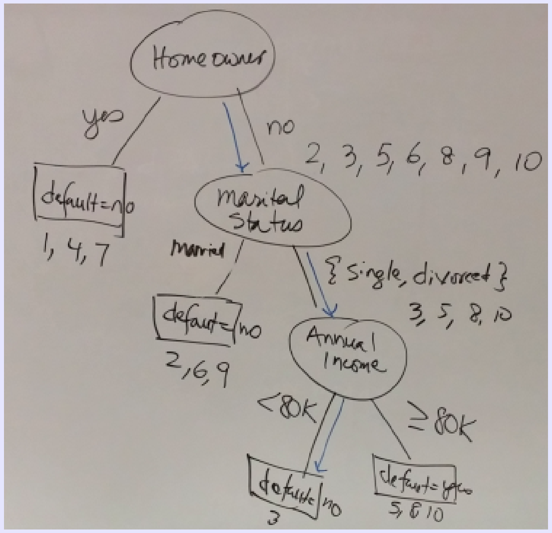
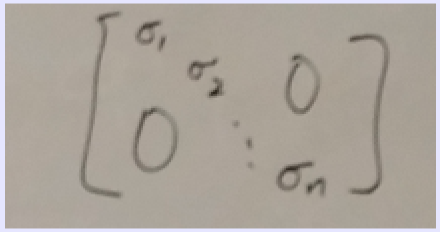
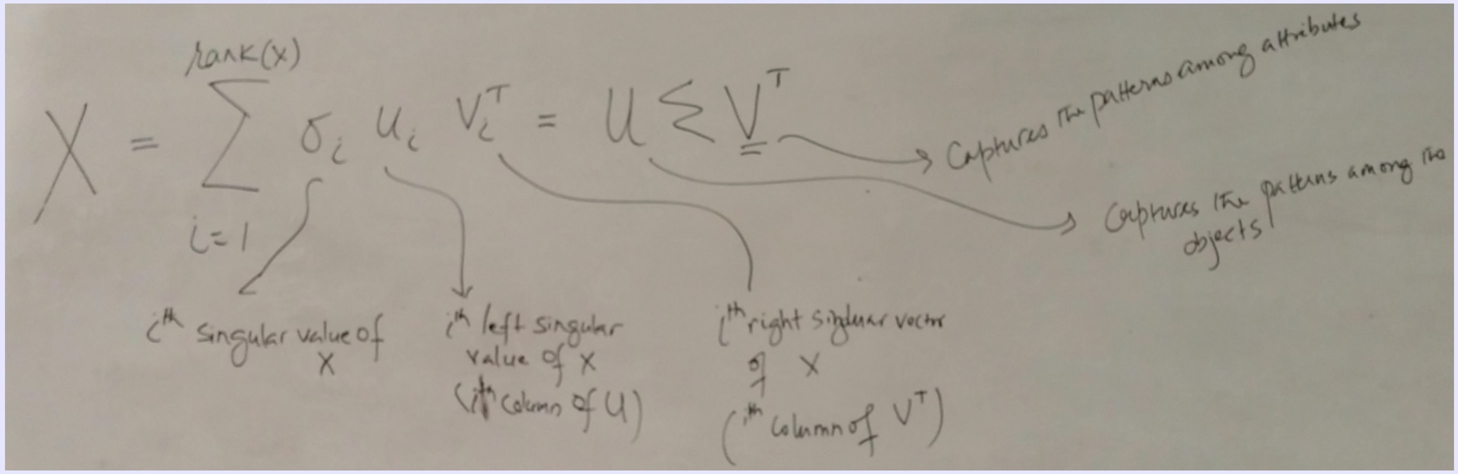
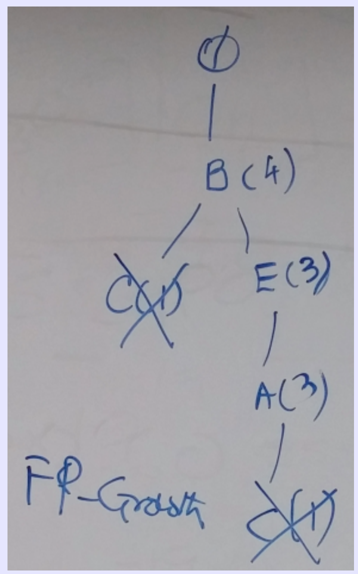
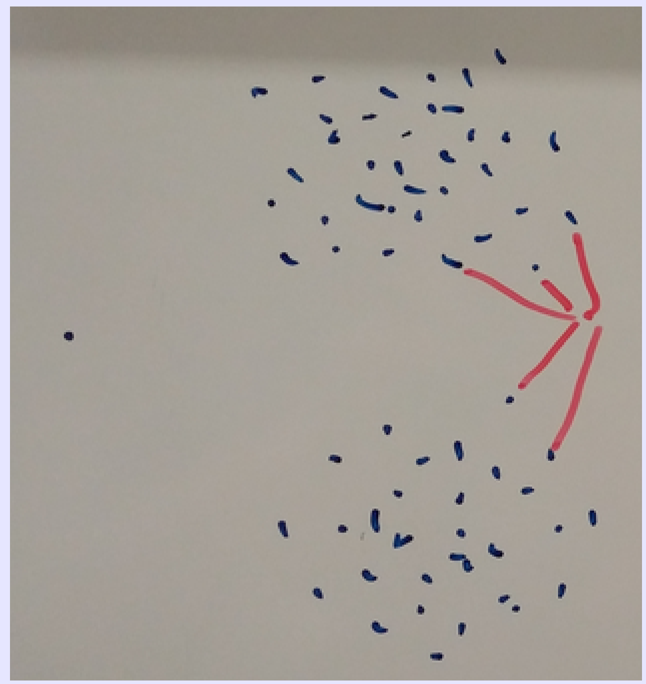










