Itemset and Association Rule Mining

Yannick Toussaint
ENSMN R407, LORIA B160
Yannick.Toussaint@loria.fr

Automn 2018

Summary of the lecture

Itemset and Association Rule Mining

- A symbolic approach: when some interpretation is needed
- Knowledge representation and reasoning: Bridging the gap with semantic web technologies, description logics, classification and case-based reasoning, ontology engineering
- Including some background knowledge such as: *a dog is an animal*, fluoroquinoles are quinolones...
- Applications in agronomy, astronomy, biology, chemistry, cooking, medicine, pharmacogenomics...

Facebook would have some 1300 symbolic features describing users.

Extracting Itemsets

What can say a binary table?

Extracting itemsets from a binary table

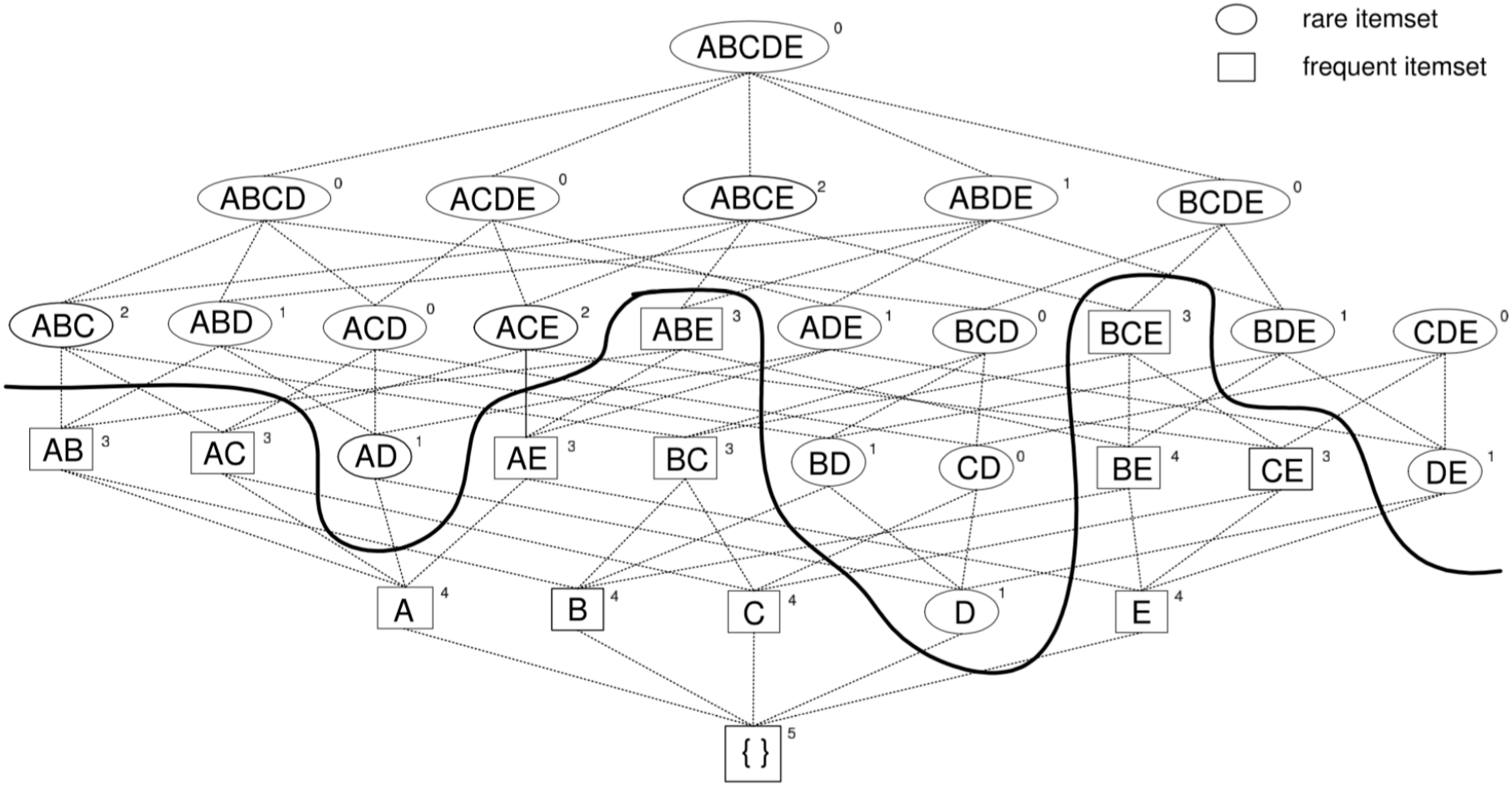
We consider a set of objects O, a set of attributes (or items) A, and a relation $R \subseteq O \times A$, where $R(o, a)$ means that the object o has the attribute a.

- An itemset is any subset of attributes.
- The support of an itemset indicates how many objects include the itemset. An itemset is called frequent, if its support is $\geqslant \sigma_s$.

Objects / Items	a	b	c	d	e
o1	x	x		x	x
o2	x		x		
o3	x	x	x		x
o4		x	x		x
o5	x	x	x		x

Extracting Itemsets

- Frequent versus rare itemsets ($\sigma_s = 3$)
- If $|A| = n$, the number of potential itemsets is equal to $2n$



Extracting Itemsets

Objects / Items	a	b	c	d	e
o1	x	x		x	x
o2	x		x		
o3	x	x	x		x
o4		x	x		x
o5	x	x	x		x

Itemsets extracted ($\sigma_S = 2$):

- Itemsets of length 1: $\{a\}$ (4), $\{b\}$ (4), $\{c\}$ (4), $\{e\}$ (4).
- Itemsets of length 2: $\{ab\}$ (3), $\{ac\}$ (3), $\{ae\}$ (3), $\{bc\}$ (3), $\{be\}$ (4), $\{ce\}$ (3),
- Itemsets of length 3: $\{abc\}$ (2), $\{abe\}$ (3), $\{ace\}$ (2), $\{bce\}$ (3),
- Itemsets of length 4: $\{abce\}$ (2),
- Itemsets of length 5: -

The support is a monotonously decreasing function.

Extracting Itemsets

- Heuristics have to be used for pruning the set of all itemsets to be tested
- Levelwise search of frequent itemsets: the **Apriori algorithm** [Agrawal et al. 93]
- 1. Every sub-itemset of a frequent itemset is a frequent itemset,
  2. Every super-itemset of an infrequent itemset is infrequent.

Extracting Itemsets

Apriori can be summarized as follows:

- The search for frequent itemsets begins with the search for frequent itemsets of length 1.
- The frequent itemsets are recorded and combined together to form candidate itemsets of greater length.
  - Infrequent itemsets are discarded and by consequence, all their super-itemsets
  - Candidate itemsets are then tested, and the process continues in the same way, until no more candidates can be formed.
- When data to be mined are huge, there is a need for minimizing the access to the data for calculating the support.

Association Rules

Association rules

An association rule has the form $A \longrightarrow B$, where A and B are two itemsets.

- The support of the rule $A \longrightarrow B$ is defined as the support of the itemset $A \cup B$.
- The confidence of a rule $A \longrightarrow B$ is defined as the quotient $\frac{supp(A \cup B)}{supp(A)}$.
- The confidence can be seen as a conditional probability $P(B|A)$, i.e. probability of B knowing A.

.

Support may be relative, *i.e.* the proportion of the set of objects.

