Universiteit Utrecht

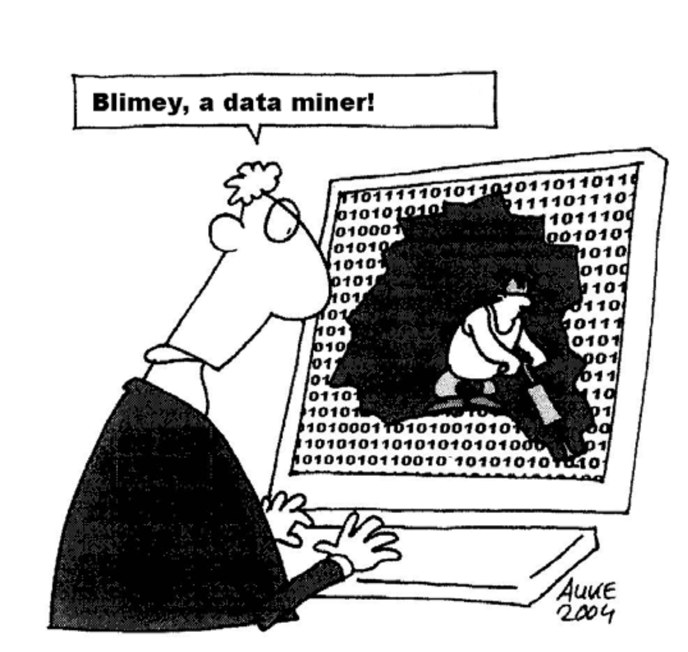
[Faculteit Bètawetenschappen
Informatica]

DAR
Frequent Item Sets

Ad Feelders

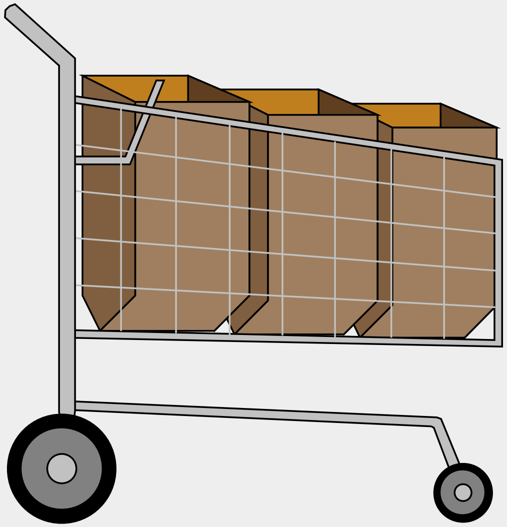
What is Data Mining?

Discovery of interesting patterns and models in data bases.



Association rules

Find groups of products that are often bought together.



Universiteit Utrecht

[Faculteit Bètawetenschappen
Informatica]

Frequent item set mining

- Item set table
- Transactions (baskets) t_k and items i_j
- We are interested in association rules $X \rightarrow Y$
- "If clients buy X, then they will also buy Y"

tid	i1	i2	i3	...	i_m
t1	1	1	0	...	1
t2	0	1	0	...	
t3	1	0	1	...	0
...				...	
t_n	1	0	0	...	1

Frequent item set mining

tid	i1	i2	i3	i4	i5
t1	1	1	0	1	1
t2	0	1	0	1	1
t3	1	1	1	1	0
t4	1	1	0	0	0
t5	1	0	0	1	1

Let X = { i1, i2 }

Let Y = { i4 }

Support (X) = 3

Support (XY) = 2

Confidence for X $\rightarrow$ Y is 2/3

Support for X $\rightarrow$ Y is Support (XY) = 2

Frequent item set mining to find association rules

- Table r(U) with $U=\{i_1,\ldots,i_m\}$, i_j is a binary attribute (item).
- For $X,Y \subseteq U$, with $X \cap Y = \emptyset$, let:
  - s(X) denote the support of X, i.e. the number of tuples that have the value 1 for all items in X.
  - for an association rule $X \rightarrow Y$, define
    - the support is s(XY)
    - the confidence is s(XY)/s(X)
- Problem: find all association rules with support $\geq t_1$ and confidence $\geq t_2$.

