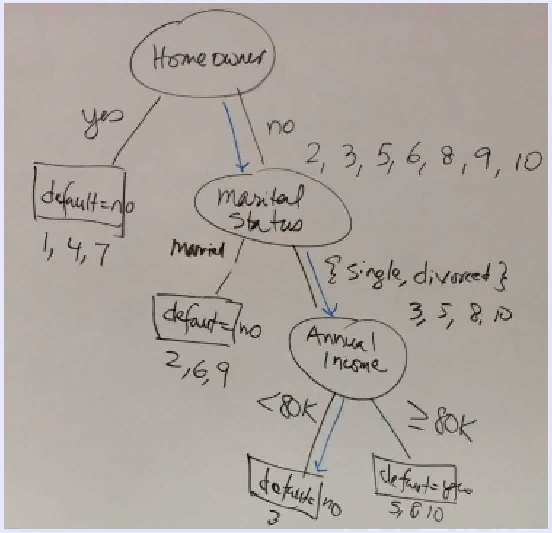
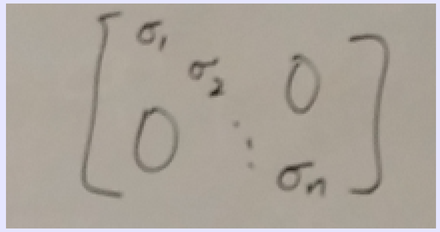
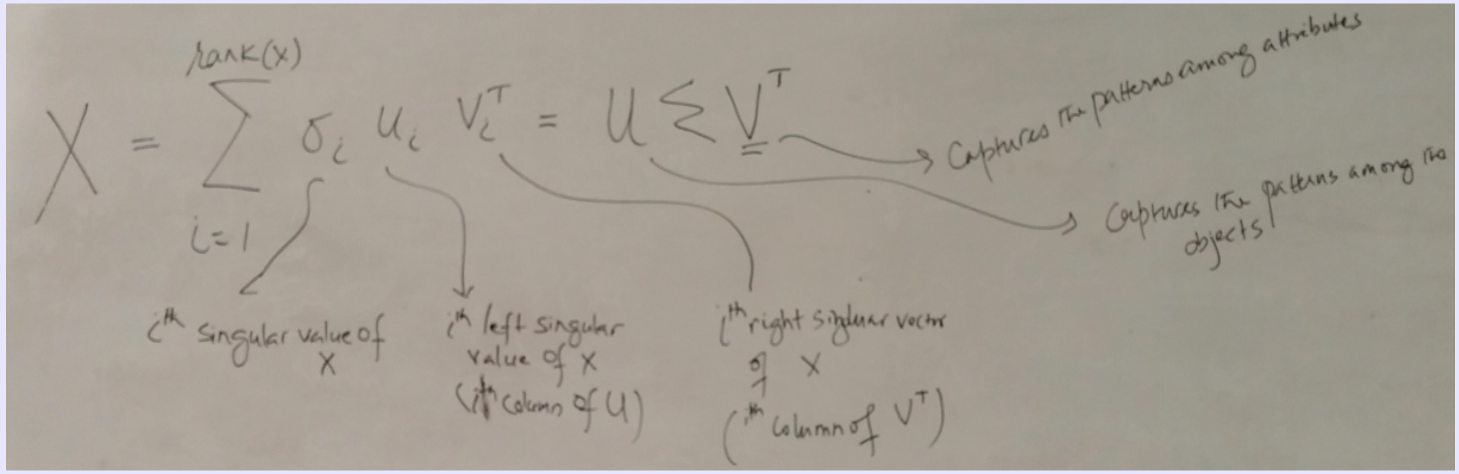
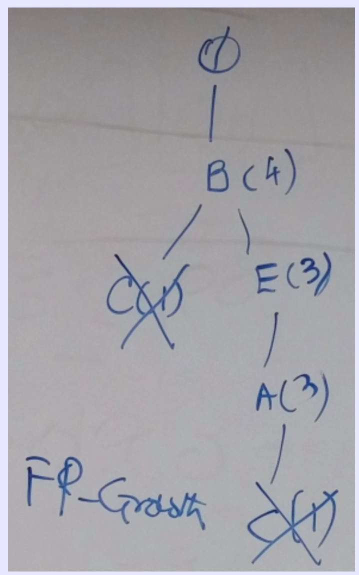
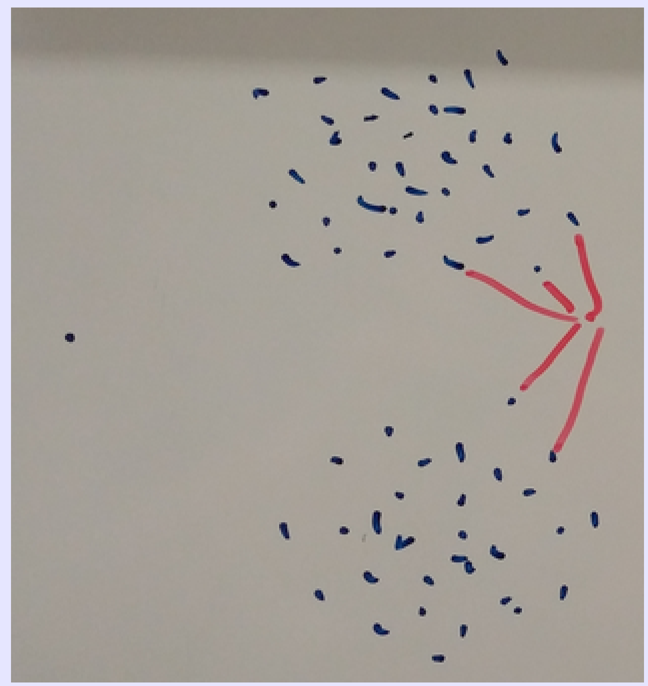