Association rules

- Given the two thresholds σ_s (support) and σ_c (confidence), a rule $A \longrightarrow B$ is said to be valid (or strong) if
  - $supp(A \longrightarrow B) \geq \sigma_s$
  - $conf(A \longrightarrow B) \geq \sigma_c$
- A valid rule can only be extracted from a frequent itemset.
- A rule is said to be exact if its confidence is equal to 1, i.e. $supp(A \cup B) = supp(A)$, otherwise the rule is approximate.

Association rules

Objects / Items	a	b	c	d	e
o1	x	x		x	x
o2	x		x		
o3	x	x	x		x
o4		x	x		x
o5	x	x	x		x

For example,

- with $\sigma_s = 3$ and $\sigma_c = 3/5$
  - ac is frequent
  - $a \longrightarrow c$ is valid (with support 3 and confidence 3/4)
  - $c \longrightarrow a$ is valid (with support 3 and confidence 3/4)
- with $\sigma_s = 1$ and $\sigma_c = 3/5$
  - abd is frequent
  - $d \longrightarrow ab$ is valid (with support 1 and confidence 1)
  - $ab \longrightarrow d$ is not valid (with support 1 and confidence 1/3)

Association rules

Generation of valid association rules

- From a frequent itemset P (of length necessarily greater than or equal to 2)
- Extraction starts by generating the valid rules with a right hand side (conclusion) of length 1,
  - rules of the form $P \setminus \{i\} \longrightarrow \{i\}$
  - where $\{i\}$ is an item of length 1
  - $P \setminus \{i\}$ denotes the itemset P without the item $\{i\}$
- Then, the conclusions of the valid rules $P \setminus \{i\} \longrightarrow \{i\}$ are combined for generating the candidate conclusions of length 2,(check confidence)
  - $P \setminus \{ij\} \longrightarrow \{ij\}$
  - and the process continues until no more valid rules can be generated from the frequent itemset.

Association rules

Example (with $\sigma_s = 2$ and $\sigma_c = 2/5$)

Objects / Items	a	b	c	d	e
o1	x	x		x	x
o2	x		x		
o3	x	x	x		x
o4		x	x		x
o5	x	x	x		x

- When $P = \{ab\}$, the generated valid rules are:
  - $\{a\} \longrightarrow \{b\}$ (supp= 3; conf= 3/4)
  - $\{b\} \longrightarrow \{a\}$ (3; 3/4)

Association rules

Example (with $\sigma_s = 2$ and $\sigma_c = 2/5$)

Objects / Items	a	b	c	d	e
o1	x	x		x	x
o2	x		x		
o3	x	x	x		x
o4		x	x		x
o5	x	x	x		x

- When $P = \{abc\}$, the generated valid rules are:
  - $\{ab\} \longrightarrow \{c\}$ (2; 2/3),
  - $\{ac\} \longrightarrow \{b\}$ (2; 2/3),
  - $\{bc\} \longrightarrow \{a\}$ (2; 2/3)

- As $\{a, b, c\}$ has three valid conclusions, they can be combined for producing the new conclusions $\{ab, ac, bc\}$ and generate the rules (cnfidence should be checked):
  - $\{c\} \longrightarrow \{ab\}$ (2; 2/4)
  - $\{b\} \longrightarrow \{ac\}$ (2; 2/4)
  - $\{a\} \longrightarrow \{bc\}$ (2; 2/4)

Measures associated to association rules

There exist a number of possible measures that can be attached to an association rule [LFZ99, CNT03]:$A \longrightarrow B$

- The confidence of the rule : the conditional probability $P(B|A)$, range $[0, 1]$.
- The interest or lift of the rule $A \longrightarrow B$ measure is defined as $P(A \cup B)/P(A) \times P(B)$
  - i.e. the interest measures the degree of compatibility of A and B, i.e. the simultaneous occurrences of both events A and B.
  - The interest measures the degree of independence of the attributes A and B. The interest is symmetrical ($\text{lift}(A \longrightarrow B) = \text{lift}(B \longrightarrow A)$)
  - its range in the interval $[0, +\infty[$
  - It is equal to 1 whenever the "events" A and B are statistically independent.

A rule may have a high confidence but a low lift

Measures associated to association rules

- The conviction of the rule $A \longrightarrow B$ is defined as

$$P(A) \times P(\overline{B})/P(A \cup \overline{B})$$

  - Conviction was developed as an alternative to confidence which was found to not capture direction of associations adequately
  - Conviction measures the deviation of the rule $A \longrightarrow B$ by taking into account the rule $A \longrightarrow \overline{B}$. The $\overline{B}$ means that at least one item of B is not present.
  - range of conviction is $[0, +\infty[$
  - Conviction is not computable for exact rules because $P(A \cup \overline{B})$ is equal to 0 (there is no counterexample for exact rules);

Measures associated to association rules

- The dependency of the rule $A \longrightarrow B$ is defined as

$$|P(B|A) - P(B)| = |P(A \cup B)/P(A) - P(B)|$$

i.e.

  - The dependency measures the distance between the confidence of the rule and the independence case
  - Range is [0, 1[
  - A dependency close to 0 (respectively to 1) means that A and B are independent (respectively dependent);

Domain of application

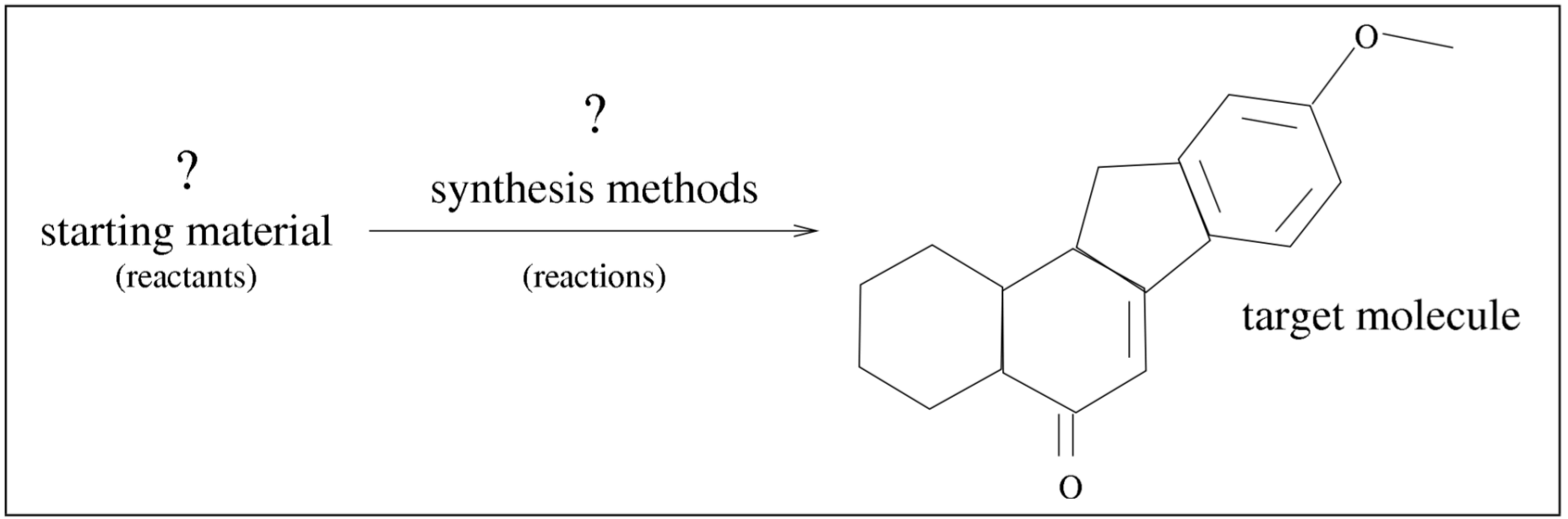
- Mining Chemical Reaction Database
- An Experiment in Biology

Mining Chemical Reaction Database

- Knowledge discovery algorithms for mining chemical reaction databases [BLNN04b]
- Synthesis planning is mainly based on retrosynthesis, i.e. a goal-directed problem-solving approach, where the target molecule is iteratively transformed by applying reactions for obtaining simpler fragments, until finding accessible starting materials
- For a given target molecule, a huge number of starting materials and reactions may exist, e.g. thousands of commercially available chemical compounds. Thus, exploring all the possible pathways issued from a target molecule leads to a combinatorial explosion, and needs a strategy for choosing reaction sequences to be used within the planning process.