CS 422: Data Mining

Vijay K. Gurbani, Ph.D.,
Illinois Institute of Technology

Association Analysis (Rules)

Association Rule Mining

- One of the early examples of data mining.
- Interested in observing which objects occur together:
  - Grocery shopping (*market-basket analysis*)
  - Website visits
- Notice that we are not ***recommending*** similar items, just seeing which items co-occur.
  - Recommendation is for a later lecture.

Association Rule Mining

- Given a set of transactions, find rules that will predict the occurrence of an item based on the occurrences of other items in the transaction. (Note: Implication means co-occurrence, not causality!)

TID	*Items*
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Example of Association Rules

{Diaper} → {Beer}
{Milk, Bread} → {Eggs,Coke}

{Beer, Bread} → {Milk}

Antecedent → Consequent

Association Rule Mining

- Preliminaries

Let $\mathcal{I} = \{x_1, x_2, ..., x_m\}$ be a set of elements called *items*.
A set $X \subseteq \mathcal{I}$ is called an *itemset*.
An itemset of cardinality k is called a k-itemset.
$\mathcal{I}^{(k)}$ is the set of all k-itemsets.
Let $\mathcal{T} = \{t_1, t_2, ..., t_n\}$ be another set of elements called transaction identifiers, or *tids*.
A set $T \subseteq \mathcal{T}$ is called an *tidset*.
A *transaction* is a tuple of the form (t, X) where t $\in$ T is a unique transaction identifier, and X is an itemset.

Database Representation

A binary database $\mathbf{D}$ is a binary relation on the set of tids and items, that is, $\mathbf{D} \subseteq \mathcal{T} \times \mathcal{I}$. We say that tid $t \in \mathcal{T}$ *contains* item $x \in \mathcal{I}$ iff $(t, x) \in \mathbf{D}$. In other words, $(t, x) \in \mathbf{D}$ iff $x \in X$ in the tuple $\langle t, X \rangle$. We say that tid t *contains* itemset $X = \{x_1, x_2, \ldots, x_k\}$ iff $(t, x_i) \in \mathbf{D}$ for all $i = 1, 2, \ldots, k$.

For a set X, we denote by 2^X the powerset of X, that is, the set of all subsets of X. Let $\mathbf{i} : 2^{\mathcal{T}} \rightarrow 2^{\mathcal{I}}$ be a function, defined as follows:

$$\mathbf{i}(T) = \{x \mid \forall t \in T,\ t \text{ contains } x\} \tag{8.1}$$

where $T \subseteq \mathcal{T}$, and $\mathbf{i}(T)$ is the set of items that are common to *all* the transactions in the tidset T. In particular, $\mathbf{i}(t)$ is the set of items contained in tid $t \in \mathcal{T}$.

