CS57300
PURDUE UNIVERSITY

NOVEMBER 22, 2021

DATA MINING

DATA MINING COMPONENTS

- Task specification
- **Knowledge representation**
- Learning technique
- Evaluation

RULE

- A rule is an expression of the form $\theta \rightarrow \varphi$
- A statement about the co-occurrence of events/patterns
- **Support** (aka frequency)
  - $s(\theta \rightarrow \varphi) = fr(\theta \wedge \varphi) / N$
  - Proportion of N items with antecedent θ and consequent φ
- **Confidence** (aka accuracy)
  - $c(\theta \rightarrow \varphi) = p(\varphi \mid \theta) = fr(\theta \wedge \varphi) / fr(\theta)$
  - Proportion of items which have antecedent θ that also have consequent φ

ASSOCIATION RULES

- Find all rules of the form $\theta \rightarrow \varphi$ that satisfy the following constraints:
  - Support of the rule is greater than threshold *s*
  - Confidence of the rule is greater than threshold *c*

DATA MINING COMPONENTS

- Task specification
- Knowledge representation
- **Learning technique**
- Evaluation

MODEL SPACE AND SEARCH

- Model space: All possible rules
- Suppose there are N binary variables
- Even if we only consider rules where θ and φ are conjunctions of $X_k=1$
  - We still have $\binom{N}{2}\binom{2}{1} + \binom{N}{3}\left(\binom{3}{1} + \binom{3}{2}\right) + \ldots + \binom{N}{N} \times \left(\binom{N}{1} + \binom{N}{2} + \ldots + \binom{N}{N-1}\right)$ rules
- Searching for all patterns is computationally intractable

SOLUTION: THE APRIORI ALGORITHM

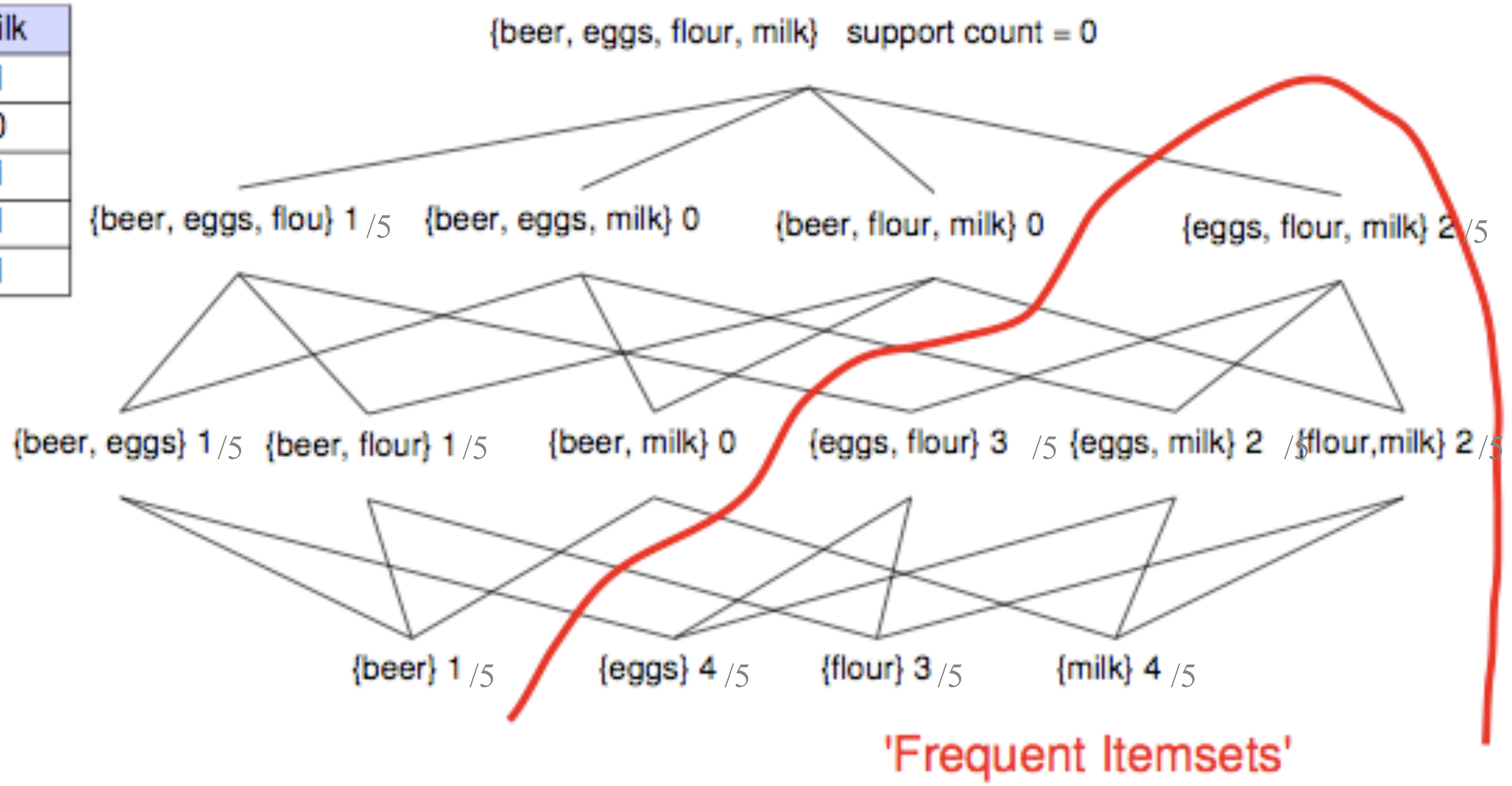
- Key idea: Decompose the search process into two steps
- First search for “frequent itemset”: combinations of predicate whose support is above the threshold
- Then search among frequent items to prune rules whose confidence is below threshold