CS 422: Data Mining
Vijay K. Gurbani, Ph.D.,
Illinois Institute of Technology

Association Analysis (Rules)

Association Rule Mining

- Goal of Association Rule Mining: Given a set of transactions, T, find all rules having:
  - support >= *minsup*
  - confidence >= *minconf*
- How do we get there?
- Two steps:
  - Frequent itemset generation: find all items that satisfy *minsup* threshold (frequent itemsets). (Is computationally expensive!!)
  - Rule generation: extract all high-confidence rules from the frequent itemsets (strong rules).

Frequent Itemset Generation: The Apriori Approach

- We looked at the brute-force approach to generate itemsets. Can we do better?
- Brute-force approach wastes computations because many of the candidates that it generates may not be frequent.
- Instead, note:

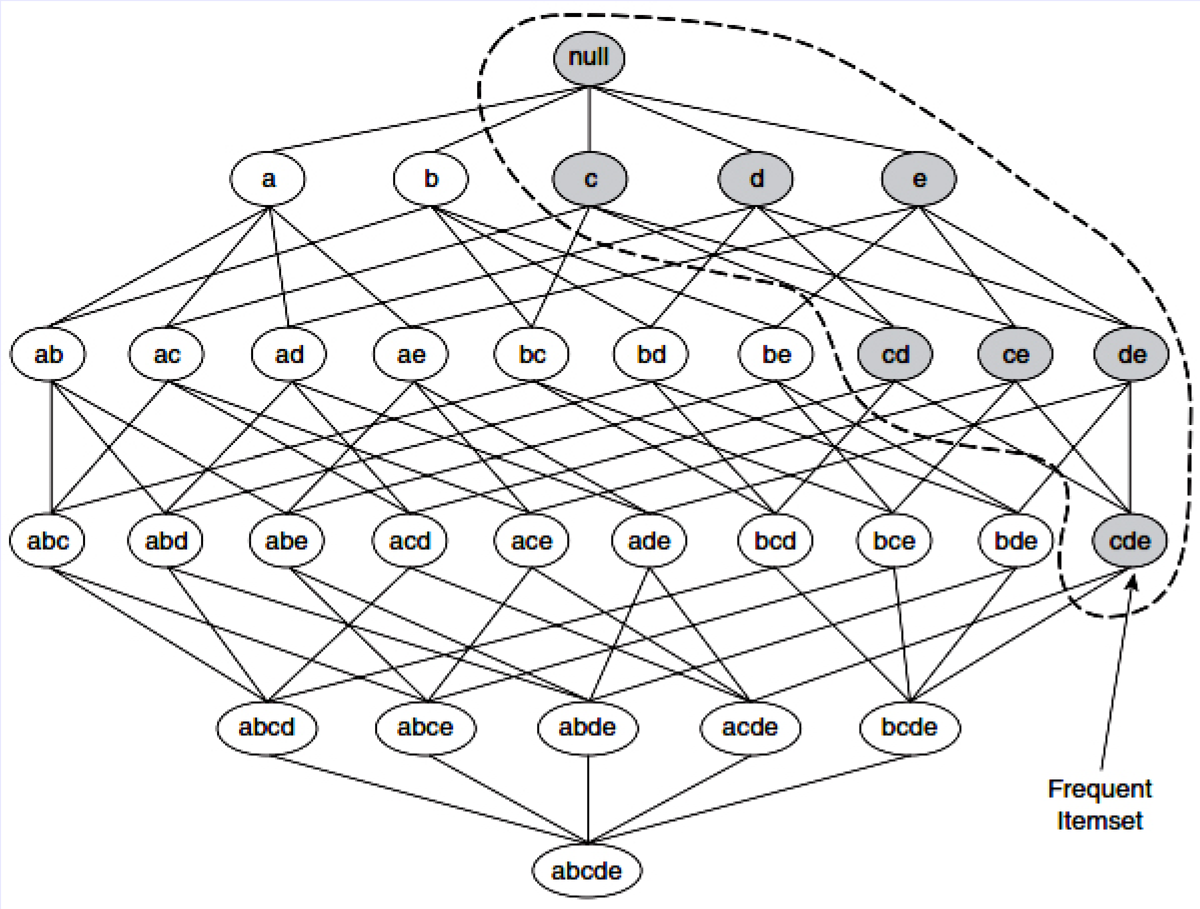
Let X and Y be two itemsets $\in \mathcal{I}$ such that $X \subseteq Y$.
If so, then $\sup(X) \geq \sup(Y)$

E.g. X=ABCD, Y=ABCDE, then sup(ABCD) >= sup(ABCDE).