Association Rule Mining

- Preliminaries

D	A	B	C	D	E
1	1	1	0	1	1
2	0	1	1	0	1
3	1	1	0	1	1
4	1	1	1	0	1
5	1	1	1	1	1
6	0	1	1	1	0

(a) Binary database

t	$\mathbf{i}(t)$
1	$ABDE$
2	BCE
3	$ABDE$
4	$ABCE$
5	$ABCDE$
6	BCD

(b) Transaction database

x	A	B	C	D	E
$\mathbf{t}(x)$	1 3 4 5	1 2 3 4 5 6	2 4 5 6	1 3 5 6	1 2 3 4 5

(c) Vertical database

Figure 8.1. An example database.

Association Rule Mining

- Preliminaries

Support and Frequent Itemsets

The *support* of an itemset X in a dataset $\mathbf{D}$, denoted $sup(X, \mathbf{D})$, is the number of transactions in $\mathbf{D}$ that contain X:

$$sup(X, \mathbf{D}) = \left|\{t \mid \langle t, \mathbf{i}(t)\rangle \in \mathbf{D} \text{ and } X \subseteq \mathbf{i}(t)\}\right| = |\mathbf{t}(X)|$$

The *relative support* of X is the fraction of transactions that contain X:

$$rsup(X, \mathbf{D}) = \frac{sup(X, \mathbf{D})}{|\mathbf{D}|}$$

$\mathbf{D}$	A	B	C	D	E
1	1	1	0	1	1
2	0	1	1	0	1
3	1	1	0	1	1
4	1	1	1	0	1
5	1	1	1	1	1
6	0	1	1	1	0

(a) Binary database

t	$\mathbf{i}(t)$
1	$ABDE$
2	BCE
3	$ABDE$
4	$ABCE$
5	$ABCDE$
6	BCD

(b) Transaction database

sup({A,B}) = 4 rsup({A,B}) = 4/6 = 0.67

sup({B}) = 6 rsup({B}) = 6/6 = 1.00

Association Rule Mining

- Preliminaries

Support and Frequent Itemsets

The *support* of an itemset X in a dataset $\mathbf{D}$, denoted $sup(X, \mathbf{D})$, is the number of transactions in $\mathbf{D}$ that contain X:

$$sup(X, \mathbf{D}) = \left|\{t \mid \langle t, \mathbf{i}(t)\rangle \in \mathbf{D} \text{ and } X \subseteq \mathbf{i}(t)\}\right| = |\mathbf{t}(X)|$$

The *relative support* of X is the fraction of transactions that contain X:

$$rsup(X, \mathbf{D}) = \frac{sup(X, \mathbf{D})}{|\mathbf{D}|}$$

$\mathbf{D}$	A	B	C	D	E
1	1	1	0	1	1
2	0	1	1	0	1
3	1	1	0	1	1
4	1	1	1	0	1
5	1	1	1	1	1
6	0	1	1	1	0

(a) Binary database

t	$\mathbf{i}(t)$
1	$ABDE$
2	BCE
3	$ABDE$
4	$ABCE$
5	$ABCDE$
6	BCD

(b) Transaction database

sup({A,B}) = 4 rsup({A,B}) = 4/6 = 0.67
sup({B}) = 6 rsup({B}) = 6/6 = 1.00

We use $\mathcal{F}$ to denote the set of all itemsets, and $\mathcal{F}^{(k)}$ to denote the set of k-itemsets.

Thus, in our transaction database shown above,
$\mathcal{F}^{(3)} = \{\text{BCE, BCD}\}$
$\mathcal{F}^{(4)} = \{\text{ABDE}\}$
$\mathcal{F}^{(5)} = \{\text{ABCDE}\}$

Association Rule Mining

- Preliminaries

D	A	B	C	D	E
1	1	1	0	1	1
2	0	1	1	0	1
3	1	1	0	1	1
4	1	1	1	0	1
5	1	1	1	1	1
6	0	1	1	1	0

(a) Binary database

t	$\mathbf{i}(t)$
1	$ABDE$
2	BCE
3	$ABDE$
4	$ABCE$
5	$ABCDE$
6	BCD

(b) Transaction database

Support and Frequent Itemsets

The *support* of an itemset X in a dataset $\mathbf{D}$, denoted $sup(X, \mathbf{D})$, is the number of transactions in $\mathbf{D}$ that contain X:

$$sup(X, \mathbf{D}) = \left|\{t \mid \langle t, \mathbf{i}(t)\rangle \in \mathbf{D} \text{ and } X \subseteq \mathbf{i}(t)\}\right| = |\mathbf{t}(X)|$$