Algorithm Sketch

There are two thresholds we have to satisfy:

1. Find all sets Z whose support exceeds the minimum support threshold. These sets are called frequent.
2. Test for all non-empty subsets X of Z whether the rule X→Y (where Y = Z-X) holds with sufficient confidence.

Find all frequent item sets

- An item set is *frequent* if its support is bigger than a user- specified minimum support threshold.
- Naive method: make a list off *all* item sets and for each item set count in how many transactions it occurs.
- For a collection of just 100 products there are 2^{100} different item sets. If we could count 1 million item sets per second we would be busy for (roughly) 4×10^{15} years.

The Apriori property

- If X is frequent, then all its subsets are also frequent.
- If X has a subset that is not frequent, then it cannot be frequent.
- This suggest a level wise search for frequent item sets, where the level is the number of items in the set:
  - A set is a candidate frequent set if all its subsets are frequent.

Find all frequent item sets

Apriori algorithm:

1. C_1 := all 1-itemsets;
2. F:= $\emptyset$; i :=1;
3. **while** $C_i \neq \emptyset$ **repeat**
4. F_i := item sets in C_i that are frequent;
5. add F_i to F;
6. C_{i+1} := item sets of size i+1 for which all subsets of size i are frequent.
7. i := i+1;
8. Return F as the result.

Universiteit Utrecht

[Faculteit Bètawetenschappen
Informatica]

Apriori: Example

tid	Items
1	ABE
2	BD
3	BC
4	ABD
5	AC
6	BC
7	AC
8	ABCE
9	ABC

Cand.	Support	Frequent?
A	6	✓
B	7	✓
C	6	✓
D	2	✓
E	2	✓

Minimum support = 2

All items ABCDE are level 1 frequent item sets

Apriori: Example

tid	Items
1	ABE
2	BD
3	BC
4	ABD
5	AC
6	BC
7	AC
8	ABCE
9	ABC

Cand.	Support	Frequent?
A	6	✓
B	7	✓
C	6	✓
D	2	✓
E	2	✓

To generate level 2 candidates, we combine all level 1 frequent item sets. For example A+B = AB.

Example: Level 2

tid	Items
1	ABE
2	BD
3	BC
4	ABD
5	AC
6	BC
7	AC
8	ABCE
9	ABC

Cand.	Support	Frequent?
AB	4	✔
AC	4	✔
AD	1	✘
AE	2	✔
BC	4	✔
BD	2	✔
BE	2	✔
CD	0	✘
CE	1	✘
DE	0	✘

Example: Level 2

Cand.	Support	Frequent?
AB	4	✓
AC	4	✓
AD	1	✗
AE	2	✓
BC	4	✓
BD	2	✓
BE	2	✓
CD	0	✗
CE	1	✗
DE	0	✗

To generate level 3 candidates we combine frequent level 2 item sets that have the first item in common.

If a candidate has a subset that is not frequent, it is pruned.

AB+AC = ABC
Since BC is also frequent, it is not pruned.
BC+BD = BCD
It is pruned because CD is not frequent.