- This leads to …

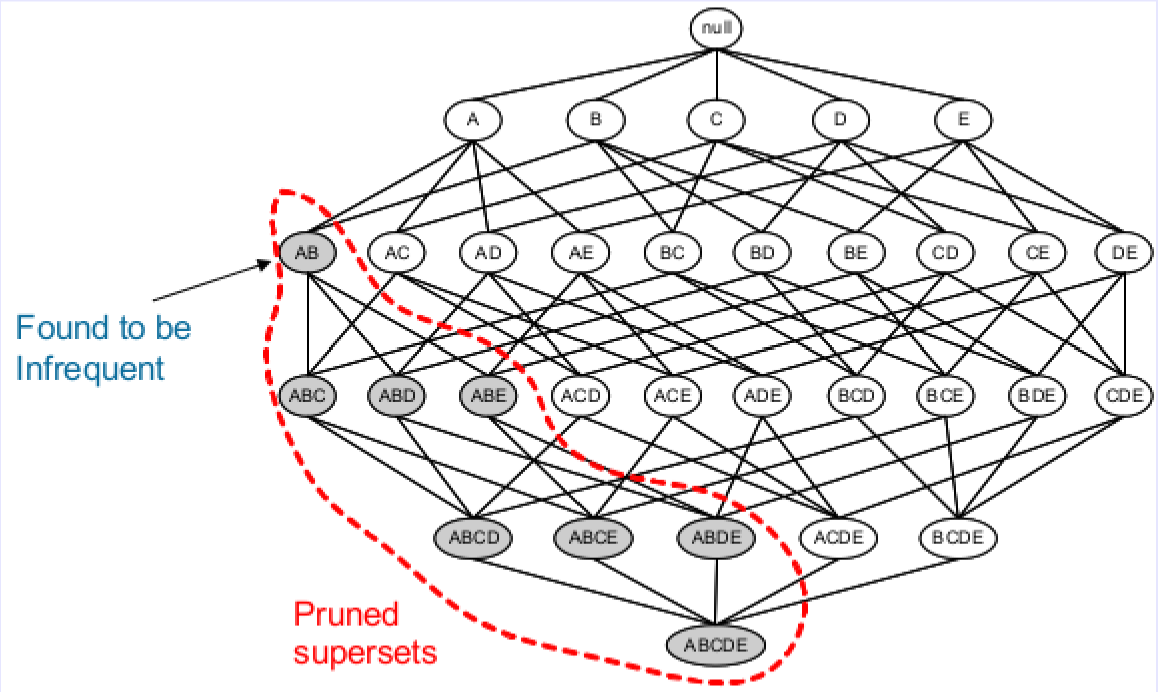
Frequent Itemset Generation: The Apriori Approach

- The *Apriori* principle:
  - If an itemset is frequent, then all of its subsets must be frequent as well.



Frequent Itemset Generation: The Apriori Approach

- The *Apriori* principle:
  - Conversely, if an itemset is infrequent, then all of its supersets must be infrequent as well.