The *relative support* of X is the fraction of transactions that contain X:

$$rsup(X, \mathbf{D}) = \frac{sup(X, \mathbf{D})}{|\mathbf{D}|}$$

sup({A,B}) = 4 rsup({A,B}) = 4/6 = 0.67
sup({B}) = 6 rsup({B}) = 6/6 = 1.00

We use $\mathcal{F}$ to denote the set of all itemsets, and $\mathcal{F}^{(k)}$ to denote the set of k-itemsets.

Thus, in our transaction database shown above,
$\mathcal{F}^{(3)} = \{\text{BCE, BCD}\}$
$\mathcal{F}^{(4)} = \{\text{ABDE}\}$
$\mathcal{F}^{(5)} = \{\text{ABCDE}\}$

The itemset mining problem:
Given a minimum support threshold (*minsup*), find all itemsets *X*, s.t. *sup(X) >= minsup*.

Can be expressed as absolute, *sup(.)*, or relative, *rsup(.)*

Association Rule Mining

- Preliminaries

D	A	B	C	D	E
1	1	1	0	1	1
2	0	1	1	0	1
3	1	1	0	1	1
4	1	1	1	0	1
5	1	1	1	1	1
6	0	1	1	1	0

(a) Binary database

t	$\mathbf{i}(t)$
1	$ABDE$
2	BCE
3	$ABDE$
4	$ABCE$
5	$ABCDE$
6	BCD

(b) Transaction database

Frequent itemsets: An itemset X is frequent if sup(X) ≥ *minsup*, where *minsup* is a user specified minimum support threshold. (If *minsup* is a fraction, then relative support is implied.)

Example: Let *minsup* = 3 (in relative support term, *minsup* = 0.5); show all such frequent itemsets.

Total possible subsets: $2^{|I|}$ (exponential; before hand we do not know how many *k*-itemsets we will end up having.)

Turns out, it is 19. The set of all 19 frequent *k*-itemsets grouped by their support value is:

Table 8.1. Frequent itemsets with $minsup = 3$

sup	itemsets
6	B
5	E, BE
4	$A, C, D, AB, AE, BC, BD, ABE$
3	$AD, CE, DE, ABD, ADE, BCE, BDE, ABDE$

$$\mathcal{F}^{(1)} = \{A, B, C, D, E\}$$
$$\mathcal{F}^{(2)} = \{AB, AD, AE, BC, BD, BE, CE, DE\}$$
$$\mathcal{F}^{(3)} = \{ABD, ABE, ADE, BCE, BDE\}$$
$$\mathcal{F}^{(4)} = \{ABDE\}$$

Association Rule Mining

- Preliminaries

D	A	B	C	D	E
1	1	1	0	1	1
2	0	1	1	0	1
3	1	1	0	1	1
4	1	1	1	0	1
5	1	1	1	1	1
6	0	1	1	1	0

(a) Binary database

t	$\mathbf{i}(t)$
1	$ABDE$
2	BCE
3	$ABDE$
4	$ABCE$
5	$ABCDE$
6	BCD

(b) Transaction database

Frequent itemsets: An itemset X is frequent if sup(X) ≥ *minsup*, where *minsup* is a user specified minimum support threshold. (If *minsup* is a fraction, then relative support is implied.)

Example: Let *minsup* = 3 (in relative support term, *minsup* = 0.5); show all such frequent itemsets.

Total possible subsets: $2^{|I|}$ (exponential; before hand we do not know how many *k*-itemsets we will end up having.)

Turns out, it is 19. The set of all 19 frequent *k*-itemsets grouped by their support value is:

Question: How do we generate all itemsets that are frequent? (i.e., have a support of at least *minsup* = 3?)

