Reducing the collection of itemsets: alternative representations and combinatorial problems

Too many frequent itemsets

- If $\{a_1, \ldots, a_{100}\}$ is a frequent itemset, then there are

$$\binom{100}{1} + \binom{100}{2} + \ldots + \binom{100}{100} = 2^{100} - 1$$

1.27*10^{30} frequent sub-patterns!

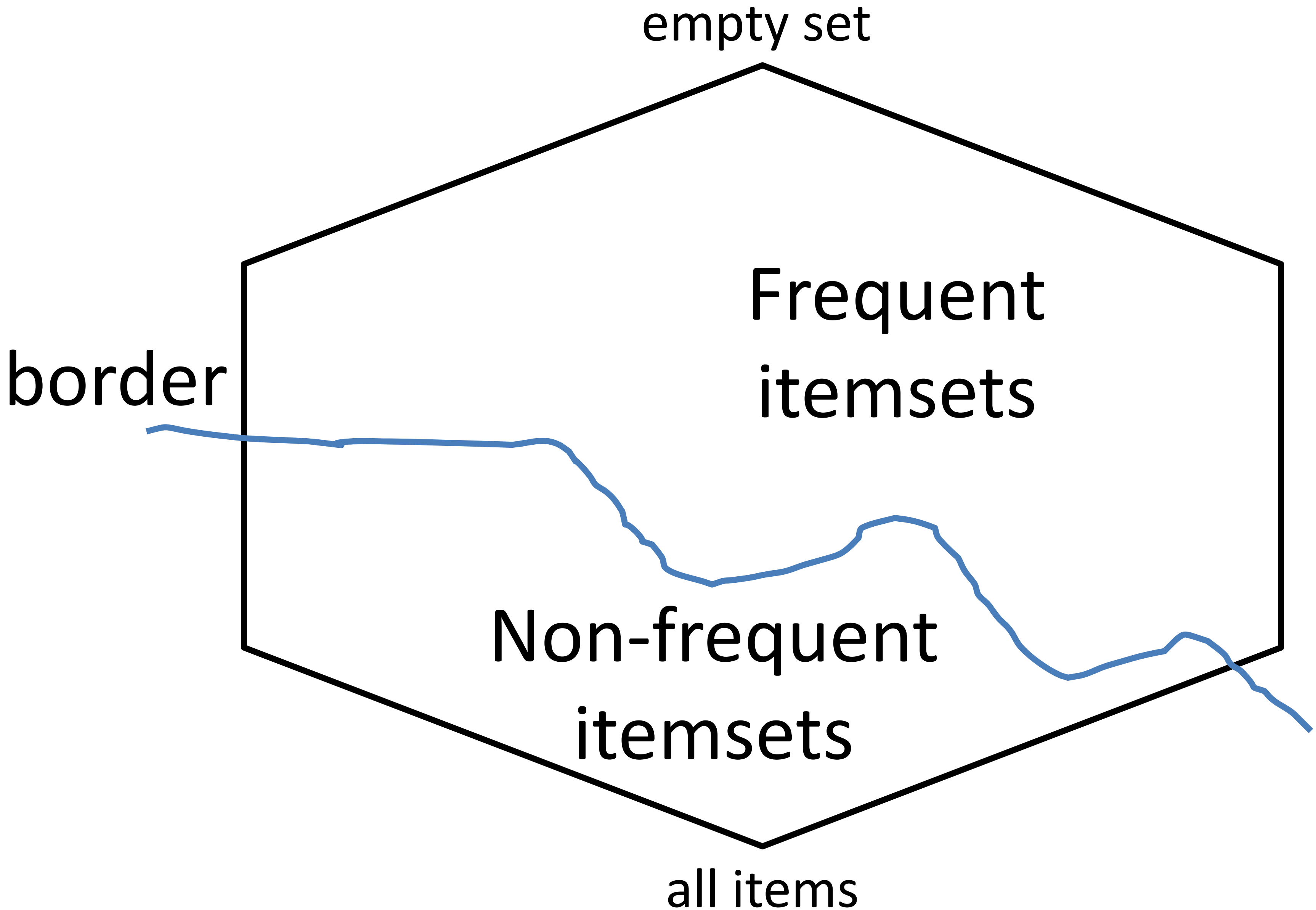
- There should be some more **condensed** way to describe the data

Frequent itemsets maybe too many to be helpful

- If there are many and large frequent itemsets enumerating all of them is costly.

