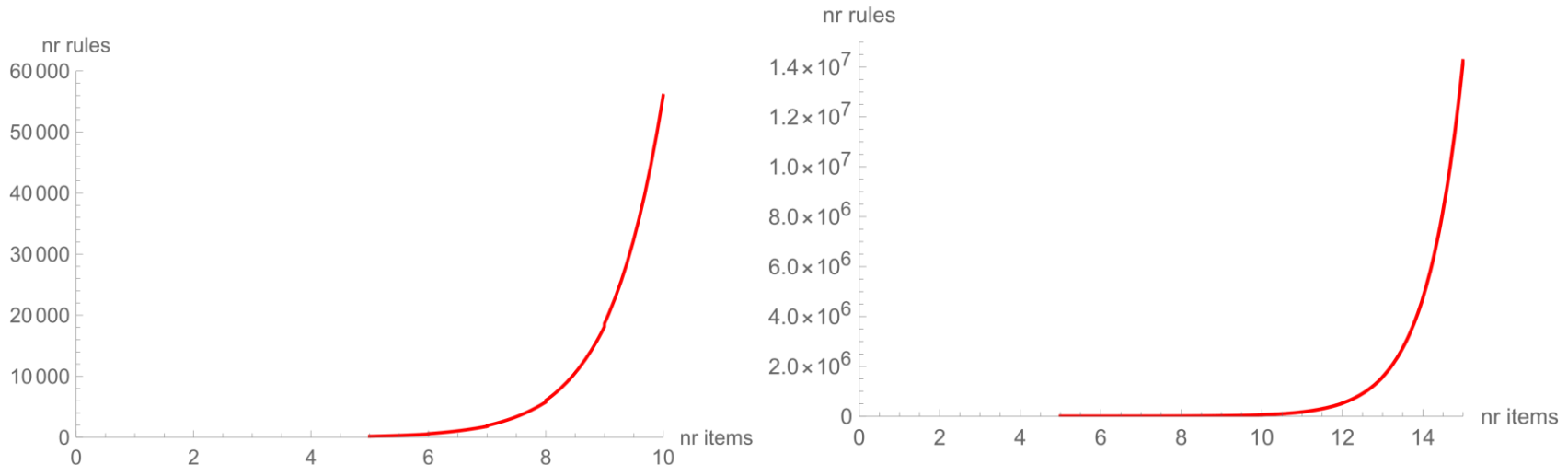
- **Forță brută:** se generează toate regulile după care se aplică filtre (first generate then filter):
 - Generează toate regulile pornind de la setul E de entități
 - Pt fiecare submulțime A a lui E (considerată ca fiind membru stâng) se selectează fiecare submulțime B a lui (E-A) cu rol de membru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

$$\sum_{k=1}^{N-1} C_N^k \sum_{i=1}^{N-k-1} C_{N-k}^i = 3^N - 3 \cdot 2^N + 3$$

Extragerea regulilor de asociere

Abordări:

- Brute force (prima dată se generează toate regulile după care se aplică filtrare):



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

Extragerea regulilor de asociere

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

Forța brută – exemplu:

- $I = \{\text{bread, butter, meat, milk, water}\}$, $N=5$
- $A = \{\text{bread}\}$; sunt 16 submulțimi ale lui $E - A = \{\text{butter, meat, milk, water}\}$ 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}
- ...
- R15: IF {bread} THEN {butter, meat, milk, water}
- ... R150 (150 reguli in cazul a 5 entități)

Extragerea regulilor de asociere

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

Forța brută – exemplu:

- $I = \{\text{bread, butter, meat, milk, water}\}$, $N=5$
- $A = \{\text{bread}\}$; sunt 16 submulțimi ale lui $E - A = \{\text{butter, meat, milk, water}\}$ 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)
- ...
- R15: 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 și abia apoi să se construiască reguli prin separarea setului între membrul stâng și 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