Mining Chemical Reaction Database

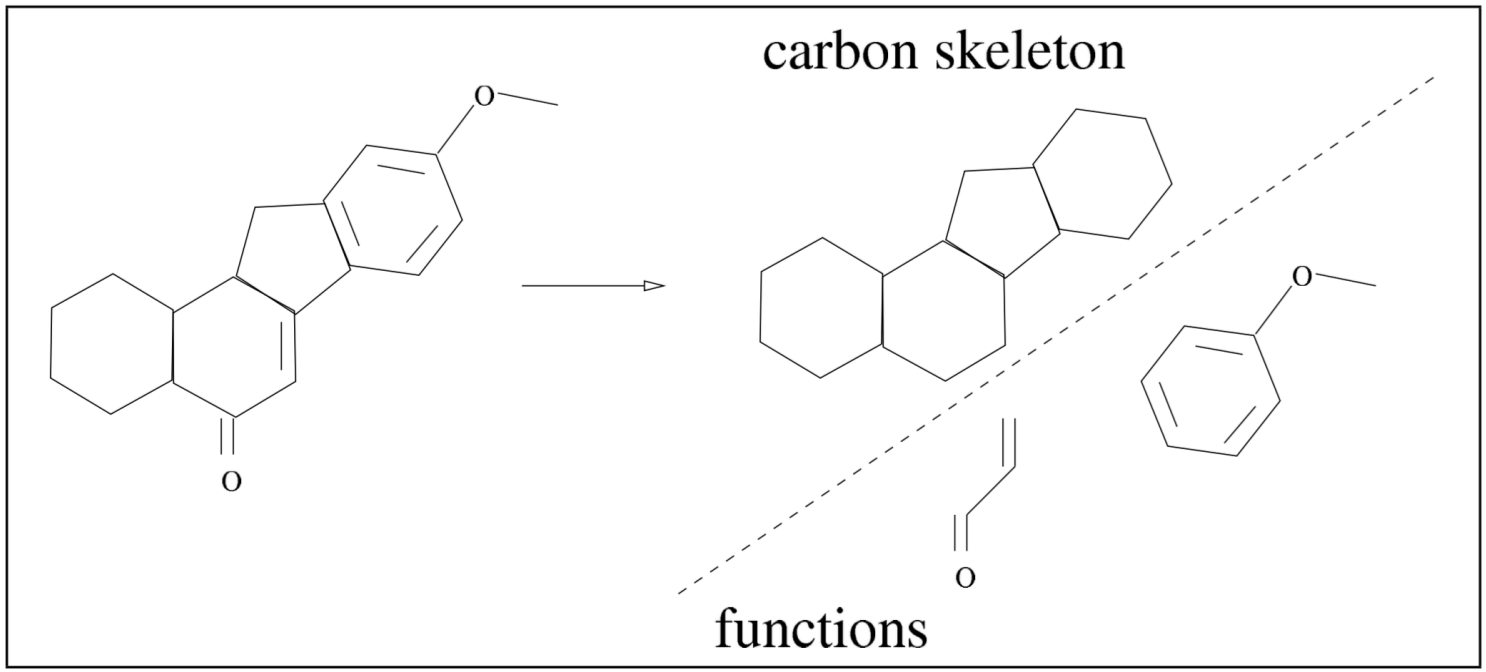
Discovering generic reactions – called synthesis methods – from chemical reaction databases in order to design generic and reusable synthesis plans.



Mining Chemical Reaction Database

- The main questions for the synthesis chemist are related to chemical families to which a target molecule belongs, i.e. the molecule that has to be built, and to the reactions or sequence of reactions building structural patterns, to be used for building these families.
- Two main categories of reactions may be distinguished:
  - reactions building the skeleton of a molecule the arrangement of carbon atoms on which relies a molecule
  - and reactions changing the functionality of a molecule, i.e. changing a function into another function

Mining Chemical Reaction Database



- Interest in reactions changing the functionality: what are the reactions allowing the transformation of a function F_i into a function F_j ?

Mining Chemical Reaction Database

Two reaction databases,

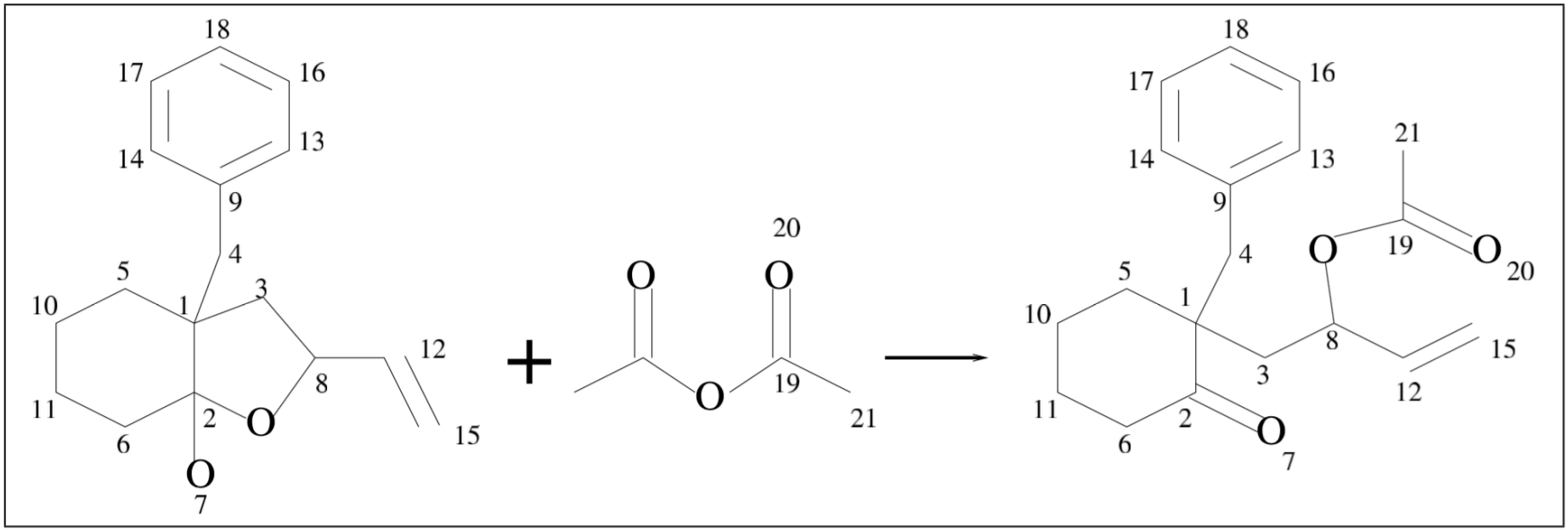
- Organic Syntheses database orgsyn-2000 including 5,486 records
- Journal of Synthetic Methods database jsm-2002 including 75,291 records

