ASSOCIATION ANALYSIS

Prof. Pasquale Rullo

Association Analysis

- **Descriptive** analysis, used for discovering interesting relationships hidden in large data bases
- Descriptive vs predictive (classification)
- The relationships are expressed in terms of association rules X ➔ Y, where X and Y are sets of objects (items)
- An association rule is a probabilistic implication

The market basket analisys

Transactions

TID	Items
1	{bread, milk}
2	{bread, beer, diapers, eggs}
3	{milk, diapers, beer, cola}
4	{bread, milk, diapers, beer}
5	{bread, milk, diapers, cola}

Association rules:

- {bread} ➔ {milk} (3/4)
- {beer} ➔ {diapers} (3/3)
- {diapers} ➔ {beer} (3/4)
- {diapers, bread} ➔ {milk} (2/3)

The market basket analisys

Transactions

TID	Items
1	{bread, milk}
2	{bread, beer, diapers, eggs}
3	{milk, diapers, beer, cola}
4	{bread, milk, diapers, beer}
5	{bread, milk, diapers, cola}

Association rules:

- {bread} ➔ {milk} (3/4)
- {beer} ➔ {diapers} (3/3)
- {diapers} ➔ {beer} (3/4)
- {diapers, bread} ➔ {milk} (2/3)

Basic definitions

- Beer, bread, etc. are called **items**
- An **itemset** is any set of items
- A **transaction is** <Tid, itemset>
- An **association rule** is of the form

$$X \rightarrow Y$$

- where X (antecedent) and Y (consequent) are disjoint itemsets
- An association rule can be seen as a probabilistic implication

Quality of rules - confidence

- A transaction T satisfies a rule $X \rightarrow Y$ if both $X \subseteq T$ and $Y \subseteq T$
- The confidence of a rule

$$X \rightarrow Y$$

- is the conditional probability

$$p(Y \subseteq T | X \subseteq T) = \sigma(X \cup Y) / \sigma(X)$$

- that is, the number of transactions satisfying the rule over the number of transactions containing only the antecedent X
- Confidence measures the reliability of an implication

Quality of rules - confidence

Transactions

TID	Items
1	{bread, milk}
2	{bread, beer, diapers, eggs}
3	{milk, diapers, beer, cola}
4	{bread, milk, diapers, beer}
5	{bread, milk, diapers, cola}

- Conf({beer} ➔ {diapers}) = 3/3 = 1
- Conf({bread} ➔ {milk}) = 3/4 = 0.75
- Conf({diapers} ➔ {beer}) = 3/4 = 0.75
- Conf({milk, diapers} ➔ {beer}) = 2/3 = 0.66

Quality of rules - support

- The support of a rule

$$X \rightarrow Y$$

- is the probability $p(X \cup Y \subseteq T) = \sigma(X \cup Y)/ N$, where N is the number of transactions - that is, the number of transactions satisfying the rule over the total number of transactions

Quality of rules - support

TID	Items
1	{bread, milk}
2	{bread, beer, diapers, eggs}
3	{milk, diapers, beer, cola}
4	{bread, milk, diapers, beer}
5	{bread, milk, diapers, cola}

- Supp({beer} ➔ {diapers}) = 2/5
- Supp({bread} ➔ {milk}) = 3/5
- Conf({diapers} ➔ {beer}) = 3/5
- Conf({milk, diapers} ➔ {beer}) = 2/5

- A rule that has very low support may occur only by chance

Association rule mining

- **Problem**: given a set of transactions, find all rules having support not less than *minsupp* and confidence not less than *minconf*, where *minsupp* and *minconf* are the support and confidence thresholds, respectively
- NP-hard problem
- Heuristic approach needed

Association rule mining – decompose the problem

- **Input**: set of transactions, along with support and confidence thresholds

1. **Frequent itemset generation**: find all itemsets that satisfy the support threshold (*frequent itemsets*)

2. **Rule generation**: extract from the frequent itemsets all rules that satisfy the confidence threshold

Association rule mining – decompose the problem

- **Input**: set of transactions, along with support and confidence thresholds

1. **Frequent itemset generation**: find all itemsets that satisfy the support threshold (*frequent itemsets*)

