Lecture 9:

Association Rules

Outline

- Motivation
  - Market basket problem
- Main concepts
  - Support, confidence
  - Frequent itemset
- Apriori algorithm

An example

Market basket analysis:

- let us consider a set of records containing the products bought by clients of a hypermarket
- each record (transaction) contains the list of products (items) placed in a client basket

- Example:

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

- Aim: find products which are purchased together; extract useful info for marketing decisions

Motivation

Problem to solve: Given a set of “transactions”, find rules that describe the relationship between the occurrence of an “item” and the occurrences of other “items”

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

Remark: the association rules do not capture causality by only co-occurrence

A “transaction” could be:

- List of products/services purchased by a customer
- List of symptoms associated to a patient
- List of keywords or named entities (names of persons, institutions, locations) in a collection of documents
- List of actions taken by the user of a social media applications

Main concepts

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

- item
  - Element of a transaction (e.g: “water”)
  - Component of a record: attribute=value (e.g. age=very young)
- itemset = set of items
  - Example: {bread, butter, meat, water}
- k-itemset = set of k items
  - Example of a 2-itemset: {bread, water}
- frequent itemset = an itemset which appears in many transactions
  - The frequency of an itemset = number of transactions which contain the itemset
  - Example: the 2-itemset {bread,water} appears in 3 out of 4 transactions

Main concepts

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

- Association rule = IF antecedent THEN consequent (rule which contains an itemset both in the antecedent and in the consequent part)
  - Example: IF {bread,meat} THEN {water}
  - How should be interpreted?
    - When bread and meat are purchased there is a high chance to buy also water
  - How reliable is such a rule? How useful it is or how can we evaluate its quality

Main concepts

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

- Support
  - For an itemset: the ratio of transactions which contain that itemset
  - For a rule: the ratio of transactions which contain the items involved in the rule (both in the left-hand and in the right-hand side): supp(IF A THEN B)=supp({A,B})

Examples:

- 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

Main concepts

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

- Confidence of a rule (IF A THEN B)
  - the ratio between the support of the itemset {A,B} and the support of {A}: supp({A,B})/supp(A)

Examples:

- 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
  - Interpretation: in all cases when are purchased milk and bread it is also purchased water.
- R2: IF {bread, water} THEN {meat}
  - conf(R2)=supp({bread,water,meat})/supp({bread,water})=2/3=0.66

Association rule mining

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

- Input: set of transactions
- Output: set of high confidence rules S={R1,R2,….}

each rule R: IF A THEN B satisfies

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

=number of trans. containing A and B/ total number of trans > supp threshold (e.g. 0.2)

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

Remark: the thresholds for the support and confidence should be provided by the user

Association rule mining

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

Approaches in association rule mining:

- Brute force approach (first generate then filter):
  - generate all rules starting from the total set of items I
    - for each subset A of I (considered as an antecedent) select each subset B of (I-A) as consequent and generate the rule IF A THEN B
  - select those satisfying the support and the confidence requirement

- Remark: this approach has a high computational cost; if N is the total number of items, the number of generated rules is of the order

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

Association rule mining

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

Brute force approach – example:

- I={bread, butter, meat, milk, water}, N=5
- A={bread}; there are 16 subsets of I-A={butter, meat, milk, water} which can be used as consequent
- 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 (more than 500000 rules in the case of a list of 5 items)

Association rule mining

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

Brute force approach – example:

- I={bread, butter, meat, milk, water}, N=5
- A={bread}; there are 16 subsets of I-A={butter, meat, milk, water} which can be used as consequent
- 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)

Association rule mining

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

Remark:

- The support of a rule IF A THEN B is higher than a given threshold only if the support of itemset {A,B} is higher than that threshold

- Idea: it would be useful to identify first itemsets with a support higher than the threshold and then split them in the antecedent part and the consequent part in order to generate a high support rule

- For instance, it does not make sense to generate rules characterized by {A,B}={bread, butter, meat, milk, water}, as the support of this itemset is 0

(in the brute force approach there are 2^N-2N rules involving the total set of items – with the items distributed in all possible ways between the antecedent and the consequent parts)

Association rule mining

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

Approaches in association rule mining:

- Apriori approach :
  - Step 1: Find the itemsets with support higher than the specified threshold (e.g. 0.2) – these are called frequent itemsets
  - Step 2: For each itemset generate all possible rules (by distributing the elements of the itemset between the antecedent and the consequent parts of the rule) and select those with a high confidence (e.g. higher than 0.7)

- Remark: the main question is how to generate frequent itemsets without analyzing all subsets of the total set of items

Association rule mining

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

Question: How to generate frequent itemsets without analyzing all subsets of the total set of items?