- We may be interested in finding the ***boundary*** frequent patterns.

- **Question:** Is there a good definition of such boundary?


empty set
Frequent itemsets
border
Non-frequent itemsets
all items

Borders of frequent itemsets

- Itemset **X** is more ***specific*** than itemset **Y** if **X superset of Y** (notation: **Y<X**). Also, **Y** is more ***general*** than **X** (notation: **X>Y**)

- **The Border:** Let **S** be a collection of frequent itemsets and **P** the lattice of itemsets. The ***border*** **Bd(S)** of **S** consists of all itemsets **X** such that ***all more general itemsets*** than **X** are in **S** and ***no pattern more specific*** than **X** is in **S**.

$$Bd(S) = \left\{ X \in P \,\middle|\, \begin{array}{l} \text{for all } Y \in P \text{ with } Y \prec X \text{ then } Y \in S, \\ \text{and for all } W \in P \text{ with } X \prec W \text{ then } W \notin S \end{array} \right\}$$

Positive and negative border

- **Border**

$$Bd(S) = \left\{ X \in P \,\middle|\, \begin{array}{l} \text{for all } Y \in P \text{ with } Y \prec X \text{ then } Y \in S, \\ \text{and for all } W \in P \text{ with } X \prec W \text{ then } W \notin S \end{array} \right\}$$

- **Positive border:** Itemsets in the border that are also frequent (belong in **S**)

$$Bd^{+}(S) = \left\{ X \in S \,\middle|\, \text{for all } Y \in P \text{ with } X \prec Y \text{ then } Y \notin S \right\}$$

- **Negative border:** Itemsets in the border that are not frequent (do not belong in **S**)

$$Bd^{-}(S) = \left\{ X \in P \setminus S \,\middle|\, \text{for all } Y \in P \text{ with } Y \prec X \text{ then } Y \in S \right\}$$

Examples with borders

- Consider a set of items from the alphabet: **{A,B,C,D,E}** and the collection of frequent sets