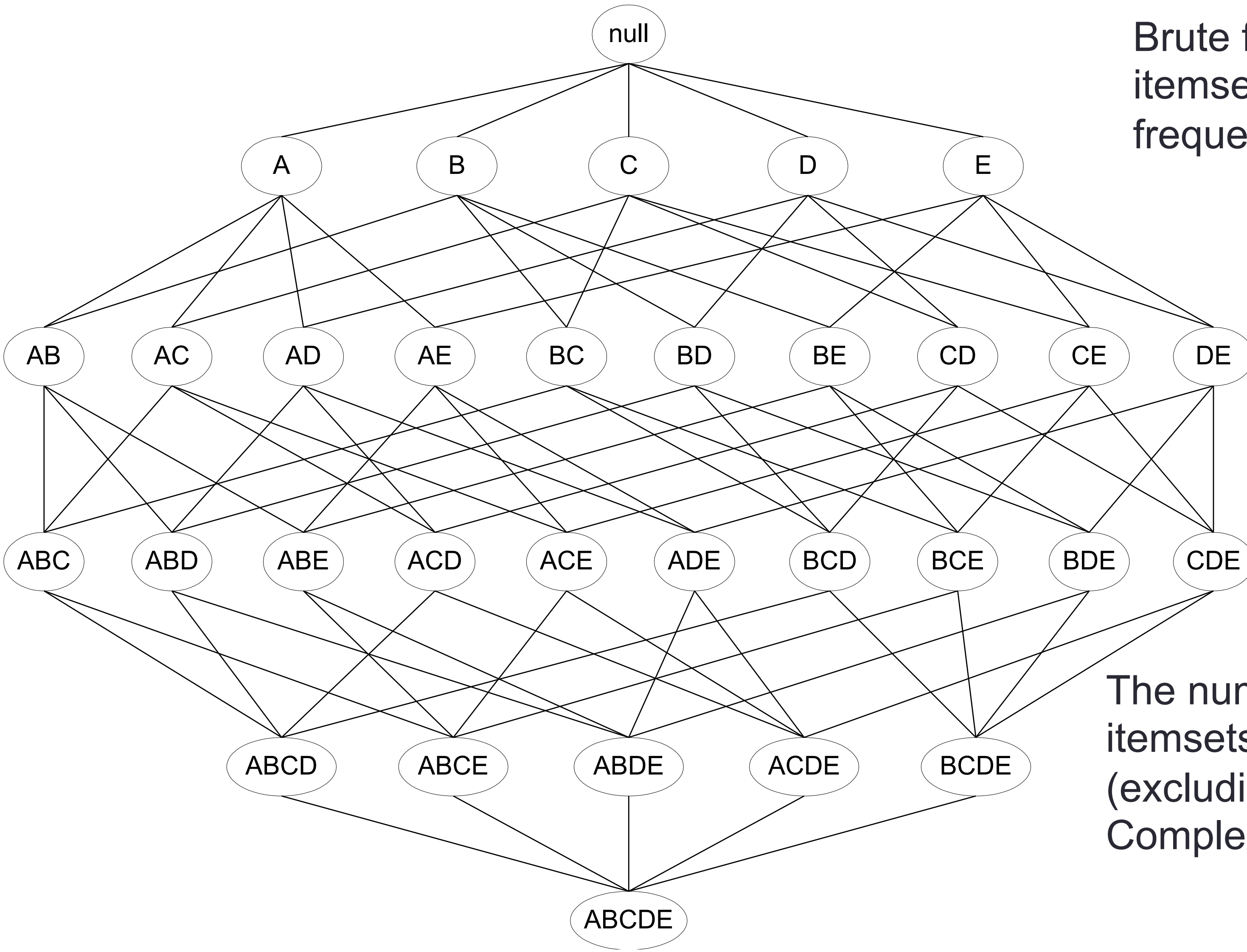
2. **Rule generation**: extract from the frequent itemsets all rules that satisfy the confidence threshold

Frequent itemset generation

- S = {beer, diapers, milk}
- Supp(S) = 2/5 = 0.4
- All rules involving all items in S, e.g.,
  - {beer, diapers} ➔ {milk},
  - {beer, milk} ➔ {diapers}, …
- have the same support 0.4 of S
- If S is frequent, then all rules from S are frequent

TID	Items
1	{bread, milk}
2	{bread, beer, diapers, eggs}
3	{milk, diapers, beer, cola}
4	{bread, milk, diapers, beer}
5	{bread, milk, diapers, cola}