FINDING FREQUENT ITEMSETS

- Find sets of items with minimum support
- Support is ***monotonic***
  - A subset of a frequent itemset must also be frequent
  - Eg. If {A,B} is a frequent itemset then both {A} and {B} are frequent itemsets as well
  - That is, if {A} is not a frequent itemset, then {A, B} can't be a frequent itemset either
- Approach
  - Iteratively find frequent itemsets with cardinality from 1 to k (k-itemset)
  - Prune any sets of size k that are not frequent

EXAMPLE

Transaction ID	beer	eggs	flour	milk
1	0	1	1	1
2	1	1	1	0
3	0	1	0	1
4	0	1	1	1
5	0	0	0	1



support threshold = 0.2

ALGORITHM TO FIND FREQUENT ITEMSETS

FrequentItemsetGeneration (D, minsup)

% C_k: candidate itemsets of size k; L_k: frequent itemsets of size k

L_1 = {frequent single items}

for (k=1; L_k!=∅; k++)

C_{k+1} = CandidateItemsetGeneration (L_k, minsup)

for each transaction t in D

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

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

Return $\bigcup_k L_k$

GENERATING CANDIDATES

CandidateItemsetGeneration (L_k, minsup)

% step 1: self-joining L_k

C_{k+1} = {}

For p in L_k, q in L_k ,p!=q:

Add p∪q in C_{k+1} if | p∪q |=k+1

% step 2: pruning

For c in C_{k+1}

For all k-item subsets s of c

If s not in L_k then delete c from C_{k+1}

EXAMPLE

support threshold = 0.3

Database D

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D

C_1

itemset	sup.
{1}	2/4
{2}	3/4
{3}	3/4
{4}	1/4
{5}	3/4

L_1

itemset	sup.
{1}	2/4
{2}	3/4
{3}	3/4
{5}	3/4

C_2

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

Scan D

C_2

itemset	sup
{1 2}	1/4
{1 3}	2/4
{1 5}	1/4
{2 3}	2/4
{2 5}	3/4
{3 5}	2/4

L_2

itemset	sup
{1 3}	2/4
{2 3}	2/4
{2 5}	3/4
{3 5}	2/4

C_3

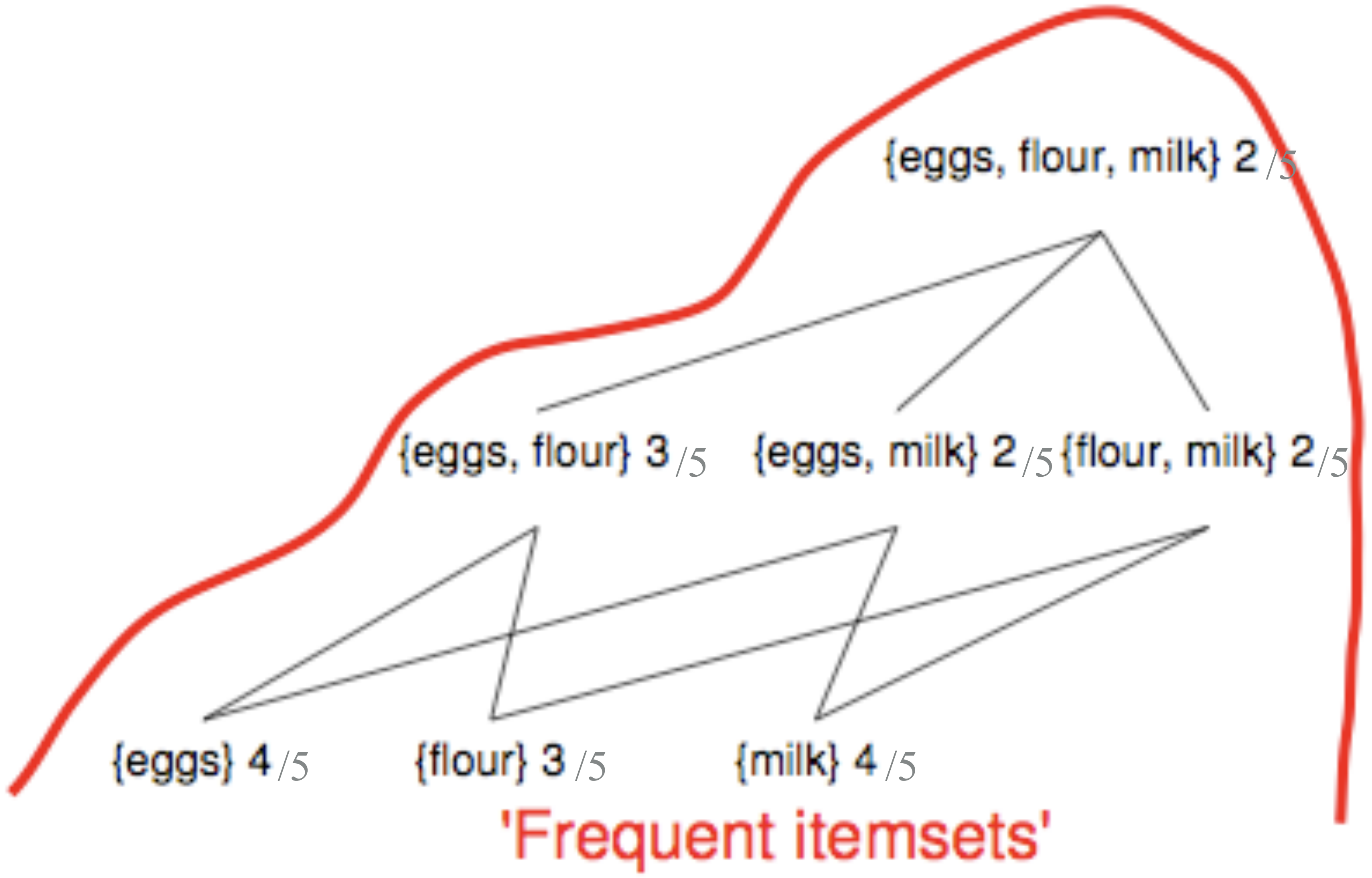
itemset
{2 3 5}

Scan D

L_3

itemset	sup
{2 3 5}	2/4

EXAMPLE



			Confidence
{eggs}	$\longrightarrow$	{flour}	$3/4 = 0.75$
{flour}	$\longrightarrow$	{eggs}	$3/3 = 1$
{eggs}	$\longrightarrow$	{milk}	$2/4 = 0.5$
{milk}	$\longrightarrow$	{eggs}	$2/4 = 0.5$
{flour}	$\longrightarrow$	{milk}	$2/3 = 0.67$
{milk}	$\longrightarrow$	{flour}	$2/4 = 0.5$
{eggs, flour}	$\longrightarrow$	{milk}	$2/3 = 0.67$
{eggs, milk}	$\longrightarrow$	{flour}	$2/2 = 1$
{flour, milk}	$\longrightarrow$	{eggs}	$2/2 = 1$
{eggs}	$\longrightarrow$	{flour, milk}	$2/4 = 0.5$
{flour}	$\longrightarrow$	{eggs, milk}	$2/3 = 0.67$
{milk}	$\longrightarrow$	{eggs, flour}	$2/4 = 0.5$

RULE GENERATION

- Given a frequent itemset L, find all non-empty subsets f ⊂ L such that f → (L - f) satisfies the minimum confidence requirement

If {A,B,C,D} is a frequent itemset, candidate rules:

ABC →D,	ABD →C,	ACD →B,	BCD →A,
A →BCD,	B →ACD,	C →ABD,	D →ABC
AB →CD,	AC → BD,	AD → BC,	BC →AD,
BD →AC,	CD →AB,		

- If |L|=k then there are 2^k-2 candidate association rules (ignoring L → ∅ and ∅ → L)