$$S = \{\{A\},\{B\},\{C\},\{E\},\{A,B\},\{A,C\},\{A,E\},\{C,E\},\{A,C,E\}\}$$

- The negative border of collection **S** is

$$Bd^{-}(S) = \{\{D\},\{B,C\},\{B,E\}\}$$

- The positive border of collection **S** is

$$Bd^{+}(S) = \{\{A,B\},\{A,C,E\}\}$$

Descriptive power of the borders

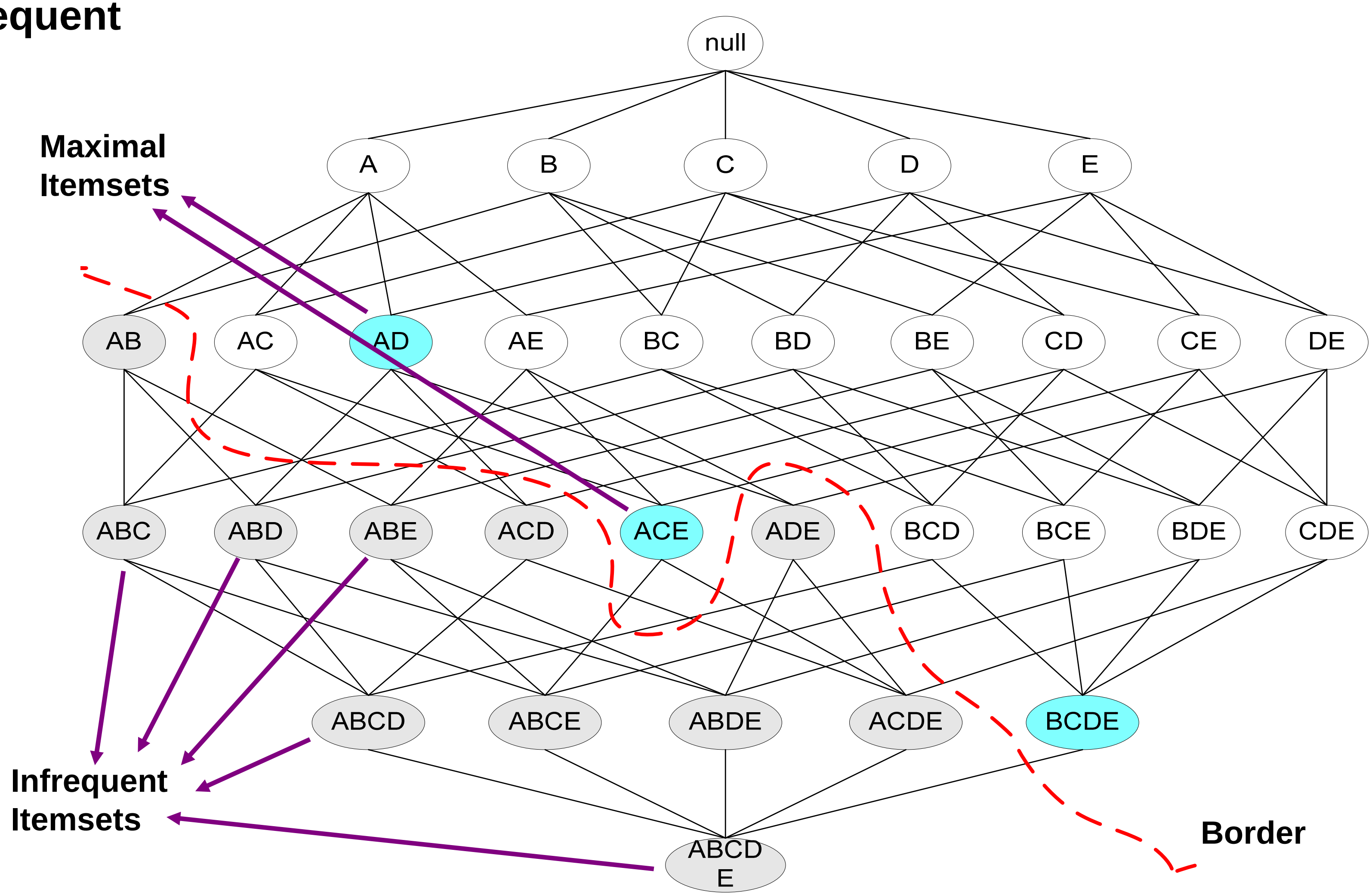
- **Claim:** A collection of frequent sets **S** can be ***fully described*** using only the positive border (**Bd⁺(S)**) or only the negative border (**Bd⁻(S)**).

Maximal patterns

Frequent patterns without proper frequent super pattern

Maximal Frequent Itemset

An itemset is maximal frequent if none of its immediate supersets is frequent



Maximal patterns

- The set of maximal patterns is the same as the positive border

- Descriptive power of maximal patterns:
  – Knowing the set of all maximal patterns allows us to reconstruct the set of all frequent itemsets!!

  – We can only reconstruct the set not the actual frequencies

Closed patterns

- An itemset is closed if none of its immediate supersets has the same support as the itemset