Frequent Itemset Generation: The Apriori Approach

minsup = 3

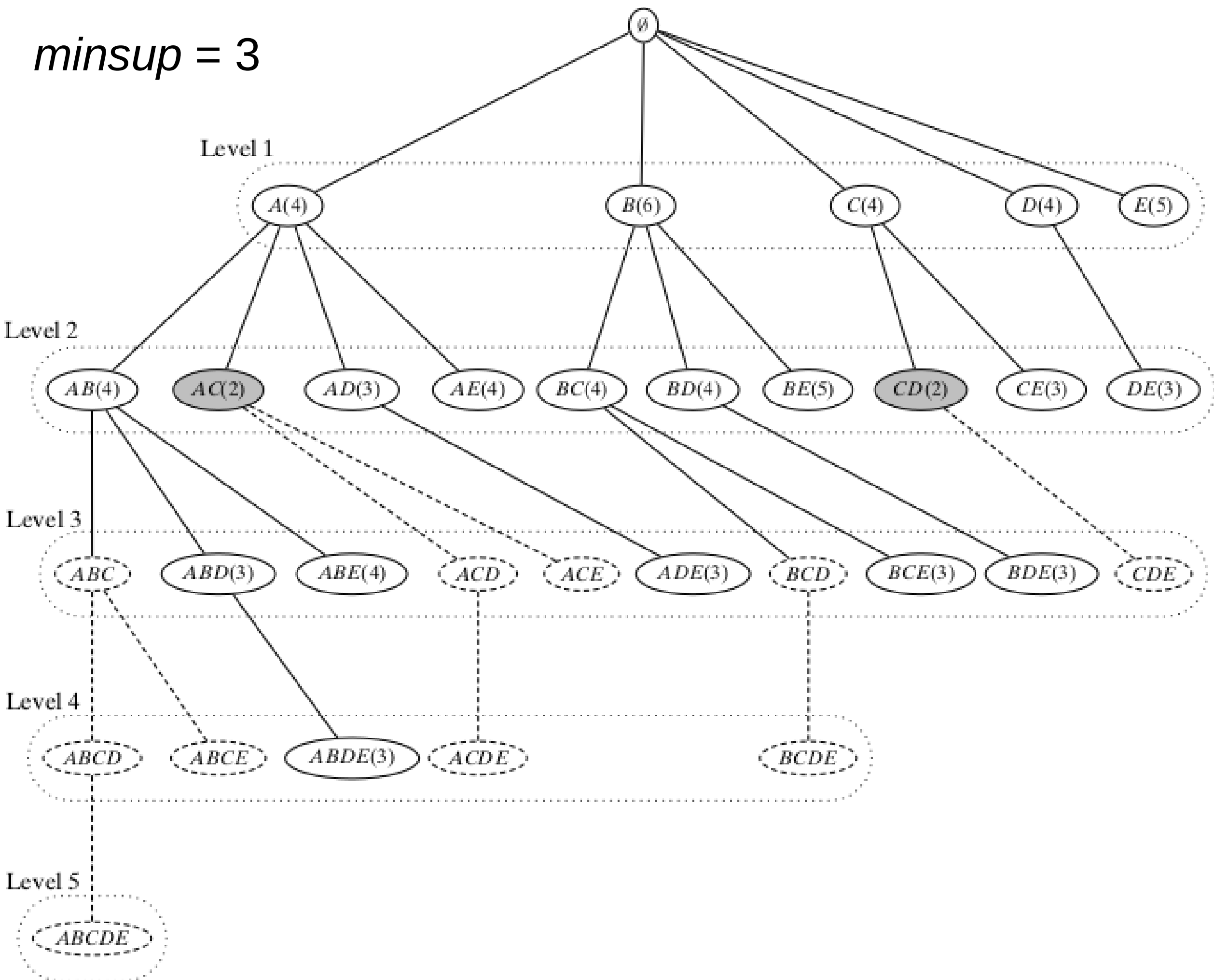


Figure 8.3. Apriori: prefix search tree and effect of pruning. Shaded nodes indicate infrequent itemsets, whereas dashed nodes and lines indicate all of the pruned nodes and branches. Solid lines indicate frequent itemsets.

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	**i**(*t*)
1	*ABDE*
2	*BCE*
3	*ABDE*
4	*ABCE*
5	*ABCDE*
6	*BCD*

(b) Transaction database

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: The Apriori Approach

ALGORITHM 8.2. Algorithm APRIORI

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

$\mathcal{F} \leftarrow \emptyset$
$\mathcal{C}^{(1)} \leftarrow \{\emptyset\}$ `// Initial prefix tree with single items`
foreach $i \in \mathcal{I}$ **do** Add i as child of $\emptyset$ in $\mathcal{C}^{(1)}$ with $sup(i) \leftarrow 0$
$k \leftarrow 1$ `// k denotes the level`
while $\mathcal{C}^{(k)} \neq \emptyset$ **do**
    COMPUTESUPPORT $(\mathcal{C}^{(k)}, \mathbf{D})$
    foreach *leaf* $X \in \mathcal{C}^{(k)}$ **do**
        if $sup(X) \geq minsup$ **then** $\mathcal{F} \leftarrow \mathcal{F} \cup \{(X, sup(X))\}$
        else remove X from $\mathcal{C}^{(k)}$
    $\mathcal{C}^{(k+1)} \leftarrow$ EXTENDPREFIXTREE $(\mathcal{C}^{(k)})$
    $k \leftarrow k + 1$
return $\mathcal{F}^{(k)}$

COMPUTESUPPORT $(\mathcal{C}^{(k)}, \mathbf{D})$:

foreach $\langle t, \mathbf{i}(t) \rangle \in \mathbf{D}$ **do**
    foreach *k-subset* $X \subseteq \mathbf{i}(t)$ **do**
        if $X \in \mathcal{C}^{(k)}$ **then** $sup(X) \leftarrow sup(X) + 1$

EXTENDPREFIXTREE $(\mathcal{C}^{(k)})$:

foreach *leaf* $X_a \in \mathcal{C}^{(k)}$ **do**
    foreach *leaf* $X_b \in \text{SIBLING}(X_a)$, *such that* $b > a$ **do**
        $X_{ab} \leftarrow X_a \cup X_b$
        `// prune candidate if there are any infrequent subsets`
        if $X_j \in \mathcal{C}^{(k)}$, ***for all*** $X_j \subset X_{ab}$, *such that* $|X_j| = |X_{ab}| - 1$ **then**
            Add X_{ab} as child of X_a with $sup(X_{ab}) \leftarrow 0$
    if *no extensions from* X_a **then**
        remove X_a, and all ancestors of X_a with no extensions, from $\mathcal{C}^{(k)}$
return $\mathcal{C}^{(k)}$