Every record contains one chemical equation involving structural information:

- Transformation of an initial state or the set of reactants
- Into a final state – or the set of products – associated with an atom-to-atom mapping between the initial and final states

Mining Chemical Reaction Database

Reaction #13426 in the jsm-2002 database:



Mining Chemical Reaction Database

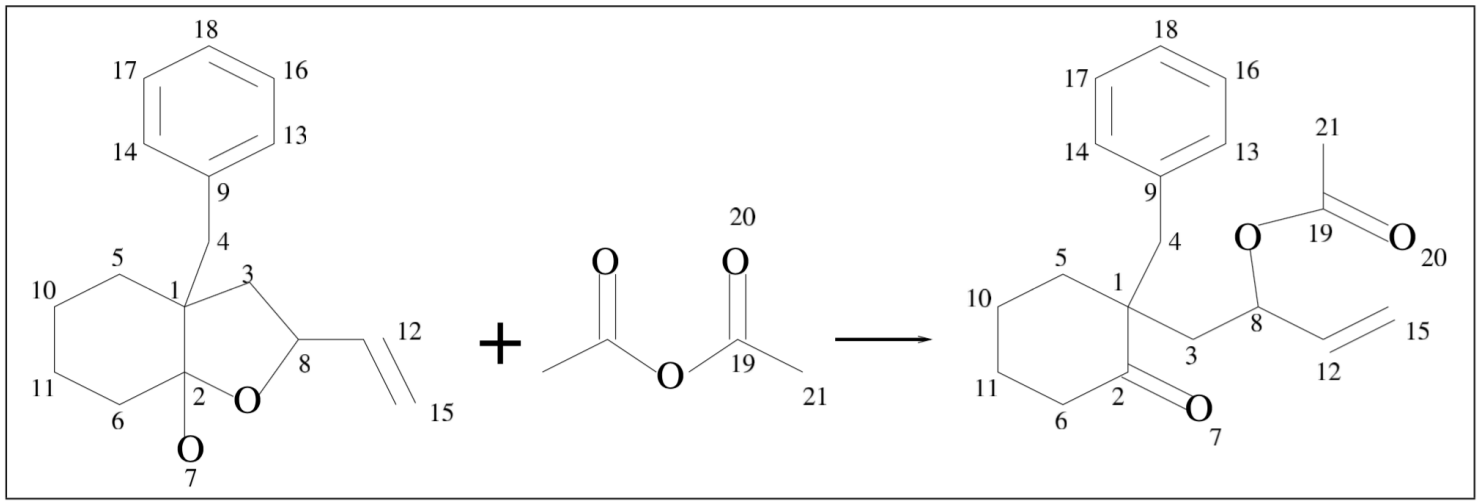
Preprocessing: Improve the quality of the selected data by cleaning and normalizing the data

- Exporting and analyzing the structural information recorded in the databases
- Extracting and representing the functional transformations in a target format

Mining Chemical Reaction Database

The considered transformations are functional modifications

- Addition of a function
- Deletion of a function
- Reactions have been considered at an abstract level (*block level*)thanks to Resyn-Assistant



Mining Chemical Reaction Database

The resyn-assistant system [VL00] has been used for

- Recognize the building blocks of reactions
- Based on the atom-to-atom mapping, the system establishes the correspondence between the recognized blocks of the same nature, and determines their role in the reaction

A function may be present:

- Only in a reactant $\longrightarrow$ the function in the reactant is destroyed
- Only in a product $\longrightarrow$ the function in the product is formed
- In both $\longrightarrow$ the function is unchanged

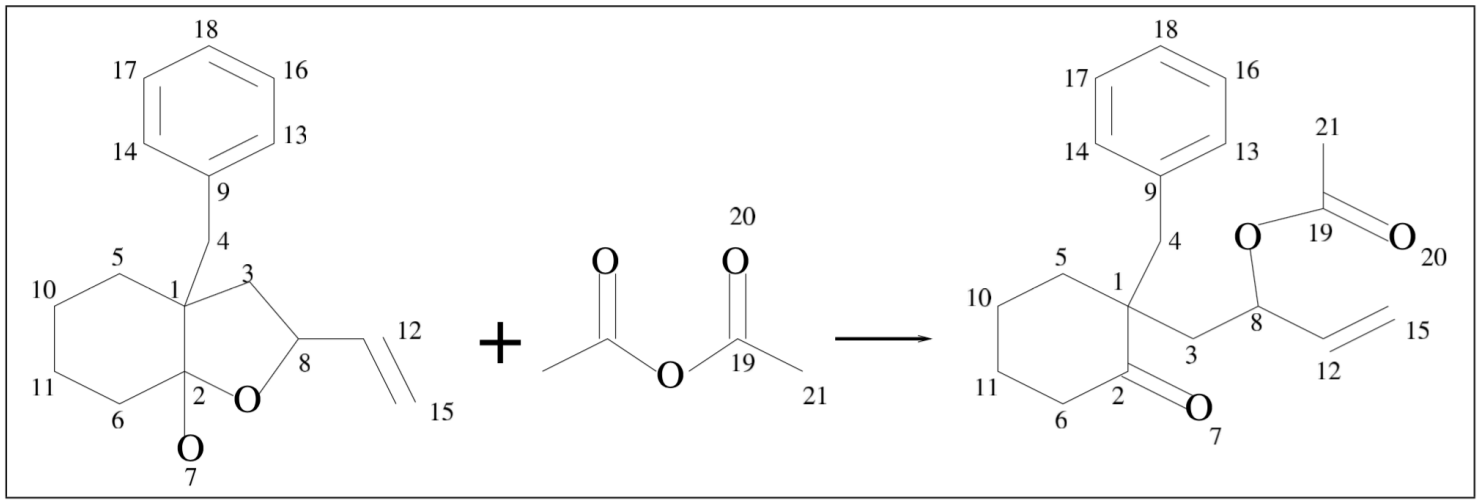
At the end of the pre-processing step, the information obtained by the recognition process is incorporated into the representation of the reaction.

Mining Chemical Reaction Database

Data on reactions have been transformed into a binary table:

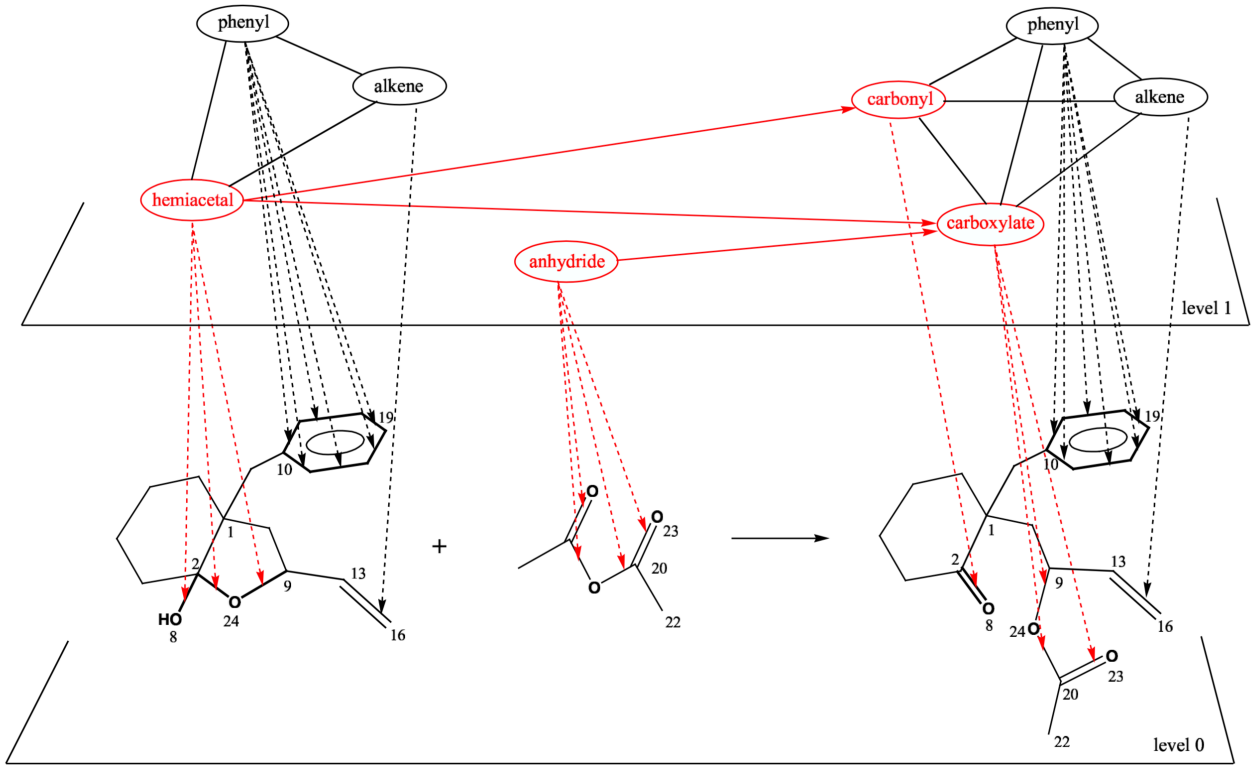
- A reaction can be considered from two main points of view:
  - a global point of view on the functionality interchanges leads to consider a single entry R corresponding to a single analyzed reaction, to which a list of properties, i.e. formed and/or destroyed and/or unchanged functions, is associated,
  - a specific point of view on the functionality transformations that is based on the consideration of two (or more) different entries R_k corresponding to the different functions being formed

Mining Chemical Reaction Database

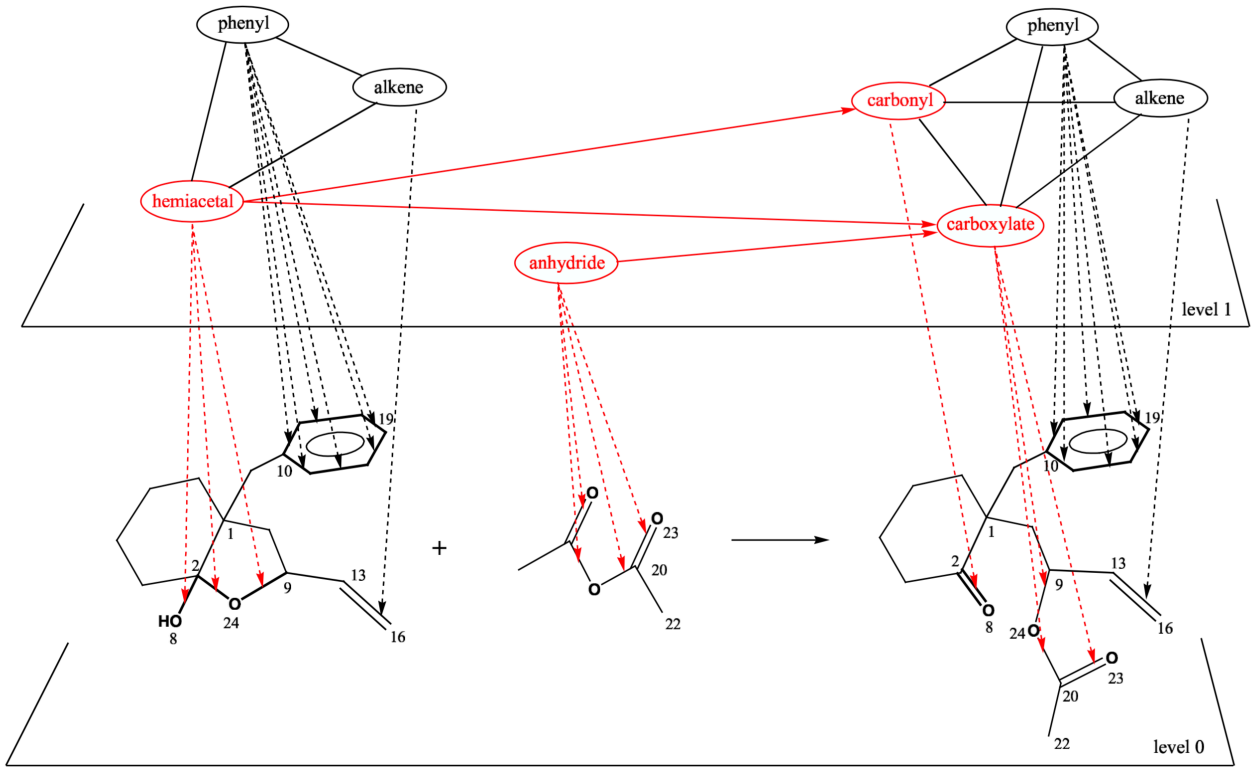


	destroyed blocks				created blocks				unchanged blocks					
functions / objects		anhydride		hemiacetal		carbonyle		carboxylate		alkene		phenyl		
T^1		X		X		X		X		X		X		

Mining Chemical Reaction Database



Mining Chemical Reaction Database



	destroyed blocks		created blocks		unchanged blocks	
functions / objects	anhydride	hemiacetal	carbonyle	carboxylate	alkene	phenyl
$T1^2$	×	×		×	×	×
$T2^2$		×	×		×	×

Entries/Blocks	**Destroyed**		**Formed**		**Unchanged**	
	anhydride	hemiacetal	carbonyle	ester	alcene	aryle
without correspondence entry **R**	x	x	x	x	x	x
with correspondence entry $\mathbf{R}_1$	x	x		x	x	x
entry $\mathbf{R}_2$		x	x		x	x

Mining Chemical Reaction Database

A 3-itemset :

$\texttt{carboxilic} - \texttt{acid}_\texttt{d} \wedge \texttt{primary} - \texttt{amine}_\texttt{d} \wedge \texttt{secondaryamine}_\texttt{f}$

- has a support of 121
- $\texttt{carboxilic} - \texttt{acid}_\texttt{d}$ and $\texttt{primary} - \texttt{amine}_\texttt{d}$ have been deleted
- $\texttt{secondaryamine}_\texttt{f}$ is formed
- Since the support of $\texttt{carboxilic} - \texttt{acid}_\texttt{d}$ and $\texttt{secondary} - \texttt{amine}_\texttt{f}$ is 154
- The rule
  $\texttt{carboxilic} - \texttt{acid}_\texttt{d} \wedge \texttt{secondary} - \texttt{amine}_\texttt{f} \longrightarrow \texttt{primary} - \texttt{amine}_\texttt{d}$
- has a support = 121 and confidence = 78.6

If "carboxilic acid is deleted and a secondary amine is formed" is true, "primary amine has been deleted" is true in 78.6% of the cases in the databases.