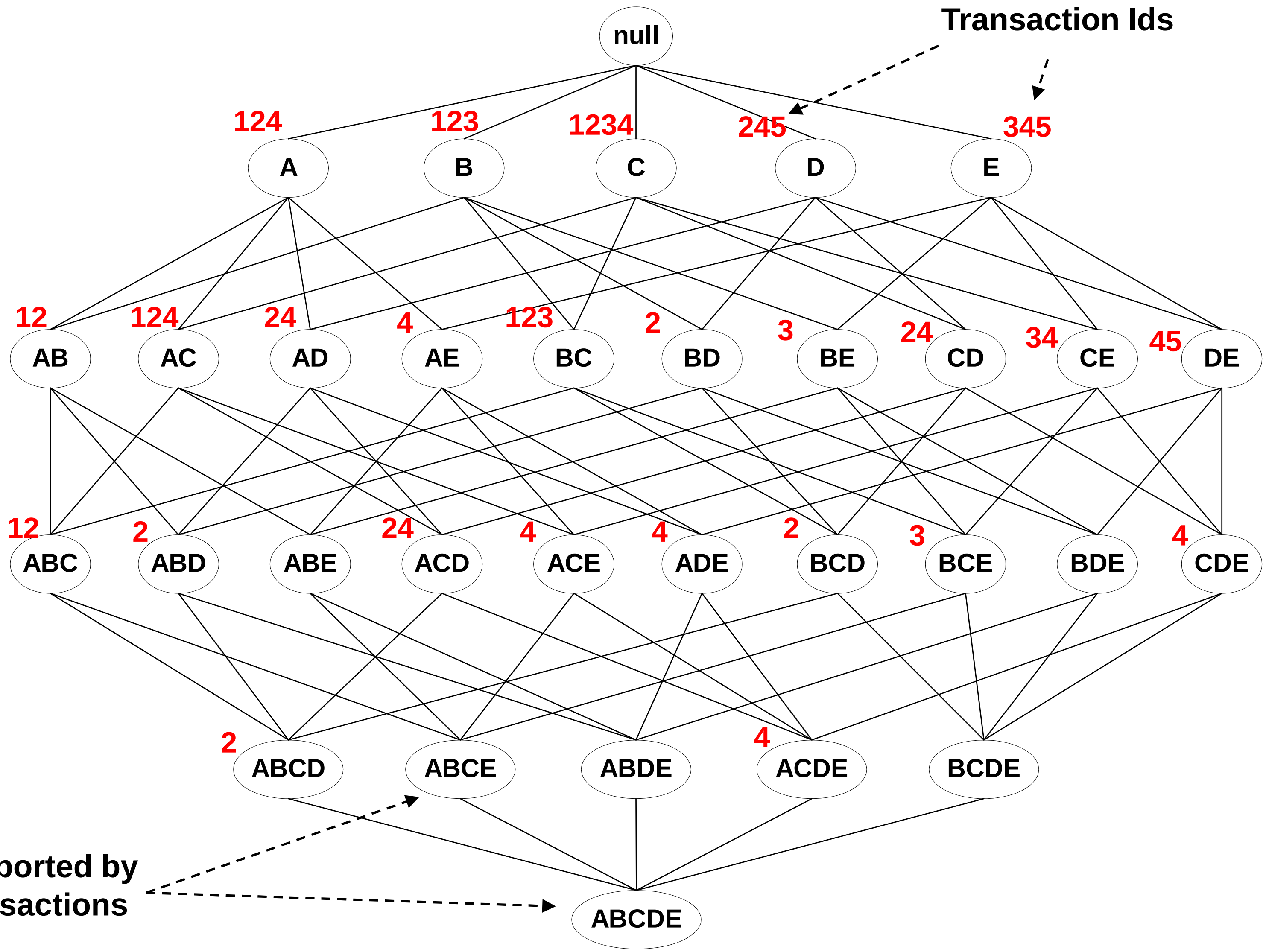
TID	Items
1	{A,B}
2	{B,C,D}
3	{A,B,C,D}
4	{A,B,D}
5	{A,B,C,D}

Itemset	Support
{A}	4
{B}	5
{C}	3
{D}	4
{A,B}	4
{A,C}	2
{A,D}	3
{B,C}	3
{B,D}	4
{C,D}	3