Remark: any subset of a frequent itemset should also have a support higher than the threshold

Example: 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

Idea: construct the frequent itemsets in an incremental way starting from 1-itemsets (sets containing one item)

Association rule mining

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

Construction of frequent itemsets

(threshold for support: 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

Association rule mining

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

Construction of frequent itemsets

(threshold for support: 0.3)

frequent 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

Association rule mining

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

Construction of frequent itemsets

(threshold for support: 0.3)

1-itemsets

{bread} supp({bread})=0.75

{meat} supp({meat})=0.5

{water} supp({water})=1

2-itemsets

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

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

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

Association rule mining

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

Construction of frequent itemsets

(threshold for support: 0.3)

frequent 1-itemsets

{bread} supp({bread})=0.75

{meat} supp({meat})=0.5

{water} supp({water})=1

frequent 2-itemsets

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

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

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

3-itemsets

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

Association rule mining

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

All frequent itemsets with at least two items

(threshold for support: 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

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

Association rule mining

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

All frequent itemsets with at least two items

(threshold for support: 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

Association rule mining

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

All rules with high confidence
(threshold for confidence: 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

Remark: only 12 instead of more than 500000 rules are generated in order to select 7 high confidence rules

Association rule mining

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

Question: Are all high confidence rules also interesting? (an interesting rule provides non-trivial, new or unexpected information)

Example: the rule IF {bread} THEN {water} has a confidence equal to 1; does it provide some novel information?

How can be measured the interestingness (novelty) of a rule?

There are different approaches. A simple one is based on the Piatesky-Shapiro argument stating that the antecedent and the consequent of a rule should not be independent (in a statistical sense)

A rule IF A THEN B is considered interesting if the ratio (called lift or interest)

supp({A,B})/(supp(A)*supp(B)) is not close to 1

Association rule mining

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

Removing the rules with low level of interest (those for which 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

Apriori algorithm

Overall structure:

Step 1: Generate the list of frequent itemsets in an incremental way starting form 1-itemsets and using the anti-monotone property of support measure:

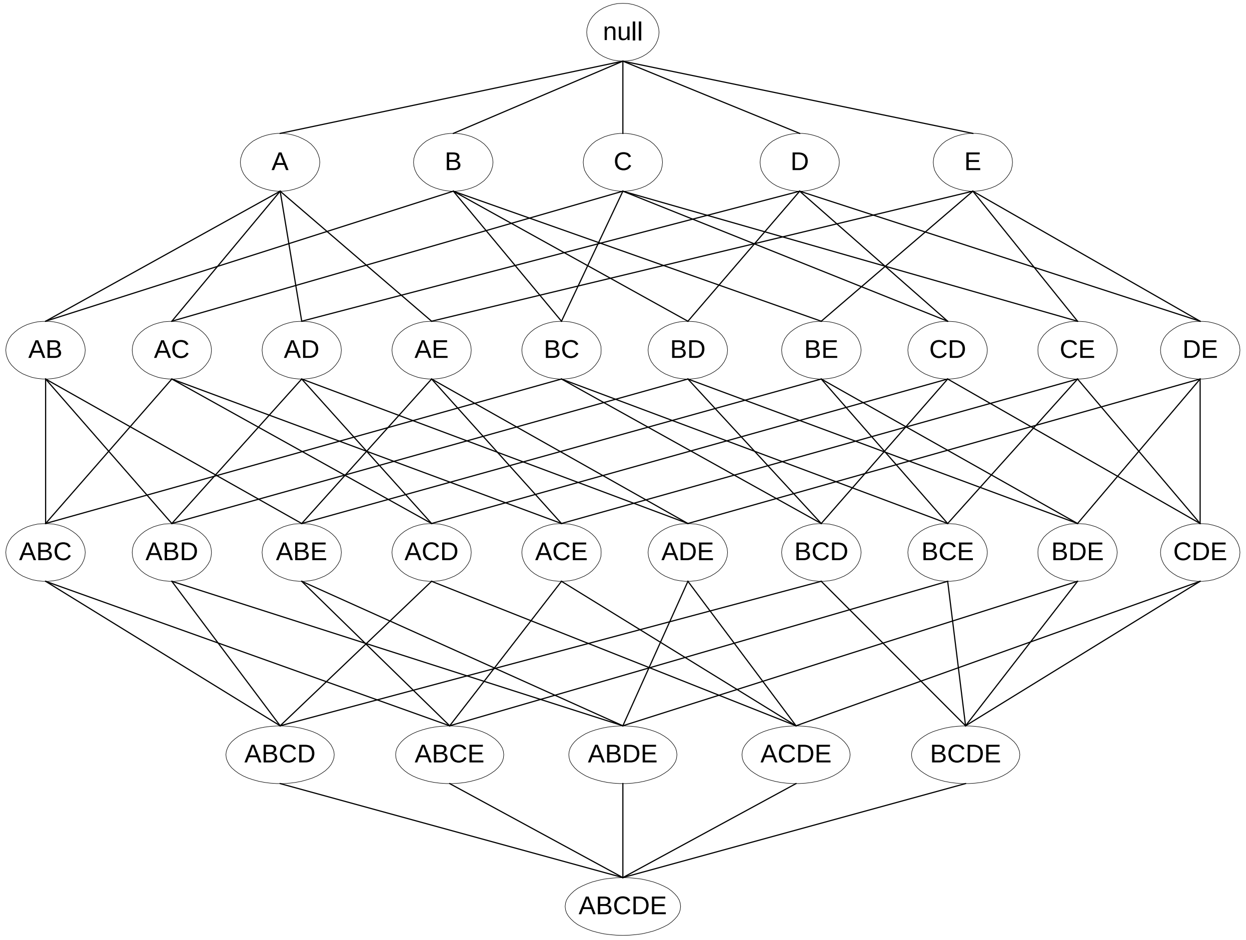
For any subset B of a set of items A: supp(B)>=supp(A)