Example: Level 3

tid	Items
1	ABE
2	BD
3	BC
4	ABD
5	AC
6	BC
7	AC
8	ABCE
9	ABC

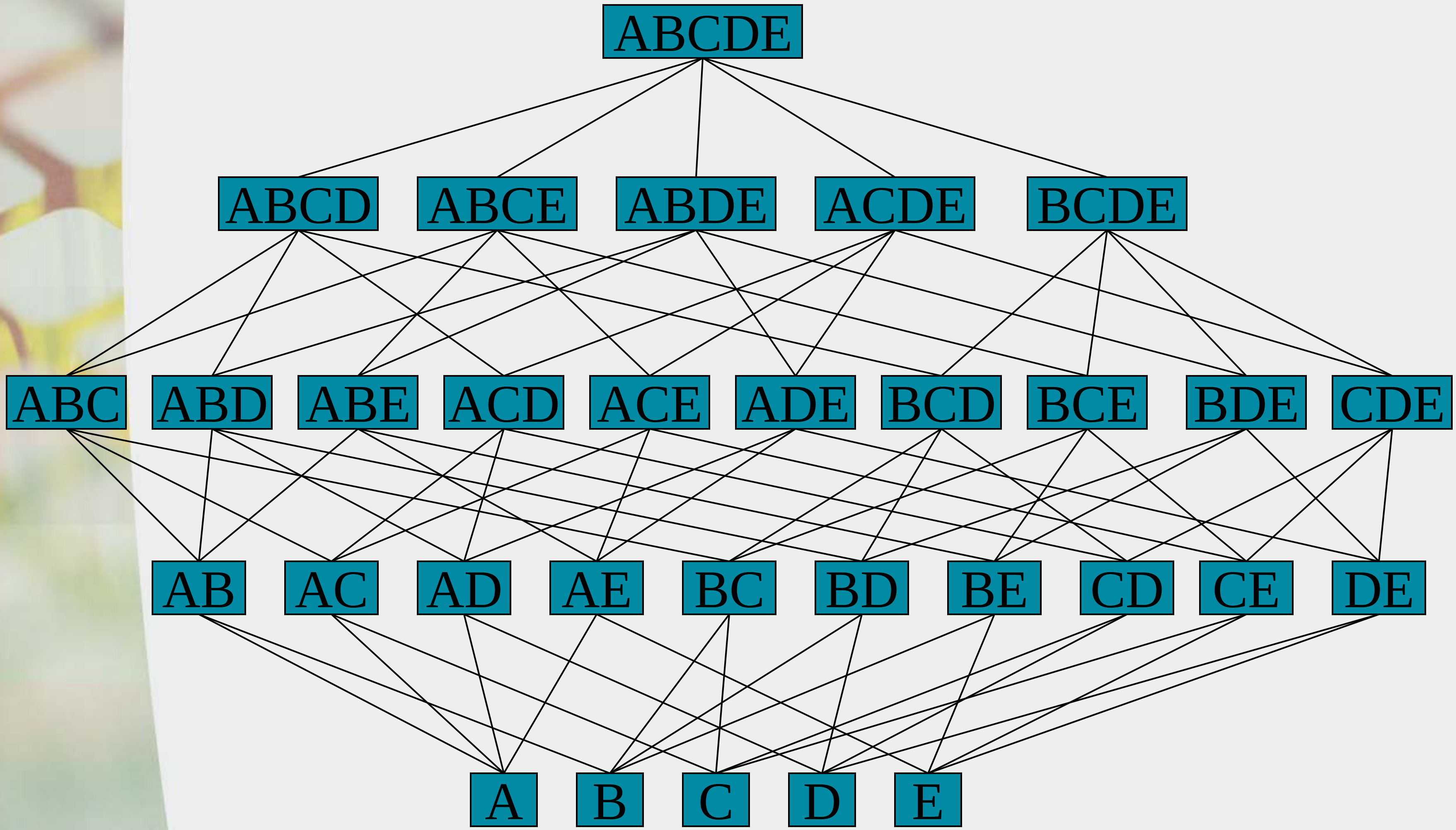
Cand.	Support	Frequent?
ABC	2	✅
ABE	2	✅

To generate level 4 candidates we combine frequent level 3 item sets that have the first 2 items in common.

ABC+ABE = ABCE

This candidate is pruned because ACE is not frequent.