Itemset	Support
{A,B,C}	2
{A,B,D}	3
{A,C,D}	2
{B,C,D}	3
{A,B,C,D}	2

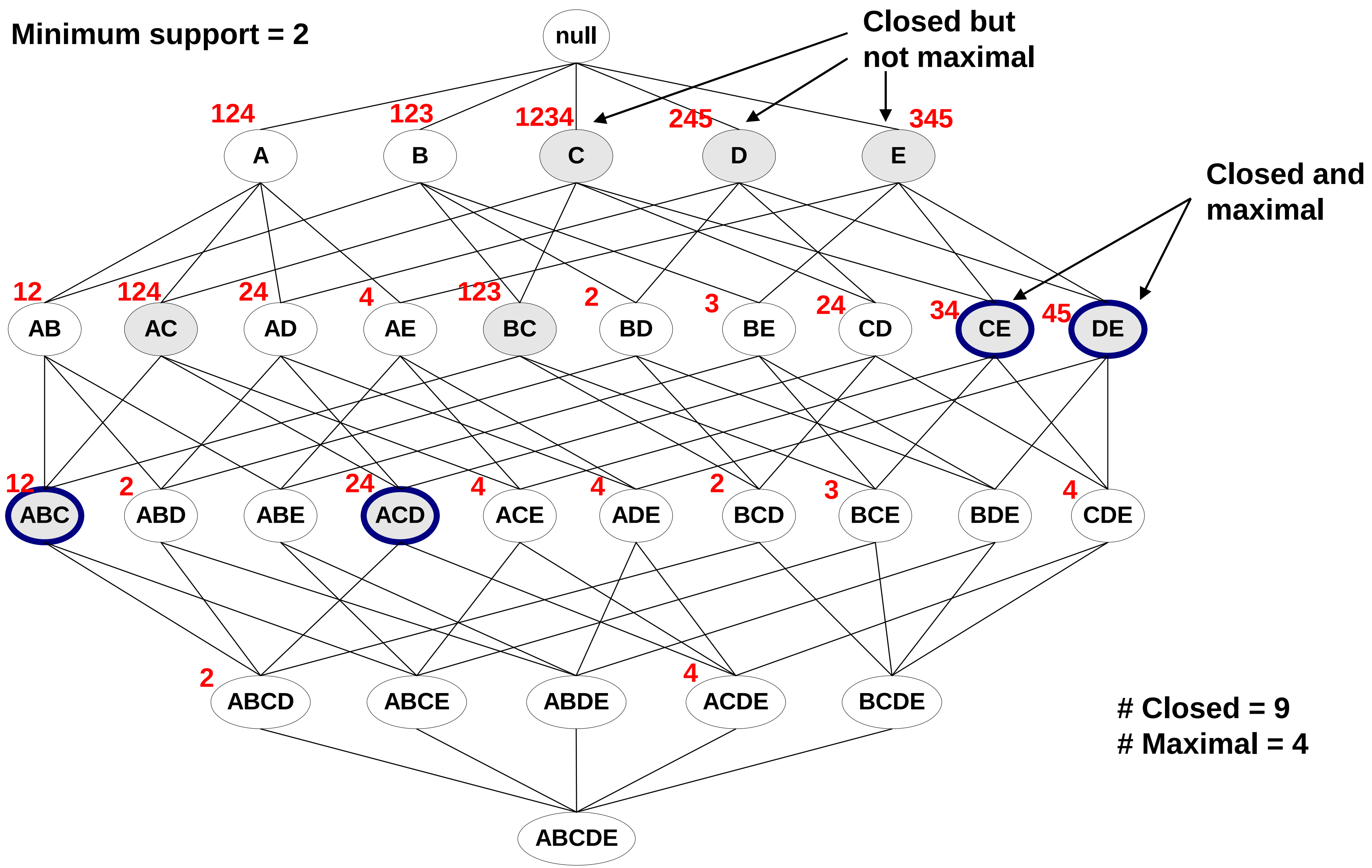
Maximal vs Closed Itemsets

TID	Items
1	ABC
2	ABCD
3	BCE
4	ACDE
5	DE



Not supported by any transactions

Maximal vs Closed Frequent Itemsets



Why are closed patterns interesting?

- **$s(\{A,B\}) = s(A)$,** i.e., **$conf(\{A\} \rightarrow \{B\}) = 1$**
- We can infer that for every itemset **X ,**
  $s(A \cup \{X\}) = s(\{A,B\} \cup X)$
- **No need to count the frequencies of sets $X \cup \{A,B\}$ from the database!**
- If there are lots of rules with confidence **1**, then a significant amount of work can be saved
  - Very useful if there are strong correlations between the items and when the transactions in the database are similar

Why closed patterns are interesting?