Table 8.1. Frequent itemsets with $minsup = 3$

sup	itemsets
6	B
5	E, BE
4	$A, C, D, AB, AE, BC, BD, ABE$
3	$AD, CE, DE, ABD, ADE, BCE, BDE, ABDE$

$$\mathcal{F}^{(1)} = \{A, B, C, D, E\}$$
$$\mathcal{F}^{(2)} = \{AB, AD, AE, BC, BD, BE, CE, DE\}$$
$$\mathcal{F}^{(3)} = \{ABD, ABE, ADE, BCE, BDE\}$$
$$\mathcal{F}^{(4)} = \{ABDE\}$$

Association Rule Mining

Question: How do we generate all itemsets that are frequent? (i.e., have a support of at least *minsup* = 3?)

Naive algorithm:

```
∀ x ⊆ I
  compute_support(x)
  if (sup(x) ≥ minsup)
     print x, sup(x)
```

Association Rule Mining

Question: How do we generate all itemsets that are frequent? (i.e., have a support of at least *minsup* = 3?)

Naive algorithm:

$\forall\ x \subseteq I$ → Subset enumeration to generate *candidates*. We do not know yet whether the itemset is frequent or not until we compute support, hence it is a *candidate.*

compute_support(x)

if (sup(x) ≥ minsup)