(pentru un set cu n elemente ar fi $2^n - 2$ reguli care implică toate entitățile – toate combinațiile 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:

- **Ideea algoritmului 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 regulile posibile (distribuind elementele setului între membrul stâng și membrul drept) și se selectează cele care au coeficientul de încredere mai mare decât pragul (e.g. 0.7)
- **Obs:** problema principală este generarea seturilor frecvente fără o analiză exhaustivă a subseturilor (cum se face în abordarea bazată pe forța 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 și el frecvent (să aibă suportul mai mare decât pragul)

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

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

Extragerea regulilor de asociere

Construirea seturilor frecvente
(prag pt suport: 0.3)

1-subseturi

{bread} $\text{supp}(\{\text{bread}\})=0.75$

{butter} $\text{supp}(\{\text{butter}\})=0.25$

{meat} $\text{supp}(\{\text{meat}\})=0.5$

{milk} $\text{supp}(\{\text{milk}\})=0.25$

{water} $\text{supp}(\{\text{water}\})=1$

T1: {milk, bread, meat, water}

T2: {bread, water}

T3: {bread, butter, meat, water}

T4: {water}

Extragerea regulilor de asociere

Construirea seturilor frecvente
(prag pt suport: 0.3)

1-subset-uri frecvente

{bread} $\text{supp}(\{\text{bread}\})=0.75$

{butter} $\text{supp}(\{\text{butter}\})=0.25$

{meat} $\text{supp}(\{\text{meat}\})=0.5$

{milk} $\text{supp}(\{\text{milk}\})=0.25$

{water} $\text{supp}(\{\text{water}\})=1$

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

Extragerea regulilor de asociere

Construirea seturilor frecvente
(prag pt suport: 0.3)

1-subset-uri

{bread} $\text{supp}(\{\text{bread}\})=0.75$

{meat} $\text{supp}(\{\text{meat}\})=0.5$

{water} $\text{supp}(\{\text{water}\})=1$

2-subset-uri

{bread,meat} $\text{supp}(\{\text{bread, meat}\})=0.5$

{bread,water} $\text{supp}(\{\text{bread,water}\})=0.75$

{meat,water} $\text{supp}(\{\text{meat, water}\})=0.5$

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

Extragerea regulilor de asociere

Construirea seturilor frecvente
(prag pt suport: 0.3)

1-subset-uri

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

3-subset-uri

{bread,meat,water} $\text{supp}(\{\text{bread, meat, water}\})=0.5$

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

2-subset-uri frecvente

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

Extragerea regulilor de asociere

Toate seturile frecvente cu cel puțin 2 entități

(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

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

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

Toate seturile frecvente cu cel puțin 2 entități

(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

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

Reguli

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

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

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

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 150 de reguli posibile sunt generate; dintre acestea se selectează 7 reguli cu nivel ridicat de încredere

Extragerea regulilor de asociere

Întrebare: sunt toate regulile cu nivel ridicat de încredere și **reguli de interes**? (o regulă de interes furnizează informație ne-trivială, nouă sau neașteptată)

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

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

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

Există diferite abordări. O variantă simplă este bazată pe ideea că antecedentul și concluzia unei reguli nu ar trebui să fie independente (în sens statistic)

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

$\text{supp}(\{A,B\})/(\text{supp}(A)*\text{supp}(B))$ NU este apropiat de 1

Extragerea regulilor de asociere

Eliminarea regulilor cu nivel mic al gradului de interes

(cele pt care $\text{supp}(\{A,B\}) = \text{supp}(\{A\}) * \text{supp}(\{B\})$)

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

R1: IF {bread} THEN {meat} $\text{supp}(R1)=0.75, \text{supp}(\{bread\}) * \text{supp}(\{meat\})=0.37$

R3: IF {bread} THEN {water} $\text{supp}(R3)=0.75, \text{supp}(\{bread\}) * \text{supp}(\{water\})=0.75$