- Closed patterns and their frequencies alone are sufficient representation for all the frequencies of all frequent patterns

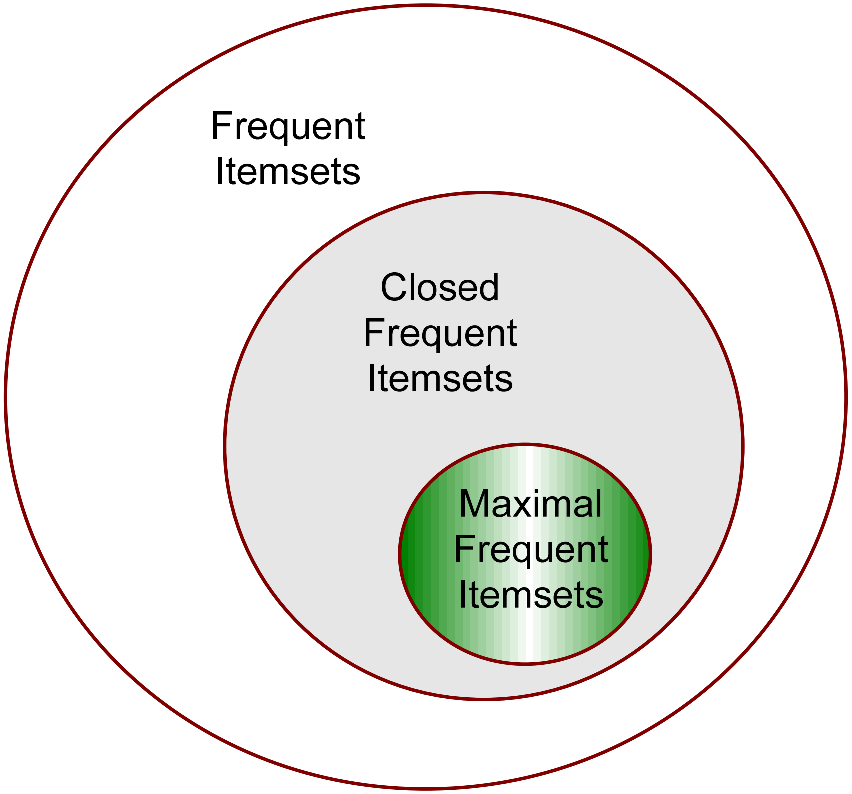
- **Proof:** Assume a frequent itemset **X**:
  - **X** is closed → **s(X)** is known
  - **X** is not closed →

    s(X) = max {s(Y) | Y is closed and X subset of Y}

Maximal vs Closed sets

- Knowing all maximal patterns (and their frequencies) allows us to reconstruct the set of frequent patterns

- Knowing all closed patterns and their frequencies allows us to reconstruct the set of all frequent patterns and their frequencies



A more algorithmic approach to reducing the collection of frequent itemsets

Prototype problems: Covering problems

- Setting:
  - Universe of N elements $U = \{U_1,\ldots,U_N\}$
  - A set of n sets $S = \{s_1,\ldots,s_n\}$
  - Find a collection C of sets in S (C subset of S) such that $\cup_{c \in C} c$ contains many elements from U
- Example:
  - U: set of documents in a collection
  - s_i: set of documents that contain term t_i
  - Find a collection of terms that cover most of the documents

Prototype covering problems

- **Set cover problem:** Find a small collection **C** of sets from **S** such that *all elements in the universe* **U** are covered by some set in **C**
- **Best collection problem:** find a collection **C** of **k** sets from **S** such that the collection covers as many elements from the universe **U** as possible
- Both problems are NP-hard
- Simple approximation algorithms with provable properties are available and very useful in practice

Set-cover problem

- Universe of $\mathbf{N}$ elements $\mathbf{U = \{U_1, \ldots, U_N\}}$
- A set of $\mathbf{n}$ sets $\mathbf{S = \{s_1, \ldots, s_n\}}$ such that $\mathbf{\cup_i s_i = U}$

- **Question:** Find the smallest number of sets from $\mathbf{S}$ to form collection $\mathbf{C}$ ($\mathbf{C}$ subset of $\mathbf{S}$) such that $\mathbf{\cup_{c \in C} c = U}$

- The set-cover problem is **NP-hard** (what does this mean?)