Mining Chemical Reaction Database

Number of itemsets and association rules extracted

		ORGSYN2000		JSM2002	
		global[1]	specific[2]	global[1]	specific[2]
Itemsets	minsup > 1	26.053	9.707	504.316	139.159
	minsup > 10	659	543	12.834	7.326
	minsup > 100	41	41	1.089	763
Association rules	minsup > 10 and confidence > 0	1.366	1.048	Nd	39.496
	minsup > 10 and confidence > 50	78	140	Nd	2.687
	minsup > 1 and confidence > 0	427.908	72.882	Nd	nd
	minsup > 1 and confidence > 50	225.800	23.801	Nd	1.326.268

Mining Chemical Reaction Database

Analysis of frequent itemsets:

- From which deleted function a formed function is created: $F_d \wedge F_f$
- Formed functions from two deleted functions: $F_f \wedge F_{1d} \wedge F_{2d}$
- Formed functions that depend on the presence of unchanged functions: $F_f \wedge F_d \wedge F_u$

Mining Chemical Reaction Database

Association rules bring further information
The more frequent way to form a molecule F_f : if function F_f is formed than is is formed from F_{id}

- F_f should be the premise of the rule: $F_f \longrightarrow \{F_{id}\}$
- Ranking rules following decreasing confidence

If you know that a function F_d is formed from two functions and you know one of them:

- F_f should be the premise of the rule: $F_f \wedge F_{1d} \longrightarrow \{F_{2d}\}$
- Ranking rules following decreasing confidence

Mining Chemical Reaction Database

Looking at Carboxylate Esters: most frequent functions involved in carboxilate creation.

$carboxilate_f \longrightarrow F_{id}$

Entry	destroyed function[1]	sup(P)[2]	conf[3]
1	alcohol	1030	21.5
2	carboxylic-acid	660	13.8
3	carboxylate	651	13.6
4	carbonyl	567	11.8
5	anhydride	419	8.7
6	alkene	334	7.0
7	ether	206	4.3
8	acetal	181	3.8
9	acyl-chloride	175	3.7
10	vinyloxycarbonyl	140	2.9

Experiments in biology

Cancer diagnosis (after discretisation of gene expression):

$$cancer \longrightarrow geneA \uparrow geneB \downarrow geneC \uparrow$$

Different association rules

A remark about association rules:

- if $\{ab\} \longrightarrow \{cd\}$ is a valid rule
- then $\{abcd\}$ is frequent, so are $\{ab\}$ {a}, {b}...
- then $\{abc\} \longrightarrow \{d\}$ and $\{abd\} \longrightarrow \{c\}$ are valid rules

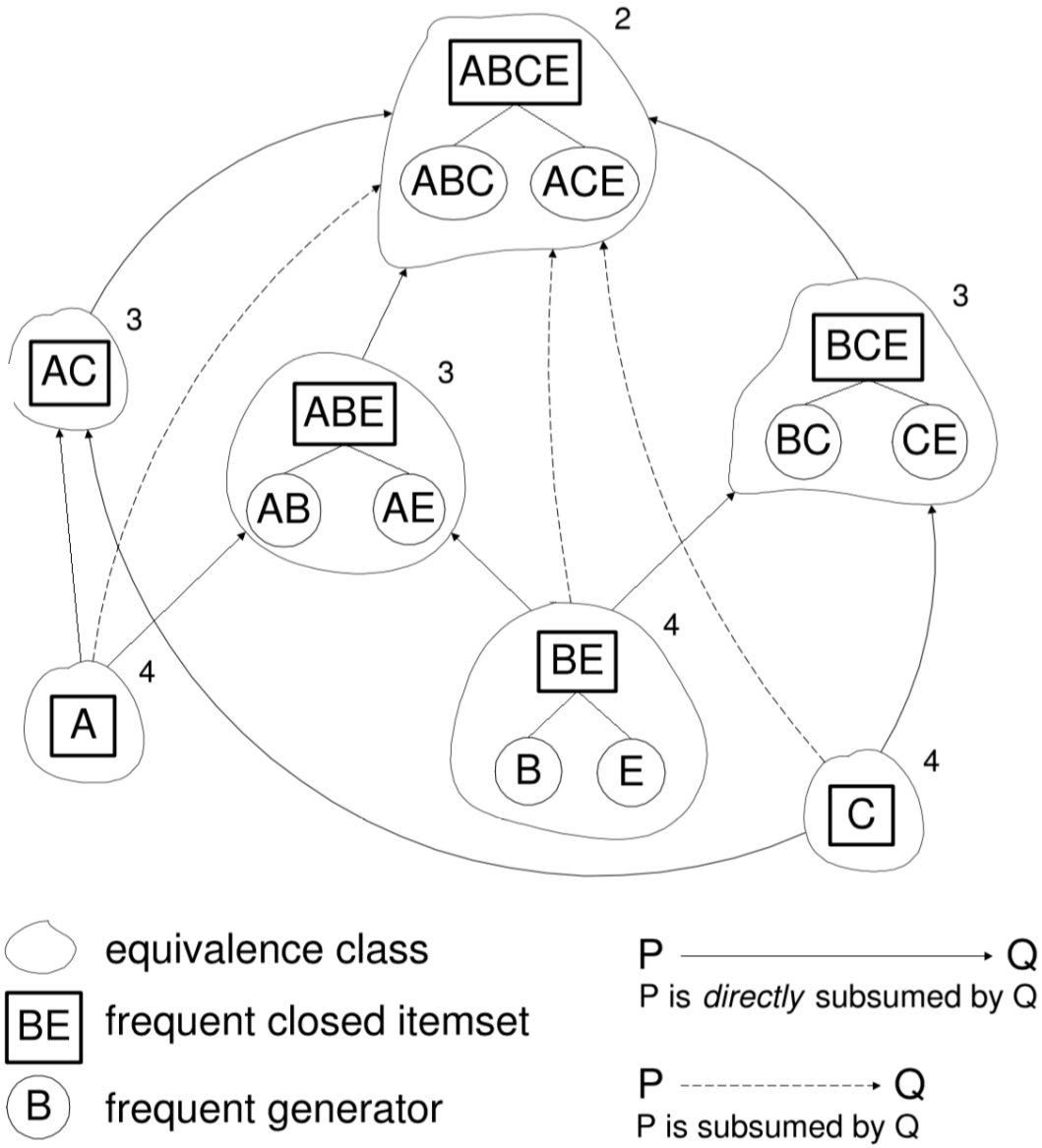
The shorter the premise (condition), the more informative the rule.

Generators and closed itemsets

- An itemset is a generator if it has no proper subset with the same support.
- An itemset is closed if it has no proper superset with the same support. The closure of an itemset X, $\gamma(X)$, is the largest superset of X with the same support.
- A frequent itemset is a Maximal frequent itemset (MFI) if all its supersets are not frequent.