Frequent itemset generation – brute force



Brute force: generate all itemsets and select the frequent ones

The number of frequent itemsets is up to 2^n-1 (excluding the empty set) - Complexity $O(2^n)$

Frequent itemset generation - Apriori Principle

- The **Apriori Principle**: if an itemset is frequent, then all of its subsets are frequent

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

- If an itemset is infrequent, all its supersets are infrequent

Frequent itemset generation – Apriori Principle

TID	Items
1	{bread, milk}
2	{bread, beer, diapers, eggs}
3	{milk, diapers, beer, cola}
4	{bread, milk, diapers, beer}
5	{bread, milk, diapers, cola}

- supp({beer, diapers}) = 3/5 =0.6
- supp({beer}= 3/5 = 0.6
- supp({diapers}} = 4/5 = 0.8
- Thus, if {beer} has a support less than, say, 0.7, then any itemset containing beer will have support less than (or equal to) 0.7

Frequent itemset generation – Apriori Principle

TID	Items
1	{bread, milk}
2	{bread, beer, diapers, eggs}
3	{milk, diapers, beer, cola}
4	{bread, milk, diapers, beer}
5	{bread, milk, diapers, cola}

- supp({beer, diapers}) = 3/5 =0.6
- supp({beer}= 3/5 = 0.6
- supp({diapers}} = 4/5 = 0.8
- Thus, if {beer} has a support less than, say, 0.7, then any itemset containing beer will have support less than (or equal to) 0.7

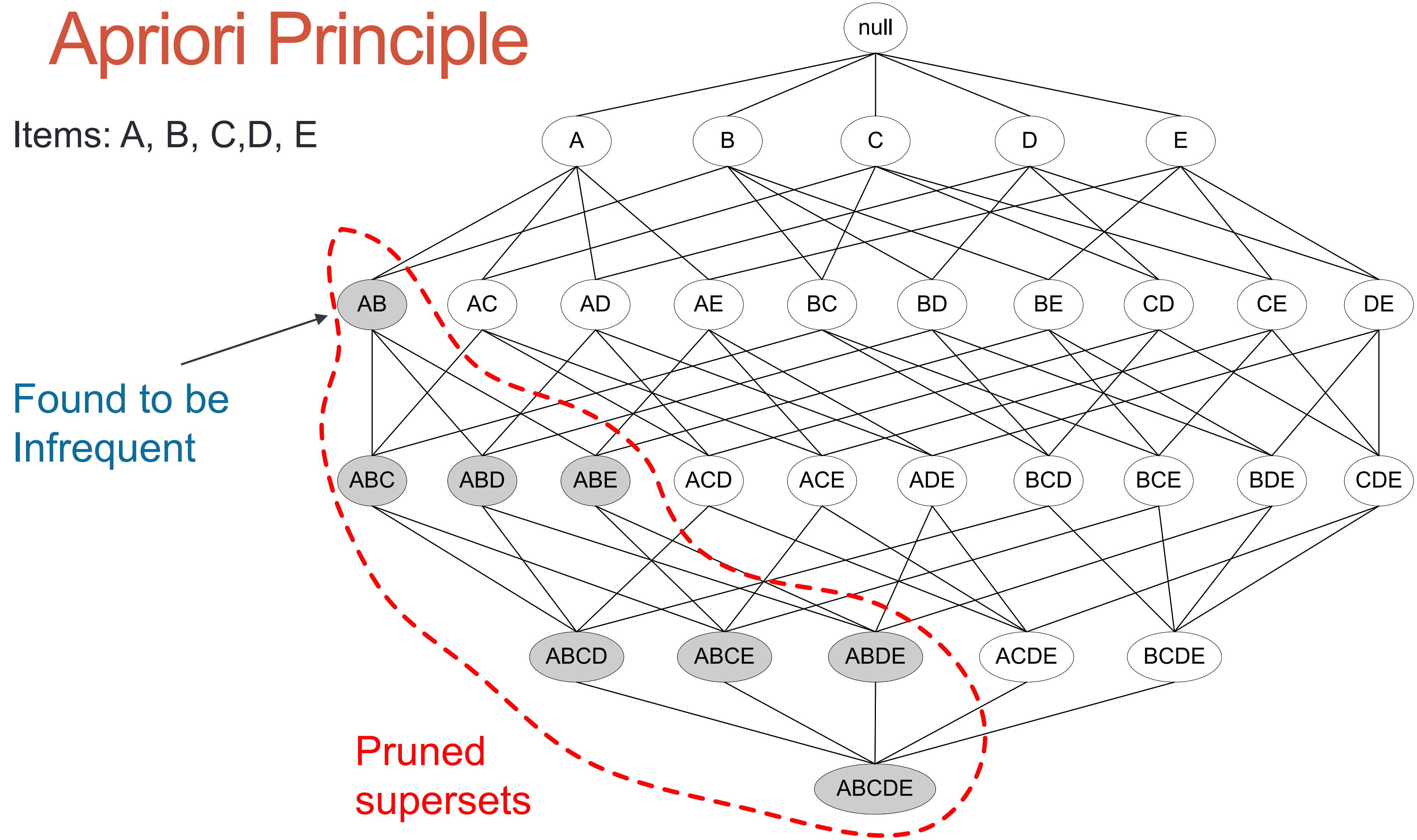
Frequent itemset generation – Apriori Principle