Trivial algorithm

- Try all subcollections of **S**
- Select the smallest one that covers all the elements in **U**
- The running time of the trivial algorithm is $O(2^{|S|}|U|)$
- This is way too slow

Greedy algorithm for set cover

- Select first the largest-cardinality set **s** from **S**
- Remove the elements from **s** from **U**
- Recompute the sizes of the remaining sets in **S**
- Go back to the first step

As an algorithm

- **X** = **U**
- **C** = {}
- **while** **X** is not empty **do**
  - For all **s**$\in$**S** let a_s=|s intersection X|
  - Let **s** be such that a_s is *maximal*
  - **C** = **C U {s}**
  - **X** = **X\ s**

How can this go wrong?

- No global consideration of how good or bad a selected set is going to be

How good is the greedy algorithm?

- Consider a minimization problem
  - In our case we want to minimize the ***cardinality*** of set **C**
- Consider an instance **I**, and cost **$a^*(I)$** of the optimal solution
  - **$a^*(I)$**: is the minimum number of sets in **C** that cover all elements in **U**
- Let **a(I)** be the cost of the approximate solution
  - **a(I)**: is the number of sets in **C** that are picked by the greedy algorithm
- An algorithm for a minimization problem has approximation factor **F** if for all instances **I** we have that

$$a(I) \leq F \times a^*(I)$$

- ***Can we prove any approximation bounds for the greedy algorithm for set cover ?***

How good is the greedy algorithm for set cover?

- ***(Trivial?) Observation***: The greedy algorithm for set cover has approximation factor **$b = |s_{max}|$**, where **s_{max}** is the set in **S** with the largest cardinality
- **Proof:**
  - **$a^*(I) \geq N/|s_{max}|$ or $N \leq |s_{max}|a^*(I)$**
  - **$a(I) \leq N \leq |s_{max}|a^*(I)$**

How good is the greedy algorithm for set cover? A tighter bound

- The greedy algorithm for set cover has approximation factor **$F = O(\log |s_{max}|)$**
- **Proof**: (From CLR "Introduction to Algorithms")

Best-collection problem

- Universe of **N** elements $\mathbf{U = \{U_1,\ldots,U_N\}}$
- A set of **n** sets $\mathbf{S = \{s_1,\ldots,s_n\}}$ such that $\mathbf{\cup_i s_i = U}$

- **Question:** Find the a collection **C** consisting of **k** sets from **S** such that $\mathbf{f(C) = |\cup_{c \in C} c|}$ is ***maximized***

- The best-colection problem is NP-hard

- Simple approximation algorithm has approximation factor $\mathbf{F = (e-1)/e}$

Greedy approximation algorithm for the best-collection problem

- $C = \{\}$
- **for every** set s in S and ***not*** in C compute the gain of s:

$$g(s) = f(C \cup \{s\}) - f(C)$$

- Select the set s with the ***maximum*** gain
- $C = C \cup \{s\}$
- **Repeat until** C has k elements

Basic theorem

- The ***greedy*** algorithm for the best-collection problem has approximation factor **$F = (e-1)/e$**

- **C^* : optimal** collection of cardinality **k**
- **C :** collection output by the ***greedy*** algorithm
- **$f(C) \geq (e-1)/e \times f(C^*)$**

Submodular functions and the greedy algorithm

- A function **f** (defined on sets of some universe) is **submodular** if
  - ***for all*** sets **S**, **T** such that **S** is ***subset*** of **T** and **x** ***any*** element in the universe
  - **$f(S \cup \{x\}) - f(S) \geq f(T \cup \{x\}) - f(T)$**

- **Theorem:** For **all maximization** problems where the optimization function is **submodular**, the ***greedy*** algorithm has approximation factor

$$F = (e-1)/e$$

Again: Can you think of a more algorithmic approach to reducing the collection of frequent itemsets