(the main implication of this property is that when constructing an k-itemset one can use only the smaller itemsets which have a support higher or at least equal to the threshold)

Step 2: Construct the list of rules by analyzing all subsets of the frequent itemsets

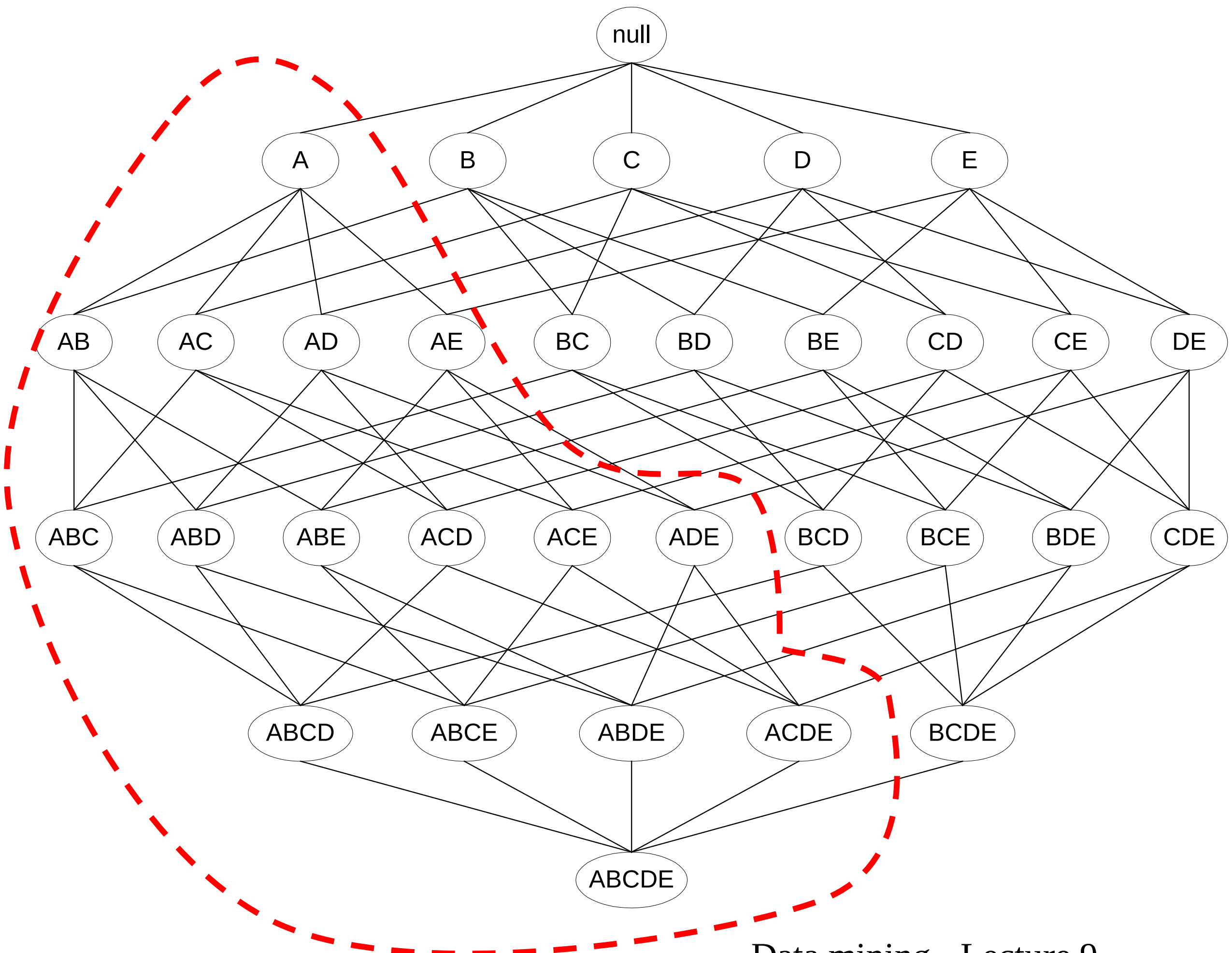
Apriori algorithm

Example: all itemsets corresponding to a list of 5 items {A,B,C,D,E}



Apriori algorithm

Example: all itemsets corresponding to a list of 5 items {A,B,C,D,E}



- If {A} is a 1-itemset with low support then the itemset search space is pruned and none of the itemsets including {A} is further generated
- In order to construct a k-itemset it is enough to join two frequent (k-1)-itemsets

Apriori algorithm

Algorithm for frequent itemsets generation:

- Let k=1
- Generate frequent itemsets of length 1

- Repeat until no new frequent itemsets are identified
  - Generate length (k+1) candidate itemsets from length k frequent itemsets (by joining two k-itemsets which have (k-1) common items)
  - Prune candidate itemsets containing subsets of length k that are infrequent
  - Count the support of each candidate by scanning the set transactions
  - Eliminate candidates that are infrequent, leaving only those that are frequent

Apriori algorithm

Algorithm for generating the rules based on the list L of frequent itemsets:

- Initialize the list LR of rules (empty list)
- FOR each itemset IS from L
  - FOR each subset A of IS construct the rule R(A,IS): IF A THEN IS-A
  - Compute the confidence of rule R(A,IS) and if the confidence is higher than the confidence threshold then add R(A,IS) to LR

Remarks:

- For a k-itemset there can be generated 2^k-2 rules (the rules with empty antecedent or empty consequent are ignored)
- In order to limit the number of rules for which the confidence should be evaluated one could use the anti-monotony property: the confidence is higher if the cardinality of the antecedent is higher, i.e

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

Apriori algorithm

Algorithm for generating the rules based on the list L of frequent itemsets:

- Initialize the list LR of rules (empty list)
- FOR each itemset IS from L
  - FOR each subset A of IS construct the rule R(A,IS): IF A THEN IS-A
  - Compute the confidence of rule R(A,IS) and if the confidence is higher than the confidence threshold then add R(A,IS) to LR