print x, sup(x)

Association Rule Mining

Question: How do we generate all itemsets that are frequent? (i.e., have a support of at least *minsup* = 3?)

Naive algorithm:

$\forall$ x $\subseteq$ I

  compute_support(x)

  if (sup(x) ≥ minsup)

    print x, sup(x)

Subset enumeration to generate candidates. We do not know yet whether the itemset is frequent or not until we compute support, hence it is a *candidate.*

Here is where we go to the dataset and compute support, *for each itemset*.

Association Rule Mining

Question: How do we generate all itemsets that are frequent? (i.e., have a support of at least *minsup* = 3?)

Naive algorithm:

$\forall$ x $\subseteq$ I

  compute_support(x)

  if (sup(x) ≥ minsup)

    print x, sup(x)

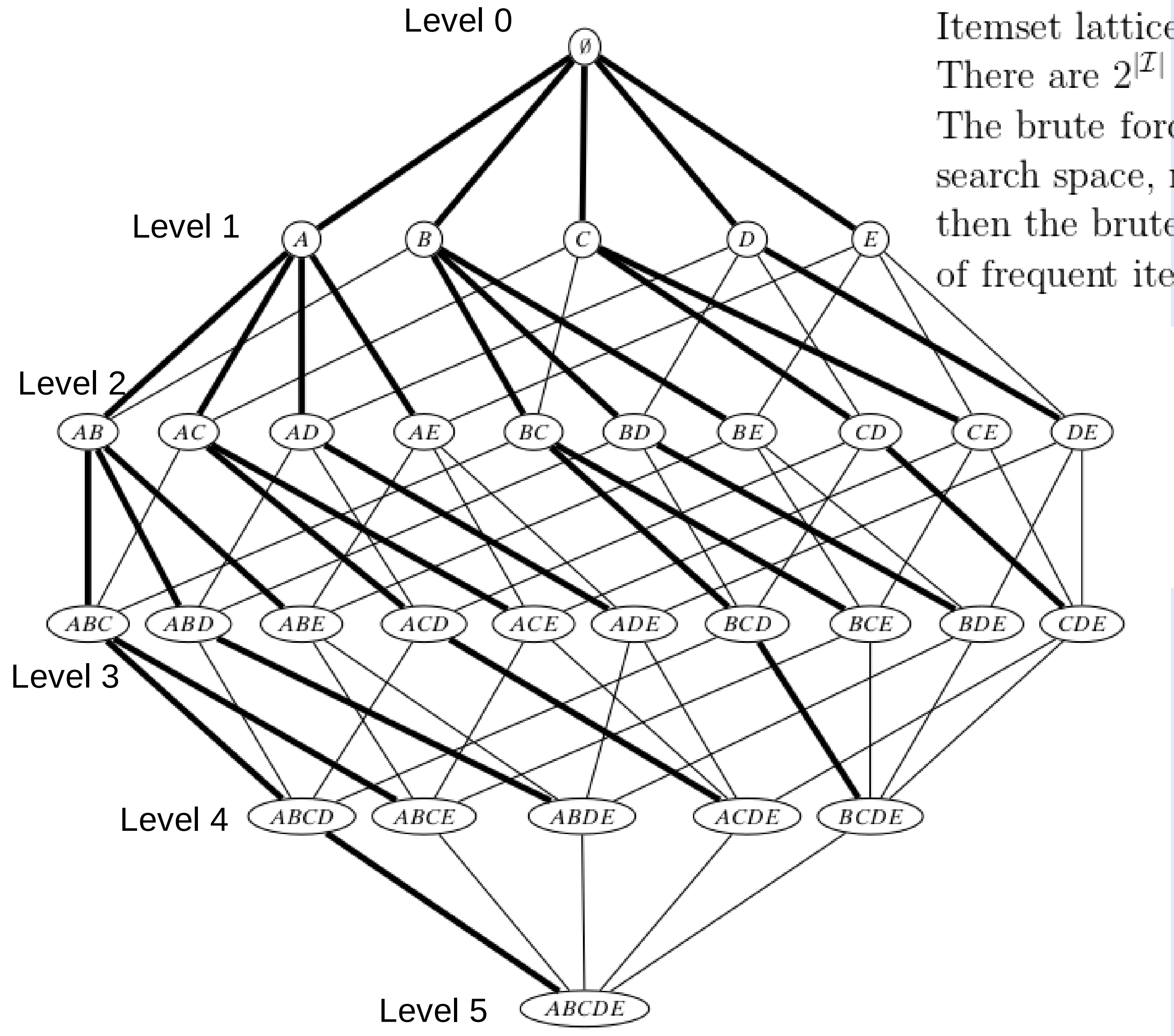
Subset enumeration to generate candidates. We do not know yet whether the itemset is frequent or not until we compute support, hence it is a *candidate.*