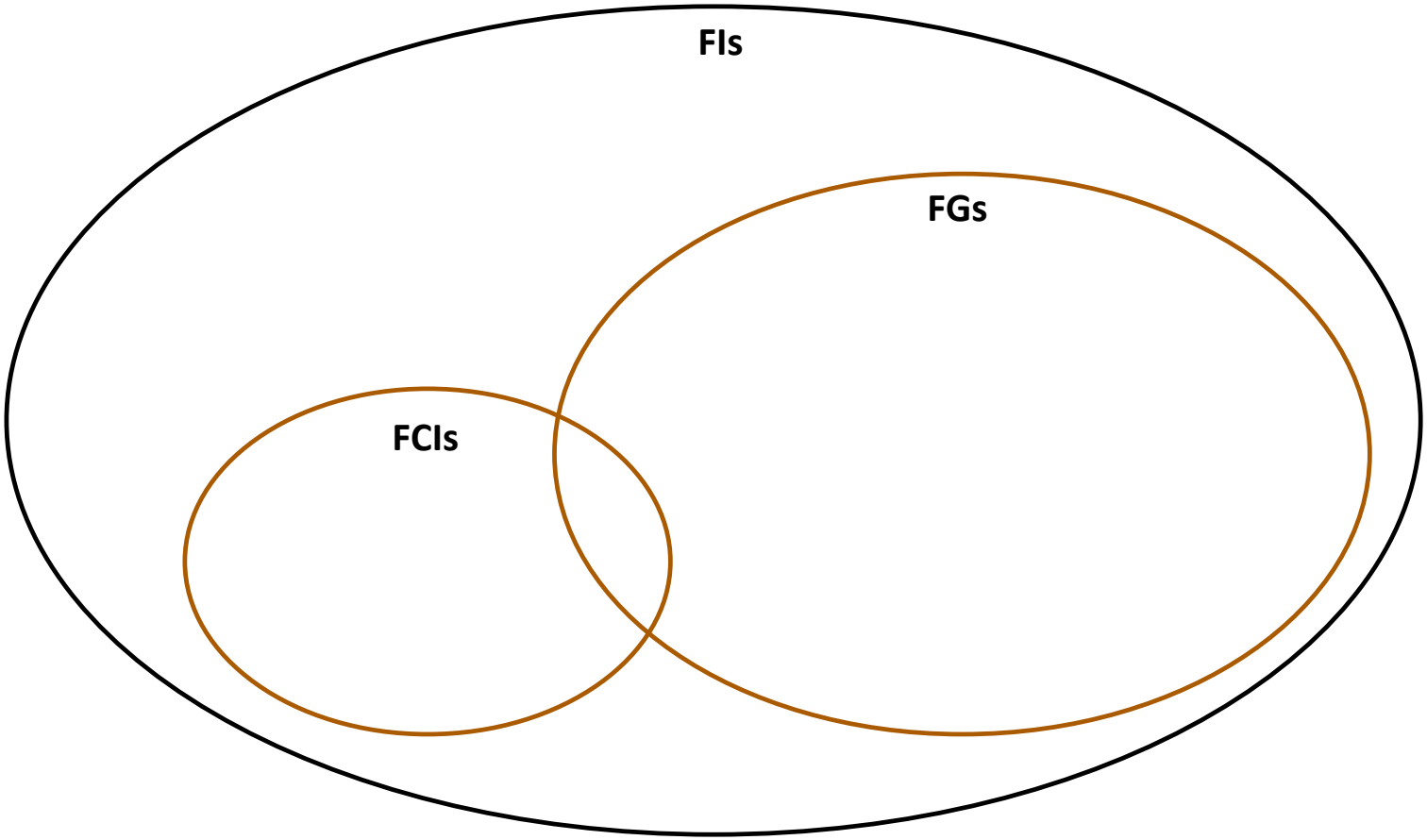
Equivalence classes

	a	b	c	d	e
o1	x	x		x	x
o2	x		x		
o3	x	x	x		x
o4		x	x		x
o5	x	x	x		x



Equivalence classes

Frequent itemsets, closed frequent itemsets and frequent generators:



Algorithms

Different kind of algorithms for mining itemsets

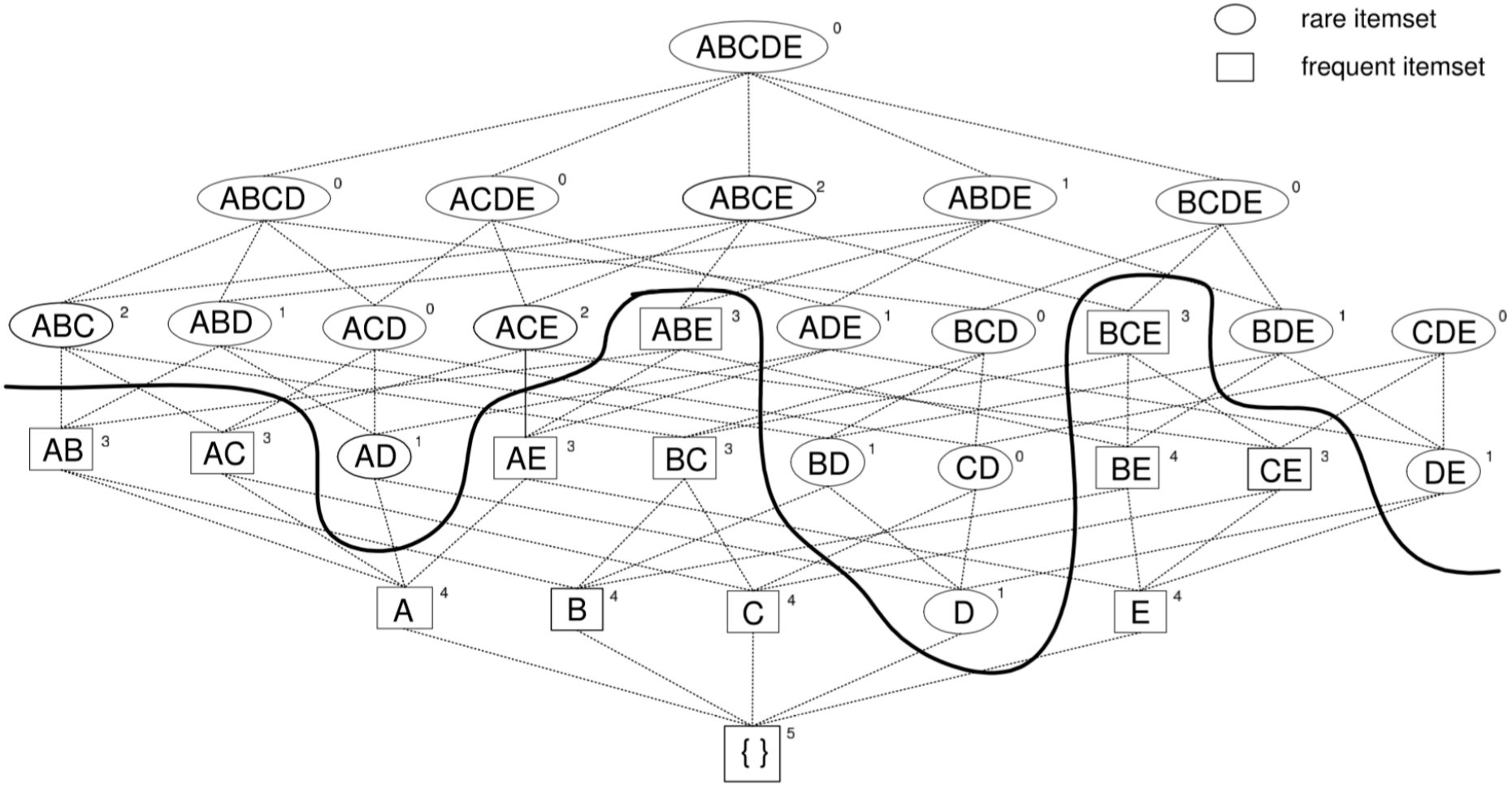
- Levelwise algorithms
- Vertical algorithms
- Hybrid algorithms (Eclat-Z,Charm-MFI)
- Other algorithms (FP-growth)

Levelwise algorithms

At each level, are itemset of the same length.