TID	Items
1	{bread, milk}
2	{bread, beer, diapers, eggs}
3	{milk, diapers, beer, cola}
4	{bread, milk, diapers, beer}
5	{bread, milk, diapers, cola}

- supp({beer, diapers}) = 3/5 =0.6
- supp({beer}= 3/5 = 0.6
- supp({diapers}} = 4/5 = 0.8
- Thus, if an itemset S has support supp(S), then any superset S will have support less or equal to supp(S)

Frequent itemset generation - Apriori Principle

Items: A, B, C,D, E



Frequent itemset generation Apriori algorithm – an example

- Assume that the support threshold is 60%

TID	Items
1	{bread, milk}
2	{bread, beer, diapers, eggs}
3	{milk, diapers, beer, cola}
4	{bread, milk, diapers, beer}
5	{bread, milk, diapers, cola}

Frequent itemset generation
Apriori principle– an example

- Candidate 1-itemsets

1-itemset	count	support
beer	3	3/5 = 0.6
bread	4	4/5 = 0.8
Cola*	**2**	**2/5 = 0.4**
diapers	4	4/5 = 0.8
milk	4	4/5 = 0.8
Eggs*	**1**	**1/5 = 0.2**

TID	Items
1	{bread, milk}
2	{bread, beer, diapers, eggs}
3	{milk, diapers, beer, cola}
4	{bread, milk, diapers, beer}
5	{bread, milk, diapers, cola}

* Below the required support, thus discarded

Frequent itemset generation Apriori principle– an example

- Candidate 2-itemsets

2-itemsets	count	support
{beer, bread}	**2**	**2/5 = 0.4**
{beer, diapers}	3	3/5 = 0.6
{beer, milk}	**2**	**2/5 = 0.4**
{bread, diapers}	3	3/5 = 0.6
{bread, milk}	3	3/5 = 0.6
{diapers, milk}	3	3/5 = 0.6

TID	Items
1	{bread, milk}
2	{bread, beer, diapers, eggs}
3	{milk, diapers, beer, cola}
4	{bread, milk, diapers, beer}
5	{bread, milk, diapers, cola}

There are binCoef(4,2) = 6
2-itemsets

Frequent itemset generation Apriori principle– an example

output

- Frequent 2-itemsets

2-itemsets	count
{beer, diapers}	3/5=0.6
{bread, diapers}	3/5=0.6
{bread, milk}	3/5=0.6
{diapers, milk}	3/5=0.6

3-itemsets	count
{beer, diapers, milk}	**1/5**
{beer, bread, diapers}	**2/5**
{bread, diapers, milk}	**2/5**
{beer, bread, milk}	**1/5**

1-itemsets
beer
bread
diapers
milk

No 3-itemset satisfies the support constraint

Frequent itemset generation
Apriori principle

1. The algorithm initially makes a single pass over the data set to determine all items having support not less than the required support
2. Next, the algorithm will iteratively generate new candidate k-itemsets using the frequent (k-1)-itemsets found in the previous iteration
3. After counting the support of each generated k-itemset, the algorithm eliminates those not meeting the support threshold
4. The algorithm terminates when there are no new frequent itemsets generated

Frequent itemset generation
Apriori principle

- **Algorithm**
- Start with individual items with support ≥ *minSupp*
- In each next step, k,
  - Use itemsets from step k-1 to generate new itemsets
  - For each new itemset, compute its support
  - Prune the ones that are below the threshold *minSupp*