EFFICIENT RULE GENERATION

- Key insight: the confidence of rules generated from the same itemset is monotonic with respect to the number of items in the consequent
  - Recall that:
    c(θ→φ) = *p(φ | θ)*
  - Consider frequent itemset L={A,B,C,D}:

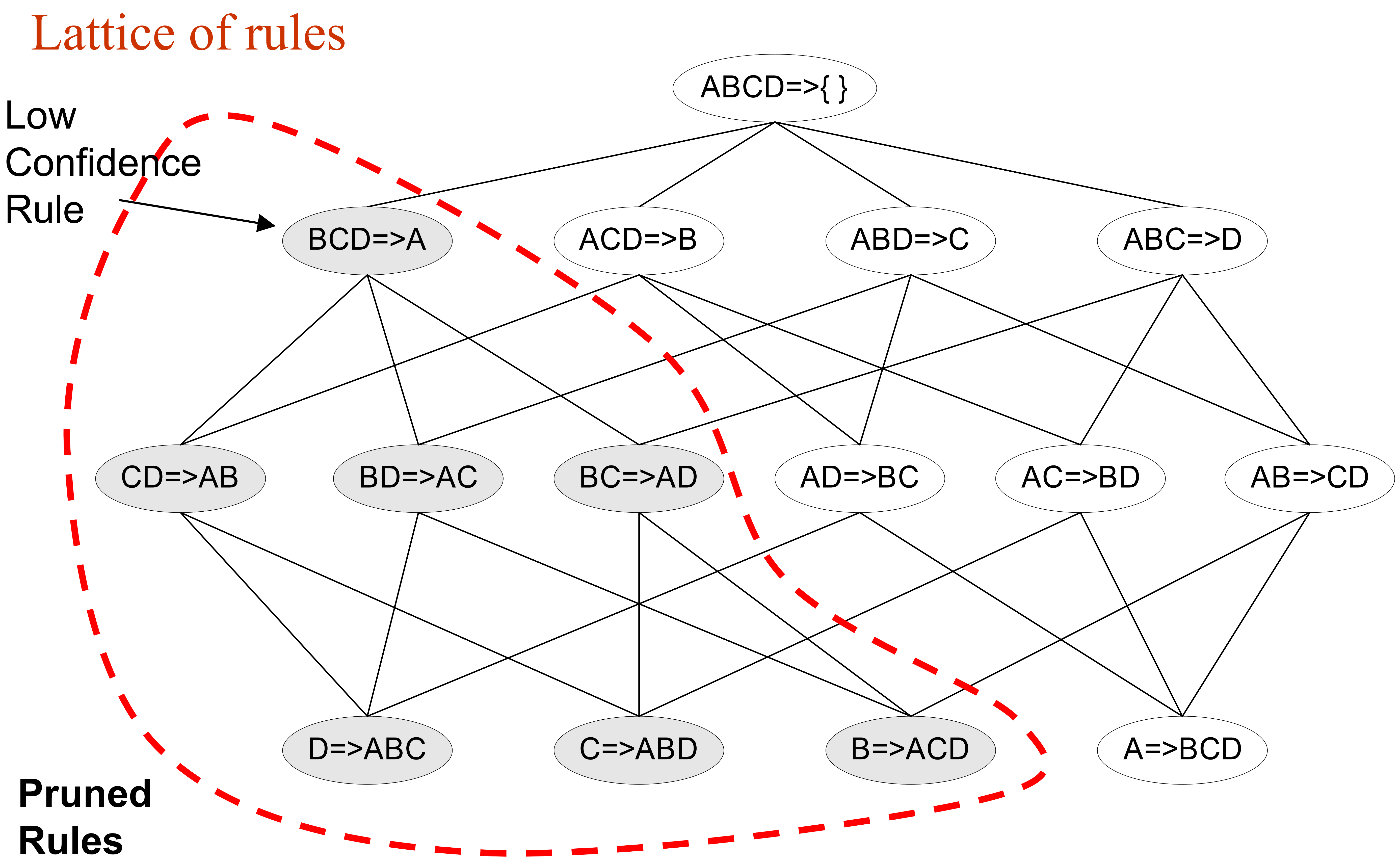
$$c(ABC \to D) = P(D|ABC) = \frac{fr(ABCD)}{fr(ABC)}$$

$$c(AB \to CD) = P(CD|AB) = \frac{fr(ABCD)}{fr(AB)}$$

$$\text{We know: } fr(ABC) \leq fr(AB) \text{ and } \frac{1}{fr(ABC)} \geq \frac{1}{fr(AB)}$$

$$\text{thus: } c(ABC \to D) \geq c(AB \to CD) \geq c(A \to BCD)$$

PRUNING RULES



ALGORITHM TO FIND RULES WITH HIGH CONFIDENCE

Let R_m=confident rules with m variable consequents

Let H_m=candidate rules with m variable consequents

RuleGeneration (**L**, minconf)

for (k=1; L_k!=∅; k++)