R4: IF {water} THEN {bread} $\text{supp}(R4)=0.75, \text{supp}(\{bread\}) * \text{supp}(\{water\})=0.75$

R5: IF {meat} THEN {water} $\text{supp}(R5)=0.5, \text{supp}(\{meat\}) * \text{supp}(\{water\})=0.5$

$\text{supp}(R8)=\text{supp}(R10)=\text{supp}(R12)=0.5$

R8: IF {meat} THEN {bread, water} $\text{supp}(\{meat\}) * \text{supp}(\{bread, water\})=0.37$

R10: IF {bread,meat} THEN {water} $\text{supp}(\{bread,meat\}) * \text{supp}(\{water\})=0.5$

R12: IF {meat,water} THEN {bread} $\text{supp}(\{meat, water\}) * \text{supp}(\{bread\})=0.37$

Algoritmul Apriori

Structura generală:

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

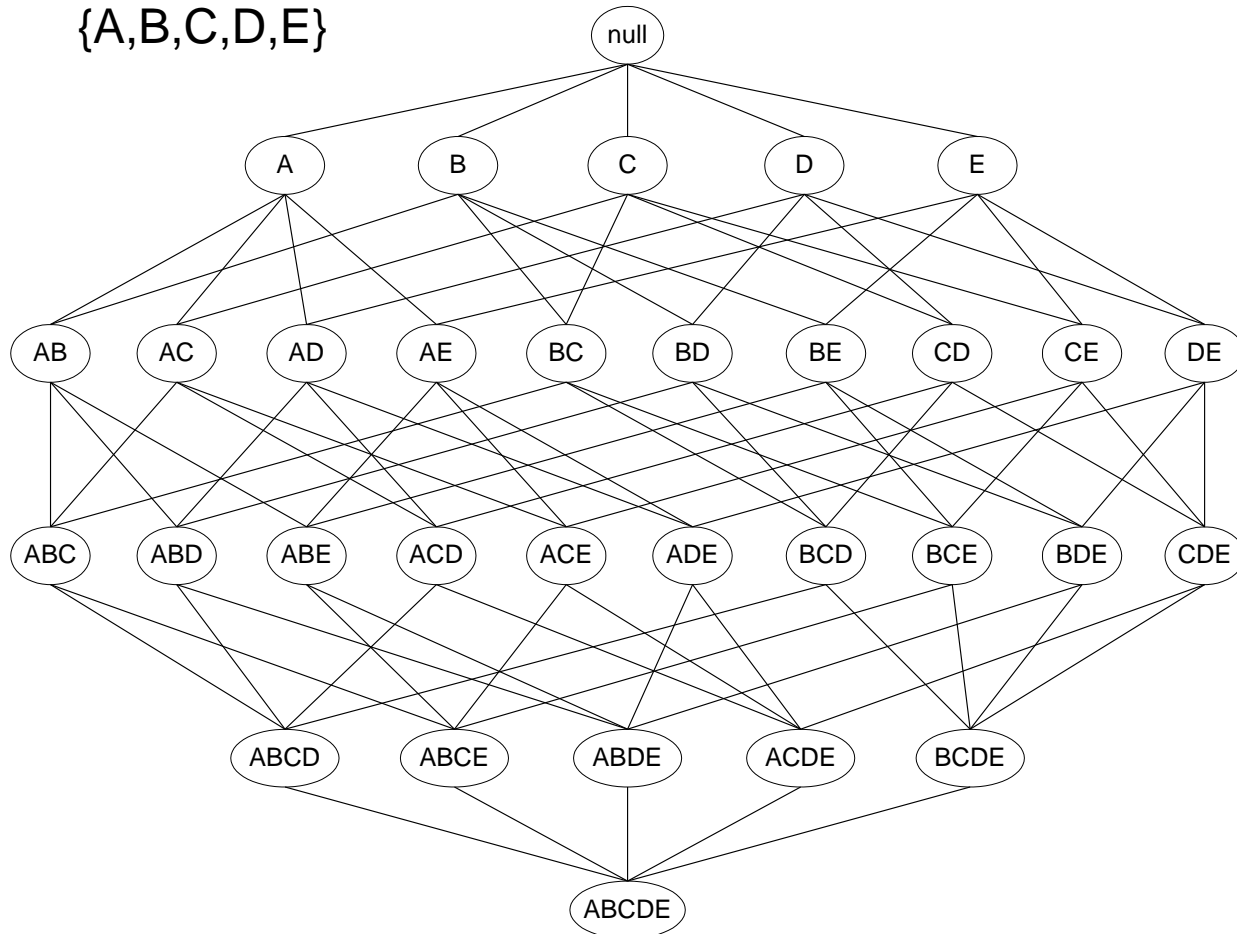
Pt orice submulțime B a setului A: $\text{supp}(B) \geq \text{supp}(A)$