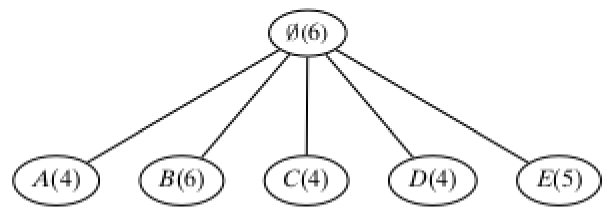
minsup = 3

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



(a) $\mathcal{C}^{(1)}$

Frequent Itemset Generation: The Apriori Approach

ALGORITHM 8.2. Algorithm APRIORI

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

$\mathcal{F} \leftarrow \emptyset$
$\mathcal{C}^{(1)} \leftarrow \{\emptyset\}$ `// Initial prefix tree with single items`
foreach $i \in \mathcal{I}$ **do** Add i as child of $\emptyset$ in $\mathcal{C}^{(1)}$ with $sup(i) \leftarrow 0$
$k \leftarrow 1$ `// k denotes the level`
while $\mathcal{C}^{(k)} \neq \emptyset$ **do**
  COMPUTESUPPORT $(\mathcal{C}^{(k)}, \mathbf{D})$
  foreach *leaf* $X \in \mathcal{C}^{(k)}$ **do**
    if $sup(X) \geq minsup$ **then** $\mathcal{F} \leftarrow \mathcal{F} \cup \{(X, sup(X))\}$
    else remove X from $\mathcal{C}^{(k)}$
  $\mathcal{C}^{(k+1)} \leftarrow$ EXTENDPREFIXTREE $(\mathcal{C}^{(k)})$
  $k \leftarrow k+1$
return $\mathcal{F}^{(k)}$

COMPUTESUPPORT $(\mathcal{C}^{(k)}, \mathbf{D})$:

foreach $\langle t, \mathbf{i}(t) \rangle \in \mathbf{D}$ **do**
  foreach *k-subset* $X \subseteq \mathbf{i}(t)$ **do**
    if $X \in \mathcal{C}^{(k)}$ **then** $sup(X) \leftarrow sup(X) + 1$

EXTENDPREFIXTREE $(\mathcal{C}^{(k)})$:

foreach *leaf* $X_a \in \mathcal{C}^{(k)}$ **do**
  foreach *leaf* $X_b \in$ SIBLING(X_a), *such that* $b > a$ **do**
    $X_{ab} \leftarrow X_a \cup X_b$
    `// prune candidate if there are any infrequent subsets`
    if $X_j \in \mathcal{C}^{(k)}$, ***for all*** $X_j \subset X_{ab}$, *such that* $|X_j| = |X_{ab}| - 1$ **then**
      Add X_{ab} as child of X_a with $sup(X_{ab}) \leftarrow 0$
  if *no extensions from* X_a **then**
    remove X_a, and all ancestors of X_a with no extensions, from $\mathcal{C}^{(k)}$
return $\mathcal{C}^{(k)}$

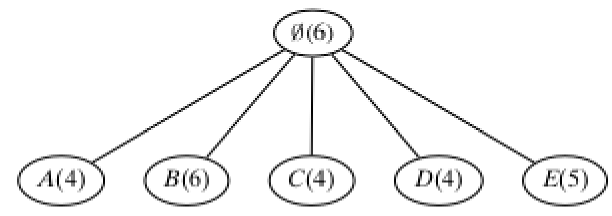
minsup = 3

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



(a) $\mathcal{C}^{(1)}$

We now extend the prefix tree from Level k to Level k+1: given two frequent k-itemsets (X_a and X_b), with common k-1 length prefix (i.e., two siblings with common parent), we generate (k+1) length candidates $X_{ab} = X_a \cup X_b$.
- X_{ab} retained only if it has no infrequent subset.

Frequent Itemset Generation: The Apriori Approach

ALGORITHM 8.2. Algorithm APRIORI

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

$\mathcal{F} \leftarrow \emptyset$
$\mathcal{C}^{(1)} \leftarrow \{\emptyset\}$ `// Initial prefix tree with single items`
foreach $i \in \mathcal{I}$ **do** Add i as child of $\emptyset$ in $\mathcal{C}^{(1)}$ with $sup(i) \leftarrow 0$
$k \leftarrow 1$ `// k denotes the level`
while $\mathcal{C}^{(k)} \neq \emptyset$ **do**
  COMPUTESUPPORT $(\mathcal{C}^{(k)}, \mathbf{D})$
  foreach *leaf* $X \in \mathcal{C}^{(k)}$ **do**
    if $sup(X) \geq minsup$ **then** $\mathcal{F} \leftarrow \mathcal{F} \cup \{(X, sup(X))\}$
    else remove X from $\mathcal{C}^{(k)}$
  $\mathcal{C}^{(k+1)} \leftarrow$ EXTENDPREFIXTREE $(\mathcal{C}^{(k)})$
  $k \leftarrow k+1$
return $\mathcal{F}^{(k)}$

COMPUTESUPPORT $(\mathcal{C}^{(k)}, \mathbf{D})$:

foreach $\langle t, \mathbf{i}(t) \rangle \in \mathbf{D}$ **do**
  foreach *k-subset* $X \subseteq \mathbf{i}(t)$ **do**
    if $X \in \mathcal{C}^{(k)}$ **then** $sup(X) \leftarrow sup(X) + 1$