Frequent itemset generation Apriori principle

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

- If *minsup* is set too low, it is computationally expensive as the number of itemsets is very large; further, rules that may occur only by chance can be generated

Association rule mining – decompose the problem

- **Input**: set of transactions, along with support and confidence thresholds

1. **Frequent itemset generation**: find all itemsets that satisfy the support threshold (*frequent itemsets*)

2. **Rule generation**: extract from the frequent itemsets all rules that satisfy the confidence threshold

Rule generation

- Rules are generated starting from frequent itemsets (why?)
- Let Y be a frequent k-itemset; there exist 2^k-2 rules of the form

$$X \rightarrow Y\text{-}X, \text{ where } X \subseteq Y$$

- **Example**: Y = {1, 2, 3}. There are 6 rules

$$\{1\} \rightarrow \{2, 3\},\ \{2\} \rightarrow \{1, 3\},\ \{3\} \rightarrow \{1, 2\}$$
$$\{1, 2\} \rightarrow \{3\},\ \{1,3\} \rightarrow \{2\},\ \{2,3\} \rightarrow \{1\}$$

Rule generation

- The support of each rule coming from an itemset Y is constant and equal to that of Y

$$\mathrm{supp}(X \cup (Y\text{-}X)) = \mathrm{supp}(Y)$$

- Therefore, each rule generated from a frequent itemset will be frequent, i.e., satisfies the support threshold *minSupp*

Rule Generation
pruning the search space

- Problem: generating all 2^k-2 rules from a (frequent) itemset is prohibitive
- We are interested only on rules satisfying the confidence constraint
- **Theorem**: If a rule r: X ➔ Y-X does not satisfy the confidence threshold, then any rule r': X' ➔ Y-X', where X' $\subseteq$ X, does not satisfy the confidence threshold as well
- Proof
  - conf(r) = $\sigma(X \cup Y) / \sigma(X) = \sigma(Y) / \sigma(X)$
  - conf(r') = $\sigma(X' \cup Y) / \sigma(X') = \sigma(Y) / \sigma(X')$
  - $X' \subseteq X$ => $\sigma(X') \geq \sigma(X)$ (apriori principle) => **conf(r) ≥ conf(r')**

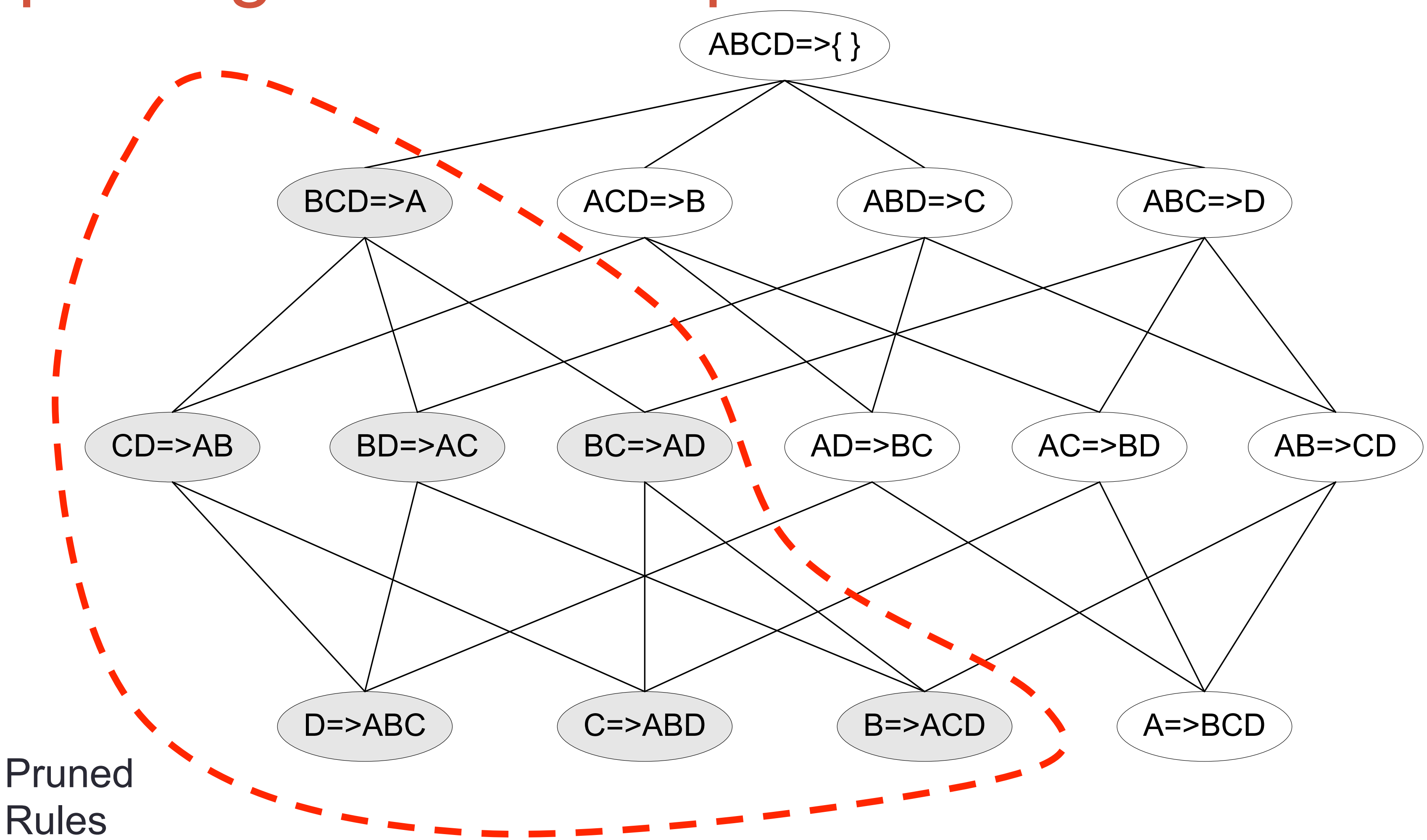
Rule Generation
pruning the search space