(principala implicație a acestei proprietăți este că la construirea unui k-itemset se folosesc doar seturi cu mai puține elemente care au o valoare a suportului mai mare decât pragul)

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

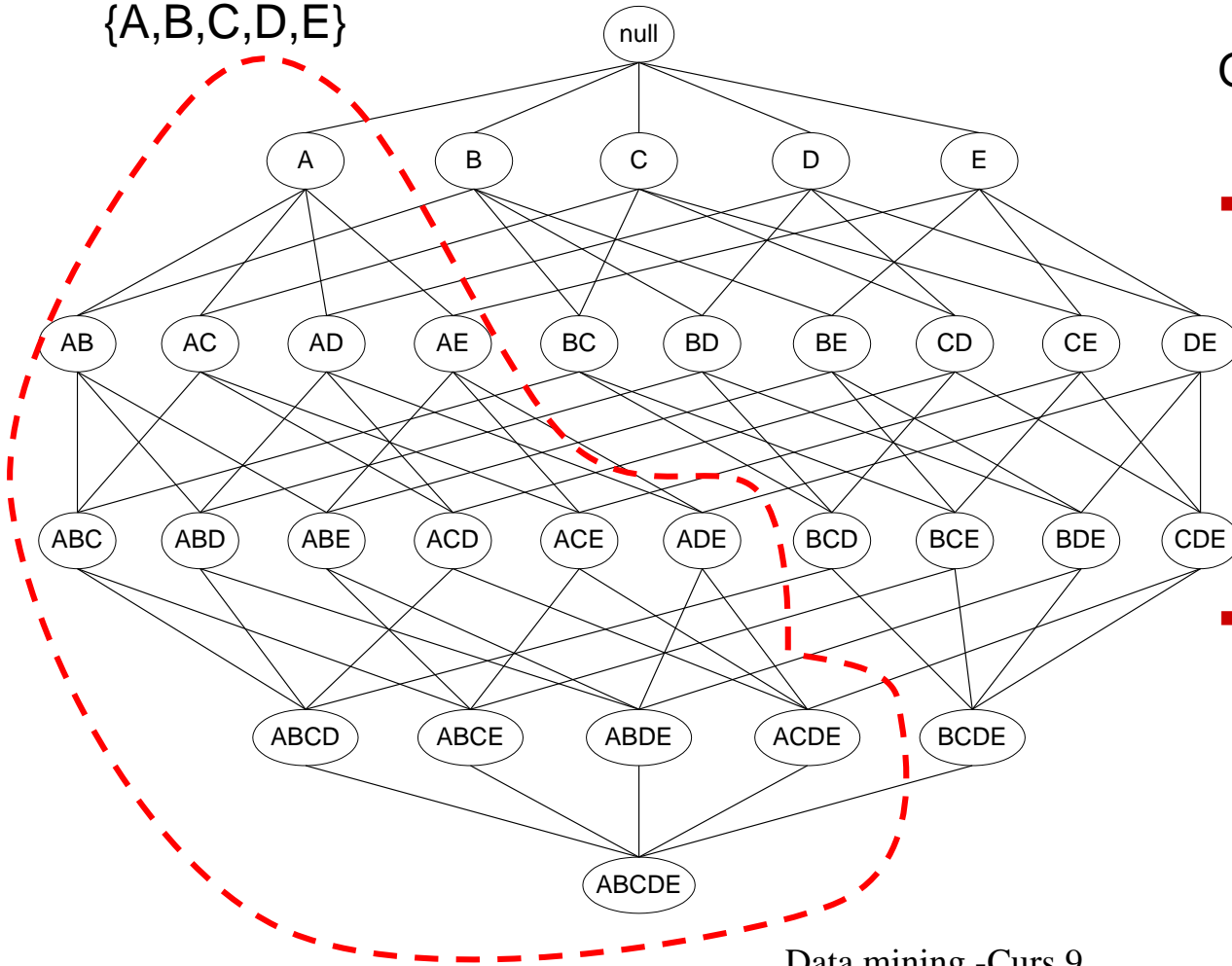
Algoritmul Apriori

Exemplu: construirea incrementală a submulțimilor unei mulțimi cu 5 elemente {A,B,C,D,E}



Algoritmul Apriori

Exemplu: construirea incrementală a submulțimilor unei mulțimi cu 5 elemente {A,B,C,D,E}



Observații

- Dacă $\{A\}$ are un suport mic atunci spațiul 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 tranzacții)