EXTENDPREFIXTREE $(\mathcal{C}^{(k)})$:

foreach *leaf* $X_a \in \mathcal{C}^{(k)}$ **do**
  foreach *leaf* $X_b \in \text{SIBLING}(X_a)$, *such that* $b > a$ **do**
    $X_{ab} \leftarrow X_a \cup X_b$
    `// prune candidate if there are any infrequent subsets`
    if $X_j \in \mathcal{C}^{(k)}$, ***for all*** $X_j \subset X_{ab}$, *such that* $|X_j| = |X_{ab}| - 1$ **then**
      Add X_{ab} as child of X_a with $sup(X_{ab}) \leftarrow 0$
  if *no extensions from* X_a **then**
    remove X_a, and all ancestors of X_a with no extensions, from $\mathcal{C}^{(k)}$
return $\mathcal{C}^{(k)}$

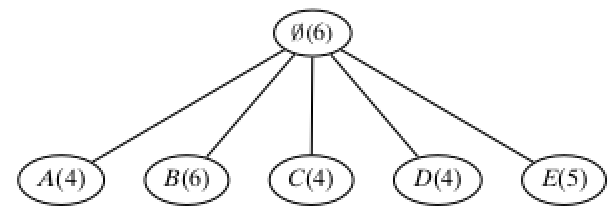
minsup = 3

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

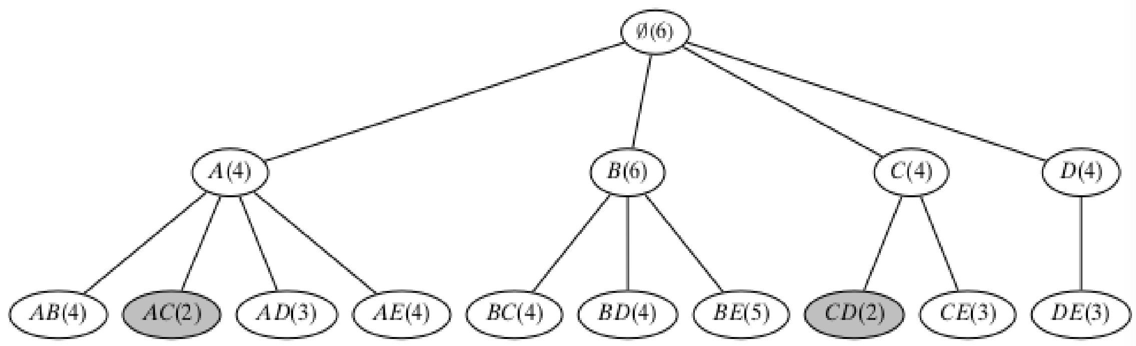
(a) Binary database

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

(b) Transaction database



(a) $\mathcal{C}^{(1)}$



(b) $\mathcal{C}^{(2)}$

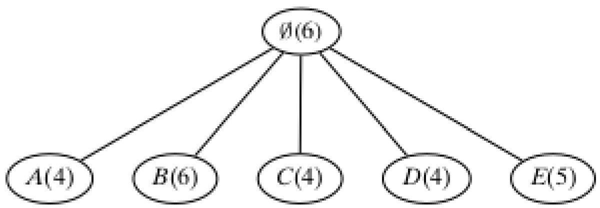
Frequent Itemset Generation: The Apriori Approach