Here is where we go to the dataset and compute support, *for each itemset*.

Computational complexity:

$$O(\underbrace{2^{|I|}}_{\text{Enumeration}} * \underbrace{|D| * |I|}_{\text{Support computation}})$$

Frequent Itemset Generation: Brute Force Method

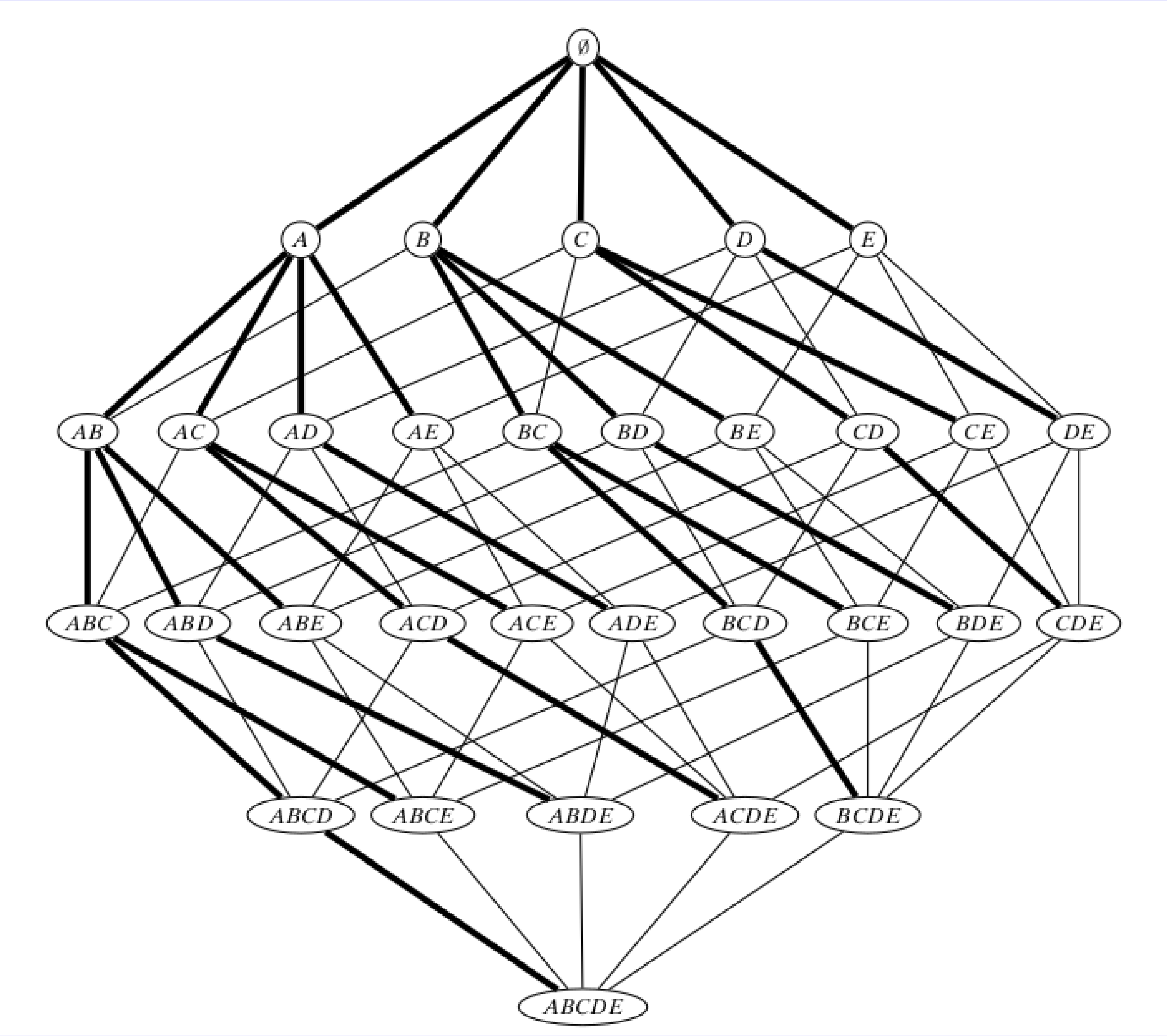


Itemset lattice for $\mathcal{I} = \{A, B, C, D, E\}$.
There are $2^{|\mathcal{I}|} = 32$ possible itemsets.
The brute force method explores the entire itemset search space, regardless of *minsup*. If $minsup = 3$, then the brute-force search method would output the set of frequent itemsets shown below.

Table 8.1. Frequent itemsets with $minsup = 3$

sup	itemsets
6	*B*
5	*E*, *BE*
4	*A*, *C*, *D*, *AB*, *AE*, *BC*, *BD*, *ABE*
3	*AD*, *CE*, *DE*, *ABD*, *ADE*, *BCE*, *BDE*, *ABDE*

Frequent Itemset Generation: Brute Force Method



ALGORITHM 8.1. Algorithm BruteForce

BruteForce $(\mathbf{D}, \mathcal{I}, minsup)$:

$\mathcal{F} \leftarrow \emptyset$ `// set of frequent itemsets`

foreach $X \subseteq \mathcal{I}$ **do**

  $sup(X) \leftarrow$ ComputeSupport $(X, \mathbf{D})$

  if $sup(X) \geq minsup$ **then**

    $\mathcal{F} \leftarrow \mathcal{F} \cup \{(X, sup(X))\}$

return $\mathcal{F}$

ComputeSupport $(X, \mathbf{D})$:

$sup(X) \leftarrow 0$

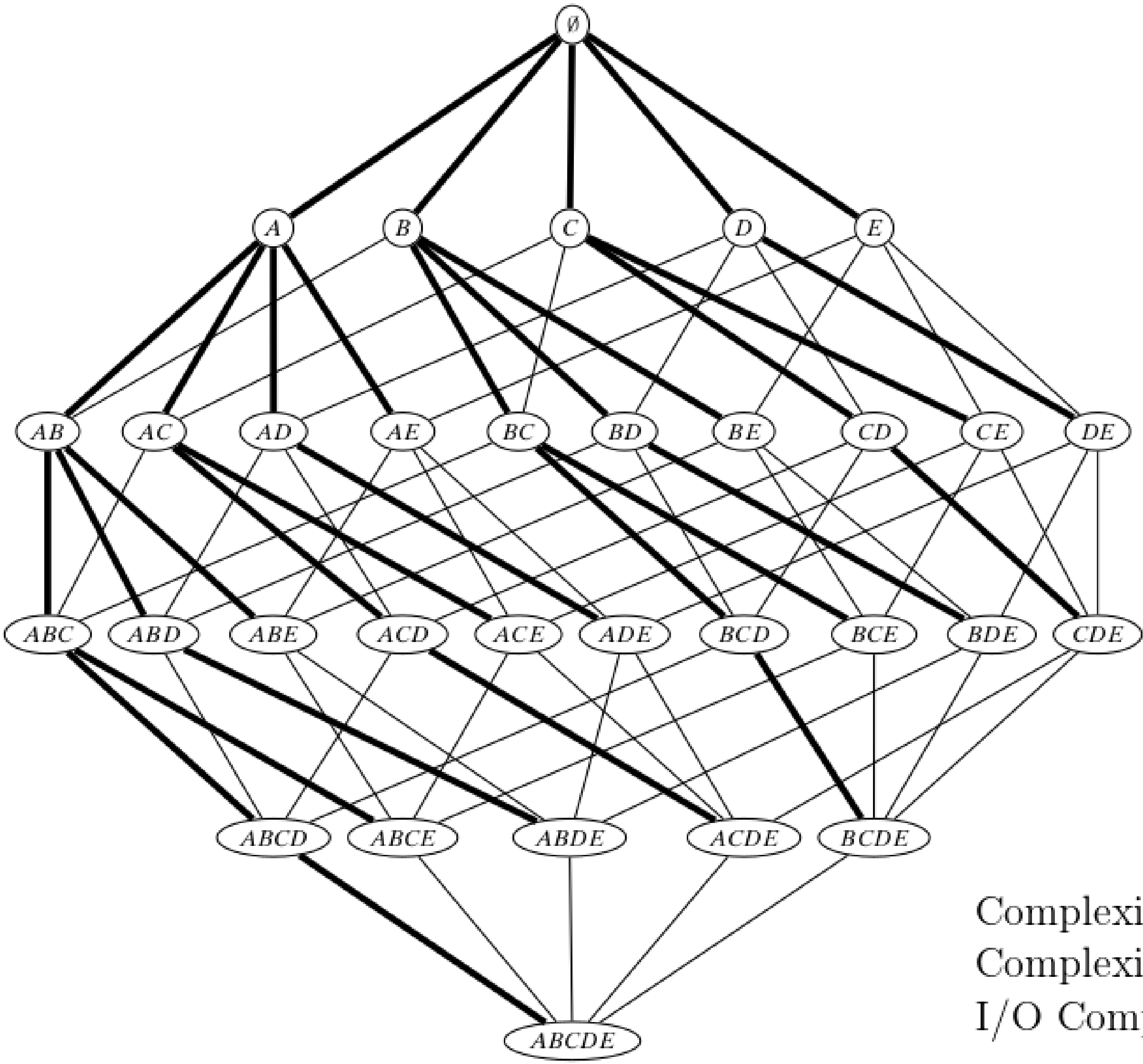
foreach $\langle t, \mathbf{i}(t) \rangle \in \mathbf{D}$ **do**

  if $X \subseteq \mathbf{i}(t)$ **then**

    $sup(X) \leftarrow sup(X) + 1$

return $sup(X)$

Frequent Itemset Generation: Brute Force Method



ALGORITHM 8.1. Algorithm BruteForce

BruteForce $(\mathbf{D}, \mathcal{I}, minsup)$:

$\mathcal{F} \leftarrow \emptyset$ `// set of frequent itemsets`

foreach $X \subseteq \mathcal{I}$ **do**

- $sup(X) \leftarrow$ ComputeSupport $(X, \mathbf{D})$
- **if** $sup(X) \geq minsup$ **then**
  - $\mathcal{F} \leftarrow \mathcal{F} \cup \{(X, sup(X))\}$

return $\mathcal{F}$

ComputeSupport $(X, \mathbf{D})$:

$sup(X) \leftarrow 0$

foreach $\langle t, \mathbf{i}(t) \rangle \in \mathbf{D}$ **do**

- **if** $X \subseteq \mathbf{i}(t)$ **then**
  - $sup(X) \leftarrow sup(X) + 1$

return $sup(X)$

Complexity of ComputeSupport: $\mathcal{O}(|\mathcal{I}| * D)$

Complexity of BruteForce: $\mathcal{O}(|\mathcal{I}| * D * 2^{|\mathcal{I}|})$

I/O Complexity of BruteForce: $\mathcal{O}(2^{|\mathcal{I}|})$ database scans