 - 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 submulțime A a lui IS se construiește regula
 $R(A, IS): \text{IF } A \text{ THEN } IS-A$
 - Calculează nivelul de încredere al regulii $R(A, IS)$ și dacă e mai mare decât pragul se adaugă $R(A, IS)$ la LR

Obs:

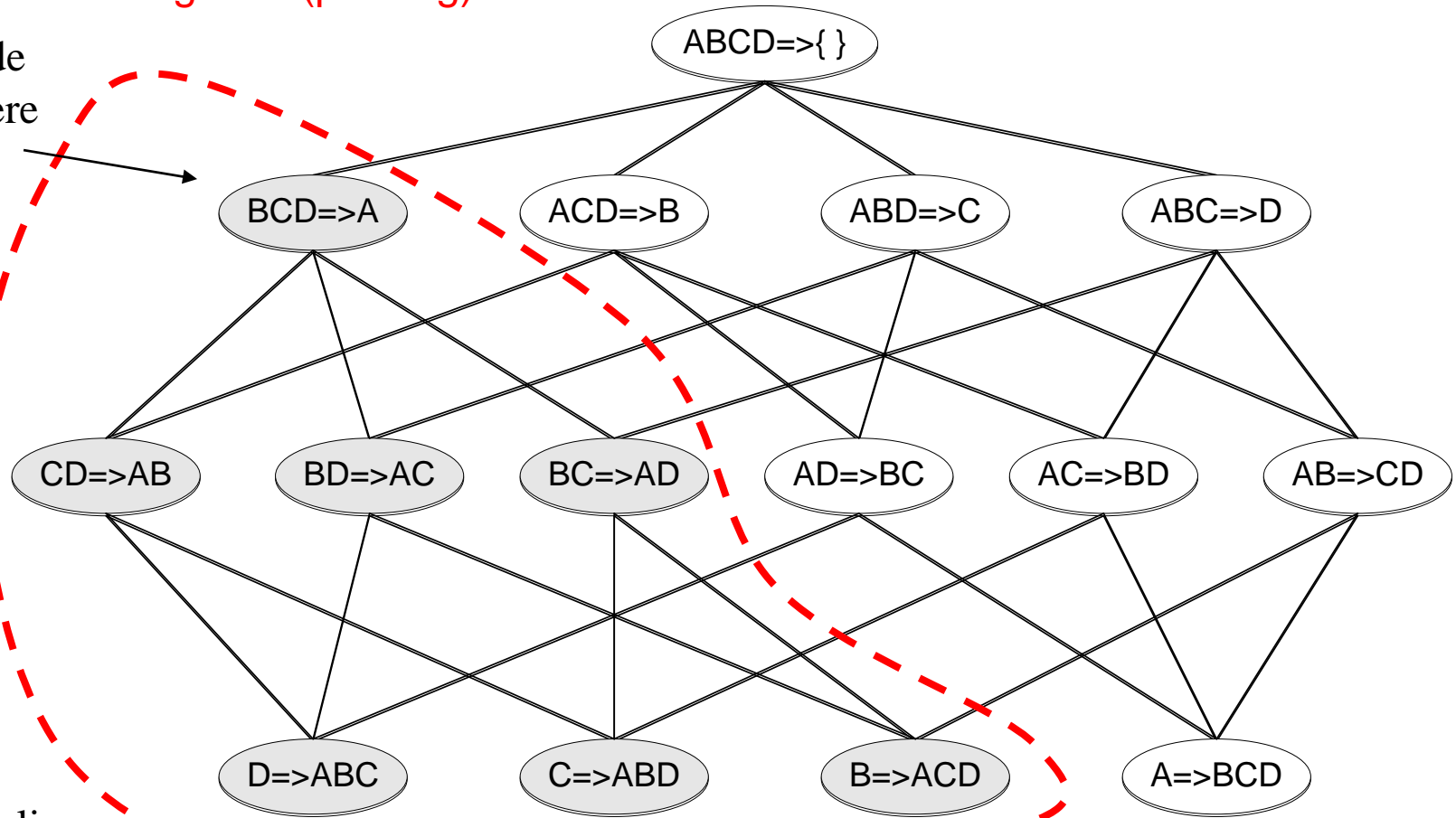
- Pt fiecare k-itemset pot fi generate $2^k - 2$ reguli (regulile cu membru stâng sau drept vid se ignoră)
- Pt a limita nr de reguli pt care se calculează nivelul de încredere se poate folosi proprietatea: nivelul de încredere este mai mare dacă cardinalitatea antecedentului este mai mare, adică