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

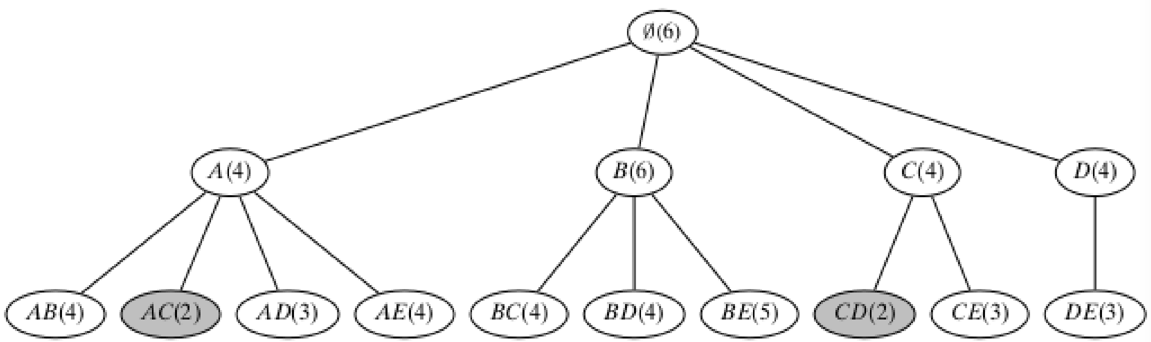
(a) Binary database

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

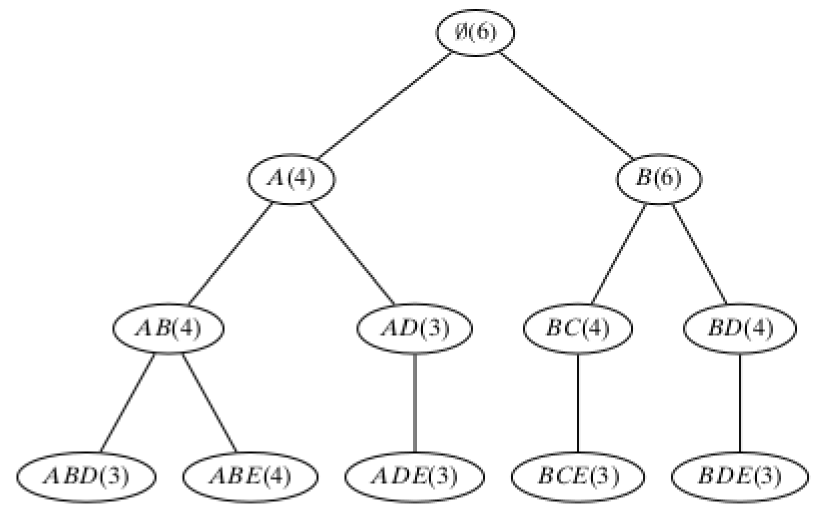
(b) Transaction database



(a) $\mathcal{C}^{(1)}$



(b) $\mathcal{C}^{(2)}$



(c) $\mathcal{C}^{(3)}$

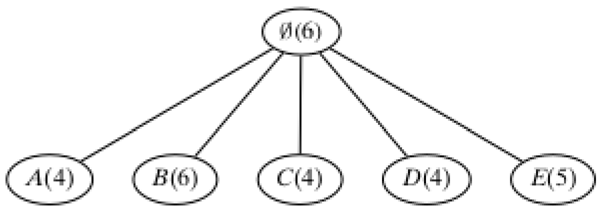
Frequent Itemset Generation: The Apriori Approach

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

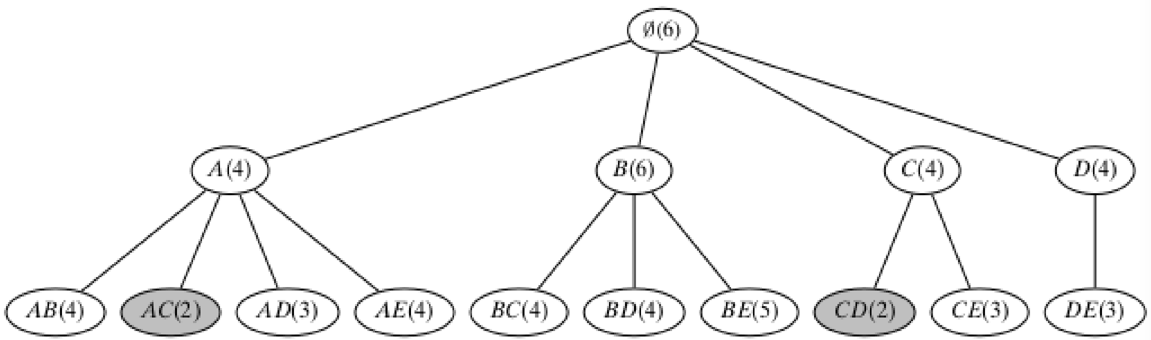
(a) Binary database

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

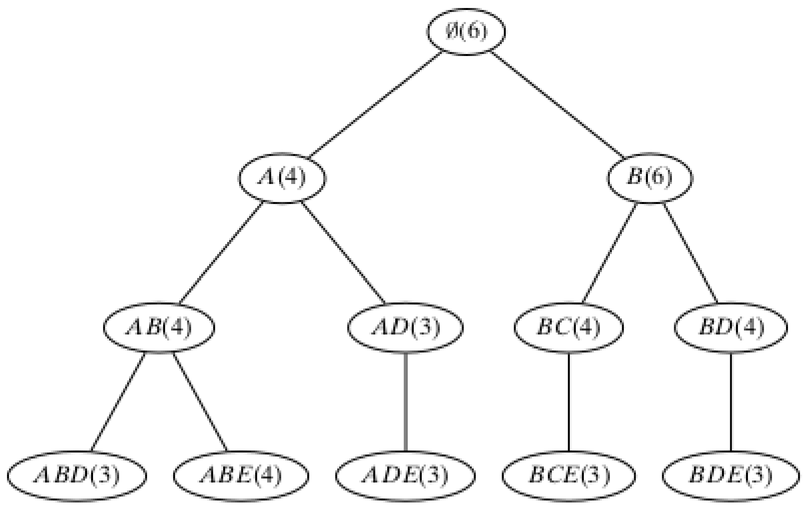
(b) Transaction database



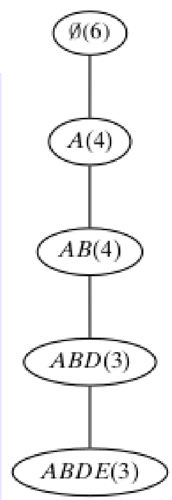
(a) $\mathcal{C}^{(1)}$



(b) $\mathcal{C}^{(2)}$



(c) $\mathcal{C}^{(3)}$



(d) $\mathcal{C}^{(4)}$

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: The Apriori Approach

ALGORITHM 8.2. Algorithm APRIORI

APRIORI $(\mathbf{D}, \mathcal{I}, minsup)$:
$\mathcal{F} \leftarrow \emptyset$
$\mathcal{C}^{(1)} \leftarrow \{\emptyset\}$ `// Initial prefix tree with single items`
foreach $i \in \mathcal{I}$ **do** Add i as child of $\emptyset$ in $\mathcal{C}^{(1)}$ with $sup(i) \leftarrow 0$
$k \leftarrow 1$ `// k denotes the level`
while $\mathcal{C}^{(k)} \neq \emptyset$ **do**
    COMPUTESUPPORT $(\mathcal{C}^{(k)}, \mathbf{D})$
    foreach *leaf* $X \in \mathcal{C}^{(k)}$ **do**
        if $sup(X) \geq minsup$ **then** $\mathcal{F} \leftarrow \mathcal{F} \cup \{(X, sup(X))\}$
        else remove X from $\mathcal{C}^{(k)}$
    $\mathcal{C}^{(k+1)} \leftarrow$ EXTENDPREFIXTREE $(\mathcal{C}^{(k)})$
    $k \leftarrow k+1$
return $\mathcal{F}^{(k)}$