- According to the above theorem, given Y = {A,B,C,D}, the following holds:

$$\text{conf}(ABC \rightarrow D) \geq \text{conf}(AB \rightarrow CD) \geq \text{conf}(A \rightarrow BCD)$$

- Given a rule r from Y, the larger the antecedent (and the smaller the consequent), the more confident r
- The most confident rules are those with one item in the consequent

Rule Generation
pruning the search space



Rule Generation
pruning the search space

- Level-wise approach for generating high-confidence rules
- The most confident rules are those with one item in the consequent (level 1)
- If any node in the lattice has low confidence, according to the above theorem, the entire subgraph spanned by the node is pruned.

Limitation of the support-confidence framework

- Support and confidence measures are in general used to eliminate uninteresting patterns
- The resulting rules may be misleading, uninteresting or redundant
- Other measures, like Interest Factor and Correlation Analysis, can be used

Example – the congressional voting records (Tan – pag 352)

- **Data set**: voting records of members of the USA House of Representative. Each transaction contains information about the party affiliation for a representative, along with his/her voting record on 16 issues
- **Goal**: inducing association rules showing the key issues dividing Democrats from Republicans

Association rule	Confidence
Budget_resolution=no, MX-missile=no, aid-Salvador=yes → republican	91%
Budget_resolution=yes, MX-missile=yes, aid-Salvador=no→ democrat	97,5%
….	

Exercise

Day	Hour	Web pages
01/03/2011	11.35	Home, News, Faq
01/03/2011	11.40	Forum, News, Faq
02/03/2011	9.45	Faq, Forum
02/03/2011	23.30	Download
03/03/2011	21.25	Faq, Download
04/03/2011	16.40	Home, download

Given the above data concerning the accesses to the pages of a website, Determine which page associations have support greater than 40% and Confidence greater than 70%

References

- Agrawal R, Imielinski T, Swami AN. "Mining Association Rules between Sets of Items in Large Databases." SIGMOD. June 1993, 22(2):207-16, pdf.
- Agrawal R, Srikant R. "Fast Algorithms for Mining Association Rules", VLDB. Sep 12-15 1994, Chile, 487-99, pdf, ISBN 1-55860-153-8.
- Mannila H, Toivonen H, Verkamo AI. "Efficient algorithms for discovering association rules." AAAI Workshop on Knowledge Discovery in Databases (SIGKDD). July 1994, Seattle, 181-92, ps.
- Tan, Steinbach, Kumar, Introduction to Data Mining, Addison Wesley