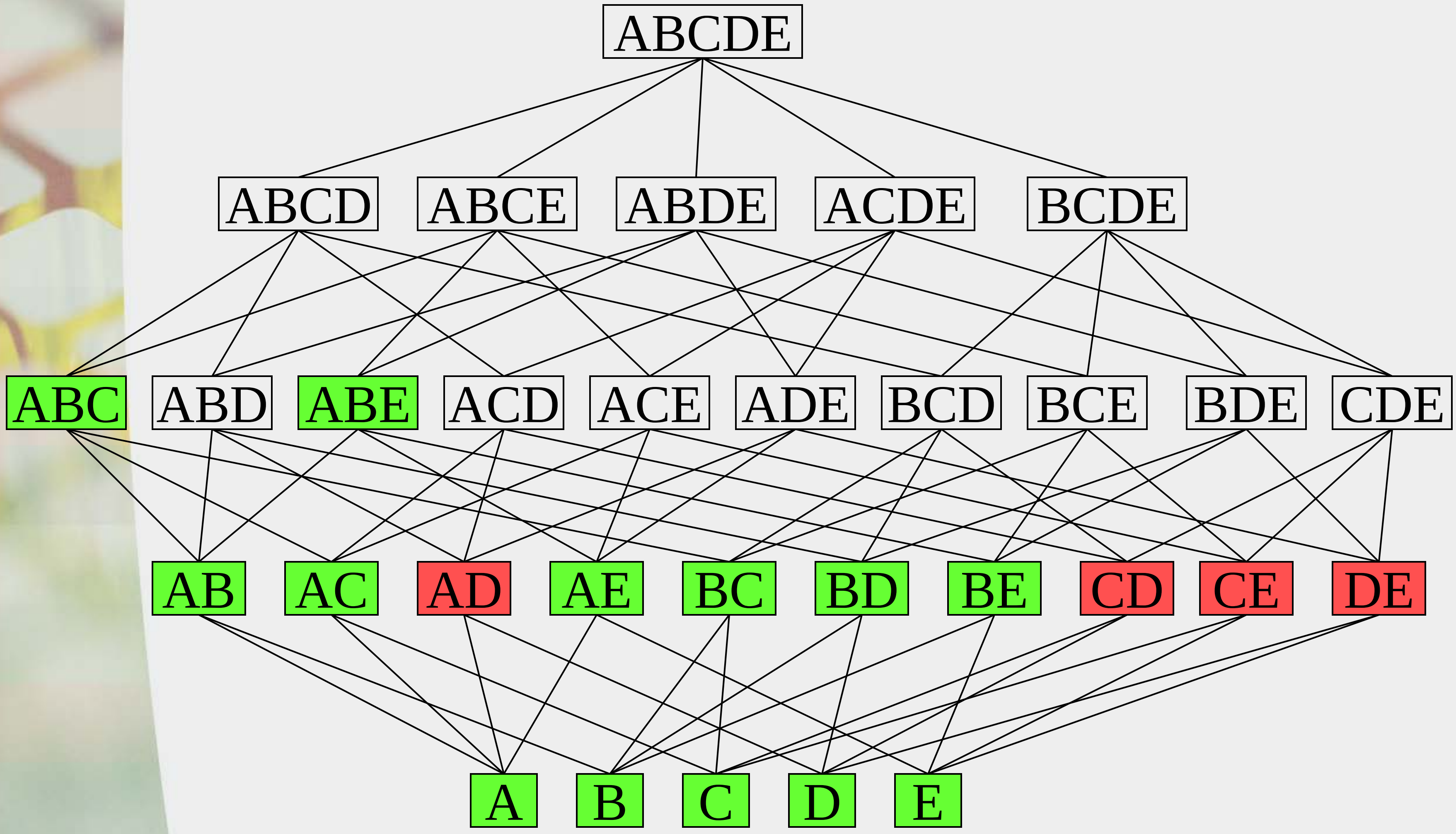
The Search Space



Universiteit Utrecht

[Faculteit Bètawetenschappen
Informatica]

Item sets counted by Apriori



Complexity of level wise search

- Recall: m is total number of items
- We rejected the naïve algorithm because its complexity was $O(2^m)$
- So what is the complexity of level wise search?
- Worst case is still $O(2^m)$. When does that occur?
- If r(U) is sparse (by far, most values are 0), then we expect that the frequent sets have maximal size k with k much smaller than m.
- In that case we have a worst-case complexity of

$$O\left(\sum_{j=1}^{k} \binom{m}{j}\right) = O(m^k) << O(2^m)$$

Association Rules

One frequent item set may produce many rules.
ABE generates:

Left side	Rule	Confidence
AB	$AB \rightarrow E$	2/4 = 50%
AE	$AE \rightarrow B$	2/2 = 100%
BE	$BE \rightarrow A$	2/2 = 100%
A	$A \rightarrow BE$	2/6 = 33%
B	$B \rightarrow AE$	2/7 = 29%
E	$E \rightarrow AB$	2/2 = 100%

$$\text{Confidence}(AB \rightarrow E) = s(ABE)/s(AB) = 2/4$$

Diapers and Beer

Diapers ⇒ Beer