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

Algoritmul Apriori

Eliminarea regulilor (pruning)

Nivel de
încredere
scăzut



Reguli
eliminate

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 pornește cu itemset-urile mari
- Noi reguli pot fi construite prin reunirea unor reguli existente

Exemplu:

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

Algoritmul Apriori

Influența pragurilor:

- Dacă pragul pt suport este prea mare atunci se pot pierde itemset-uri care includ entități rare (de exemplu produse scumpe, sau simptome rare)
- Dacă pragul pt suport este prea mic atunci numărul de itemset-uri generate este mare și costul de calcul e mare

Complexitatea computațională a algoritmului Apriori este influențată de:

- Pragul pentru suport
- Numărul de entități
- Numărul de tranzacții
- Numărul mediu de entități

Sumar

Etapele procesului de extragere a regulilor de asociere:

- **Etapa 1:** determinare subseturi frecvente (parametru control: prag pt suport)
 - Algoritm Apriori (exemplu implementare: <https://github.com/tommyod/Efficient-Apriori>)
 - Algoritm FP (Frequent Patterns) Growth – utilizează o reprezentare arborescentă a tranzacțiilor [C. Borgelt, An Implementation of the FP-growth Algorithm, 2010, exemplu implementare: https://github.com/chonyy/fpgrowth_py]
- **Etapa 2:** construirea regulilor (parametru control: prag pt nivelul de încredere)
 - Se pornește de la subseturile frecvente și se construiesc regulile pornind de la cele care conțin cele mai multe entități în antecedent

Obs: există variante ale algoritmilor de extragere a regulilor specifice:

- Atributelor continue
- Identificării tiparelor secvențiale
- Identificării de asocieri între subgrafuri

Cursul următor

Modele de regresie neliniară

- Modele liniare generalizate
- Arbori de regresie
- Rețele bazate pe funcții radiale (RBF networks)