Approximating a collection of frequent patterns

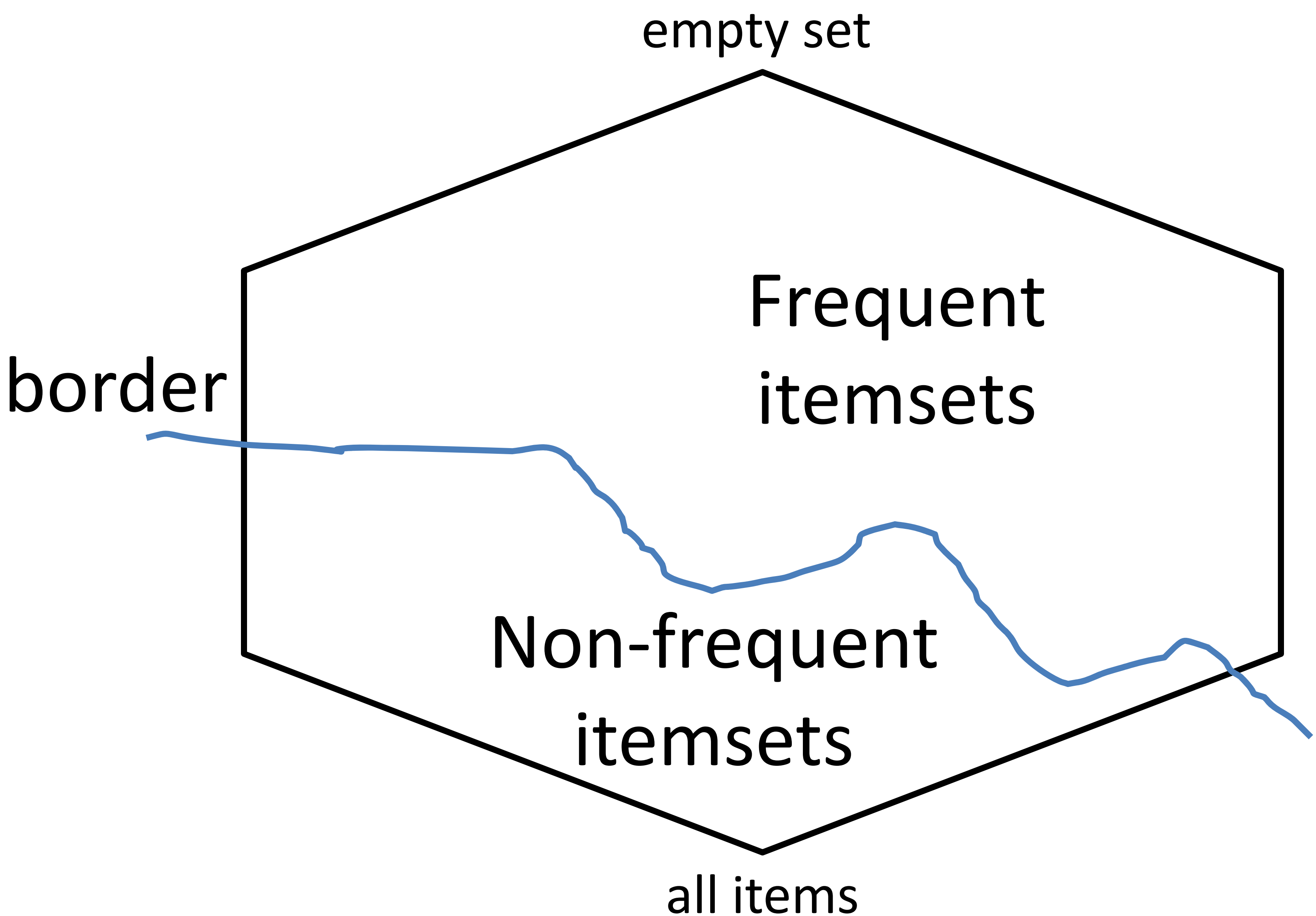
- Assume a collection of frequent patterns **S**

- Each pattern **X ϵ S** is ***described by the patterns that covers***
- **Cov(X) = { Y | Y ϵ S and Y subset of X}**

- **Problem:** Find **k** patterns from **S** to form set **C** such that

$$\left| \bigcup_{X \in C} \mathrm{Cov}(X) \right|$$

is maximized


empty set
Frequent itemsets
border
Non-frequent itemsets
all items