The most well-known algorithm: Apriori
Property 1 (downward closure): All subsets of a frequent itemset are frequent.
Property 2 (anti-monotonocity): All supersets of an infrequent itemset are infrequent.
Examples: Apriori-Close, Close, Titanic, Pascal, Pascal+, Zart, etc.

Levelwise algorithms



Levelwise algorithms

- Apriori was the first efficient algorithm for finding FIs. It is a breadth-first, bottom up levelwise algorithm
- Two tables : C_i candidate itemsets, F_i frequent itemsets.
- i-itemsets are combined to produce $i + 1$-candidtae itemsets

Levelwise algorithms

- Two tables : C_i candidate itemsets, F_i frequent itemsets.

Apriori

Running example

	A	B	C	D	E
1	x		x	x	
2		x	x		x
3	x	x	x		x
4		x			x
5	x	x	x		x

min_supp = 3

C_1	supp
{A}	3
{B}	4
{C}	4
{D}	1
{E}	4

F_1	supp
{A}	3
{B}	4
{C}	4
{E}	4

Levelwise algorithms

- With one database pass, the support of candidate itemsets is counted and infrequent itemsets are pruned.
- Possibility to test close itemsets and generators

C_2	supp
{AB}	2
{AC}	3
{AE}	2
{BC}	3
{BE}	4
{CE}	3

F_2	supp
{AC}	3
{BC}	3
{BE}	4
{CE}	3

C_3	supp
{BCE}	3

F_3	supp
{BCE}	3

C_4	supp

STOP!

FIs: U F_i

20

Levelwise algorithms

The candidate generation and the support counting require a subset test.

- To generate C_i we run over F_i
- In the support counting process, the database is read object per object
- We need to find the subsets of the corresponding itemset for each object in C_k and the support value of each subset in C_k is incremented (by 1)
- require the use of hash-tree or prefix-tree for the data structure

Vertical Algorithms

Algorithms that process the database vertically (deep-first algorithms)

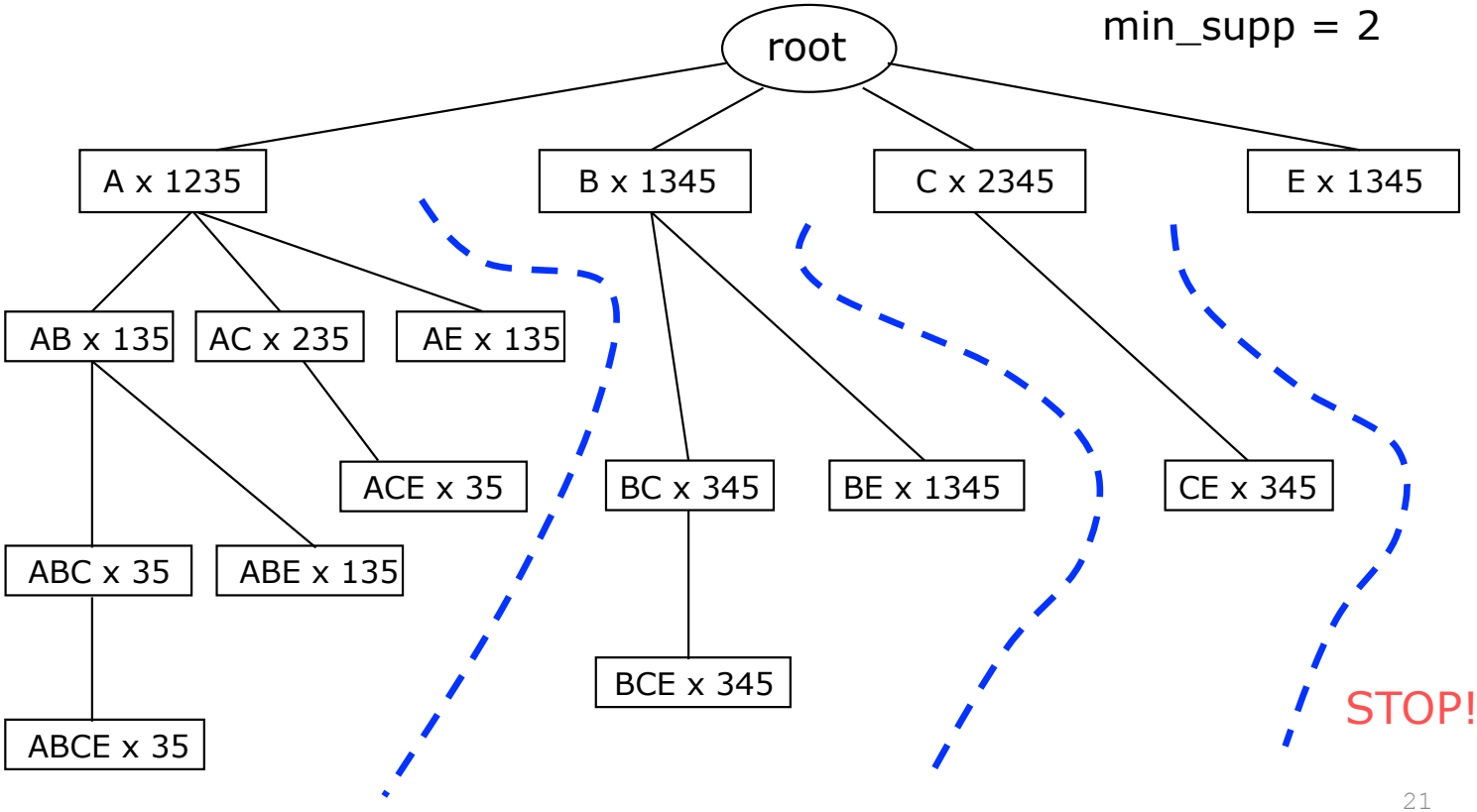
- Eclat
- Charm

Eclat [Zaki, 97]

Running example

	A	B	C	D	E
1	x	x		x	x
2	x		x		
3	x	x	x		x
4		x	x		x
5	x	x	x		x

min_supp = 2



Sets of rules

The goal is to find interesting association rules

- All valid association rules $\longrightarrow$ too many rules, many of them are redundant
- Different concise representations (bases) : Closed Rules ($\mathcal{CR}$), Generic basis ($\mathcal{GB}$), informative basis ($\mathcal{IB}$), minimal non-redundant association rules ($\mathcal{MNR}$)...

A very good comparative study of these bases can be found in [Kry02]. A "good" representation of association rules:

- should enable the derivation of all valid rules
- should forbid the derivation of rules that are not valid
- should allow the determination of rules parameters such as support and confidence

Sets of rules

Generic Basis (for exact association rules): Let FC be the set of frequent closed itemsets. For each frequent closed itemset f, let FG_f denote the set of frequent generators of f. The generic basis:

$$\mathcal{GB} = \{r : g \longrightarrow f \setminus g | f \in FC \wedge g \in FG_f \wedge g \neq f\}$$

Informative basis for approximate association rules: Let FC be the set of frequent closed itemsets and let FG denote the set of frequent generators. The notation $\gamma(g)$ means the closure of itemset g. The informative basis:

$$(IB) = \{r : g \longrightarrow f \setminus g | f \in FC \wedge g \in FG \wedge \gamma(g) \subset f\}$$