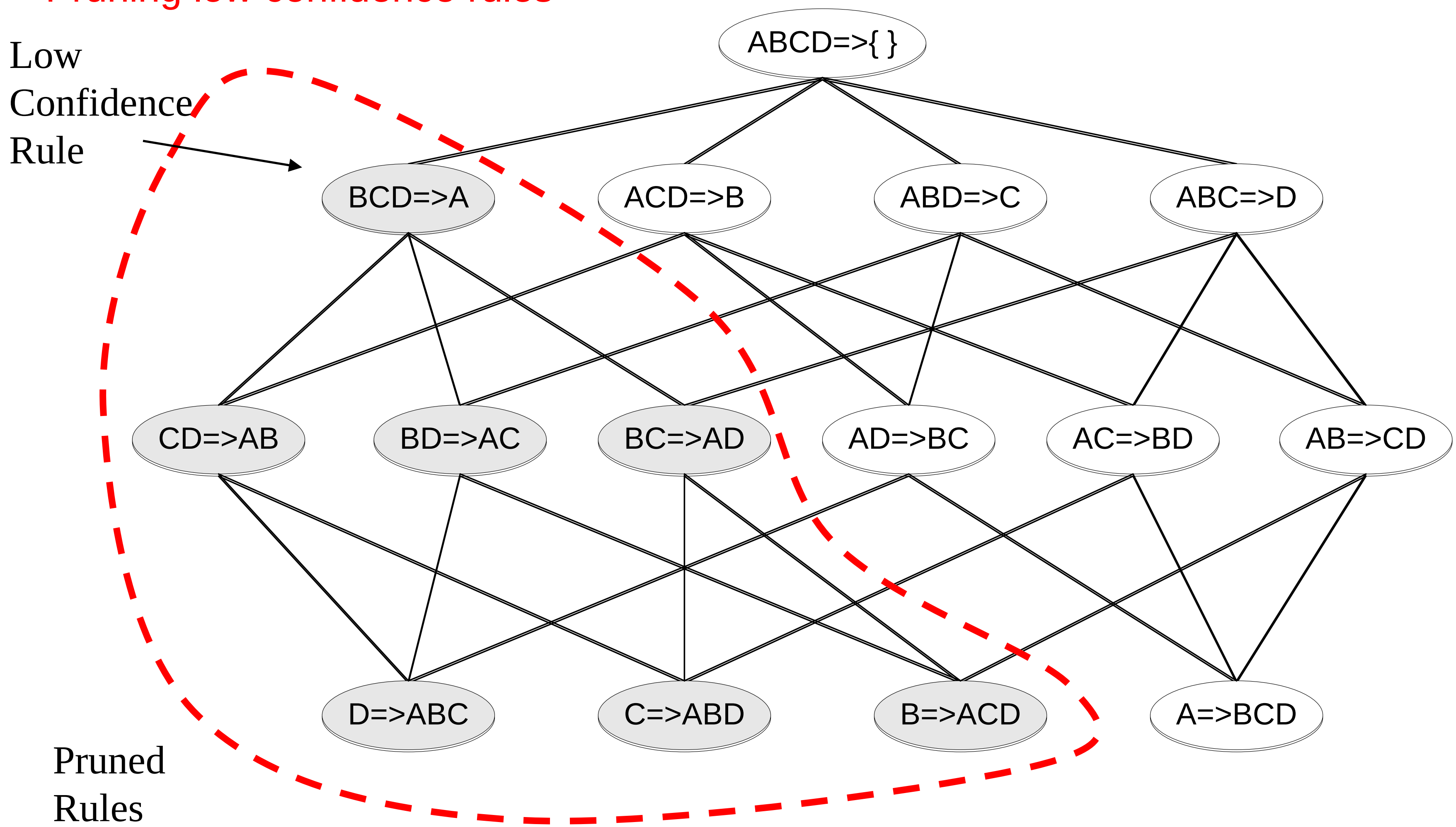
Remarks:

- For a k-itemset there can be generated 2^k-2 rules (the rules with empty antecedent or empty consequent are ignored)
- In order to limit the number of rules for which the confidence should be evaluated one could use the anti-monotony property: the confidence is higher if the cardinality of the antecedent is higher, i.e

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

Apriori algorithm

Pruning low confidence rules



Apriori algorithm

Ideas to reduce the computation during the generations of rules from frequent itemsets:

- It is more efficient to start with antecedents represented by large itemsets
- Use the idea of joining rules in order to create new rules: new candidate rules can be generated by merging two rules that share the same prefix in the consequent

Example:

- join(IF {C,D} THEN {A,B}, IF {B,D} THEN {A,C}) lead to the rule
IF {D} THEN {A,B,C}

- If the rule IF {A,D} THEN {B,C} has a confidence lower than the threshold then the joined rule should be pruned (its confidence will be also lower than the threshold)

Apriori algorithm

Influence of the thresholds:

- If *support threshold* is set too high, we could miss itemsets involving interesting rare items (e.g., expensive products)

- If *support threshold* is set too low, it is computationally expensive and the number of itemsets is very large