H₁=candidate rules with single variable consequent from L_k

for (m=1; H_m!=∅; m++)

If k > m + 1:

H_{m+1} = generate candidate rules from R_m

R_{m+1} = select candidates in H_{m+1} with minconf

Return $\bigcup_m R_m$

APRIORI ALGORITHM

- Input: data (D), minsup, minconf
- Output: All rules (**R**) with support ≥ minsup and confidence ≥ minconf

Apriori Algorithm (D, minsup, minconf)

% Find all itemsets with support ≥ minsup
L = FrequentItemsetGeneration (D, minsup)

% Find all rules with confidence ≥ minconf
R = RuleGeneration (**L**, minconf)

Return R

EVALUATION

EVALUATION

- Association rules algorithms usually return many, many rules
  - Many are uninteresting or redundant (e.g., ABC→D and AB→D may have same support and confidence)
- How to quantify interestingness?
  - Objective: statistical measures
  - Subjective: *unexpected* and/or *actionable* patterns (requires domain knowledge)

OBJECTIVE MEASURES

- Given a rule X→Y, can compute statistics based on contingency tables

Contingency table for $X \rightarrow Y$

	Y	$\overline{Y}$	
X	f_{11}	f_{10}	f_{1+}
$\overline{X}$	f_{01}	f_{00}	f_{o+}
	f_{+1}	f_{+0}	$\|T\|$

f_{11}: support of X and Y
f_{10}: support of X and $\overline{Y}$
f_{01}: support of $\overline{X}$ and Y
f_{00}: support of $\overline{X}$ and $\overline{Y}$

Used to define various measures

- support, confidence, lift, Gini, J-measure, etc.

DRAWBACK OF SUPPORT

- Support suffers from the **rare item problem** (Liu et al.,1999)
  - Infrequent items not meeting minimum support are ignored which is problematic if rare items are important
  - E.g. rarely sold products which account for a large part of revenue or profit
- Support falls rapidly with itemset size. A threshold on support favors short itemsets

DRAWBACK OF CONFIDENCE

	Coffee	$\overline{\text{Coffee}}$	
Tea	15	5	20
$\overline{\text{Tea}}$	75	5	80
	90	10	100

Association Rule: Tea $\rightarrow$ Coffee

Confidence= P(Coffee|Tea) = 0.75

but P(Coffee) = 0.9

$\Rightarrow$ Although confidence is high, rule is misleading

$\Rightarrow$ P(Coffee|$\overline{\text{Tea}}$) = 0.9375

LIFT EXAMPLE

	Coffee	$\overline{\text{Coffee}}$	
Tea	15	5	20
$\overline{\text{Tea}}$	75	5	80
	90	10	100

Association Rule: Tea $\rightarrow$ Coffee

Confidence= P(Coffee|Tea) = 0.75

but P(Coffee) = 0.9

$\Rightarrow$ Lift = 0.75/0.9= 0.8333 (< 1, therefore is negatively associated)

FINAL PROJECT PRESENTATION

- No class on Nov 29, 2021
- 3 classes in the next two weeks (12.1, 12.6, 12.8) will all be final project presentation sessions
    - The presentation order is out; check BrightSpace for it (in the Project Presentation Slides assignment)
    - Each team gets 6 minutes for presentation, 2 minute for Q&A
    - Zoom: https://us02web.zoom.us/j/4512436356?pwd=VXJ6ZHlKVm9HRlBvYTBPMC9wc0xMZz09 (passcode: hci)

WHAT TO INCLUDE IN YOUR PRESENTATION?

- What is the problem your team is solving in the final project?
- What are the methods you use and what are the results/finding?
- What are the insights you obtain (about the data, or about the problem domain) through your final project?
- What makes your final project different from other team’s project?

TIMELINE

- Submit your final project presentation slides through BrightSpace, before **11:59pm, Nov 30, 2021**
- Your final project report is due at 11:59pm, December 12, 2021 (submit via Blackboard)
- Each team only needs one person to submit! Do NOT have more than one person submit!