COMPUTESUPPORT $(\mathcal{C}^{(k)}, \mathbf{D})$:
foreach $\langle t, \mathbf{i}(t) \rangle \in \mathbf{D}$ **do**
    foreach *k-subset* $X \subseteq \mathbf{i}(t)$ **do**
        if $X \in \mathcal{C}^{(k)}$ **then** $sup(X) \leftarrow sup(X) + 1$

EXTENDPREFIXTREE $(\mathcal{C}^{(k)})$:
foreach *leaf* $X_a \in \mathcal{C}^{(k)}$ **do**
    foreach *leaf* $X_b \in \text{SIBLING}(X_a)$, *such that* $b > a$ **do**
        $X_{ab} \leftarrow X_a \cup X_b$
        `// prune candidate if there are any infrequent subsets`
        if $X_j \in \mathcal{C}^{(k)}$, ***for all*** $X_j \subset X_{ab}$, *such that* $|X_j| = |X_{ab}| - 1$ **then**
            Add X_{ab} as child of X_a with $sup(X_{ab}) \leftarrow 0$
    if *no extensions from* X_a **then**
        remove X_a, and all ancestors of X_a with no extensions, from $\mathcal{C}^{(k)}$
return $\mathcal{C}^{(k)}$

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

Worst case complexity:

Complexity of Apriori: $\mathcal{O}(|\mathcal{I}| * D * 2^{|\mathcal{I}|})$ as all itemsets may be frequent. In practice, much lower due to pruning.
I/O costs are much lower, to the tune of $\mathcal{O}(|\mathcal{I}|)$ database scans as opposed to $\mathcal{O}(2^{|\mathcal{I}|})$ scans for brute-force. In practice, the algorithm only requires l database scans, where l is the length of the longest frequent itemset.

Association Rule Mining

- Rule mining: Preliminaries and foundation

Association Rules

An *association rule* is an expression $X \xrightarrow{s,c} Y$, where X and Y are itemsets and they are disjoint, that is, $X, Y \subseteq \mathcal{I}$, and $X \cap Y = \emptyset$. Let the itemset $X \cup Y$ be denoted as XY. The *support* of the rule is the number of transactions in which both X and Y co-occur as subsets:

$$s = sup(X \longrightarrow Y) = |\mathbf{t}(XY)| = sup(XY)$$

The *relative support* of the rule is defined as the fraction of transactions where X and Y co-occur, and it provides an estimate of the joint probability of X and Y:

$$rsup(X \longrightarrow Y) = \frac{sup(XY)}{|\mathbf{D}|} = P(X \wedge Y)$$

The *confidence* of a rule is the conditional probability that a transaction contains Y given that it contains X:

$$c = conf(X \longrightarrow Y) = P(Y|X) = \frac{P(X \wedge Y)}{P(X)} = \frac{sup(XY)}{sup(X)}$$

A rule is *frequent* if the itemset XY is frequent, that is, $sup(XY) \geq minsup$ and a rule is *strong* if $conf \geq minconf$, where $minconf$ is a user-specified minimum confidence threshold.

Association Rule Mining

- Rule mining: Preliminaries and foundation

Association Rules

An *association rule* is an expression $X \xrightarrow{s,c} Y$, where X and Y are itemsets and they are disjoint, that is, $X, Y \subseteq \mathcal{I}$, and $X \cap Y = \emptyset$. Let the itemset $X \cup Y$ be denoted as XY. The *support* of the rule is the number of transactions in which both X and Y co-occur as subsets:

$$s = sup(X \longrightarrow Y) = |\mathbf{t}(XY)| = sup(XY)$$

The *relative support* of the rule is defined as the fraction of transactions where X and Y co-occur, and it provides an estimate of the joint probability of X and Y:

$$rsup(X \longrightarrow Y) = \frac{sup(XY)}{|\mathbf{D}|} = P(X \wedge Y)$$

The *confidence* of a rule is the conditional probability that a transaction contains Y given that it contains X:

$$c = conf(X \longrightarrow Y) = P(Y|X) = \frac{P(X \wedge Y)}{P(X)} = \frac{sup(XY)}{sup(X)}$$

A rule is *frequent* if the itemset XY is frequent, that is, $sup(XY) \geq minsup$ and a rule is *strong* if $conf \geq minconf$, where $minconf$ is a user-specified minimum confidence threshold.

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	$AD, CE, DE, ABD, ADE, BCE, BDE, ABDE$

Example: BC → E.

sup(BC → E) = sup(BCE) = 3

conf(BC → E) = sup(BCE)
----
sup(BC)
= ¾ = 0.75

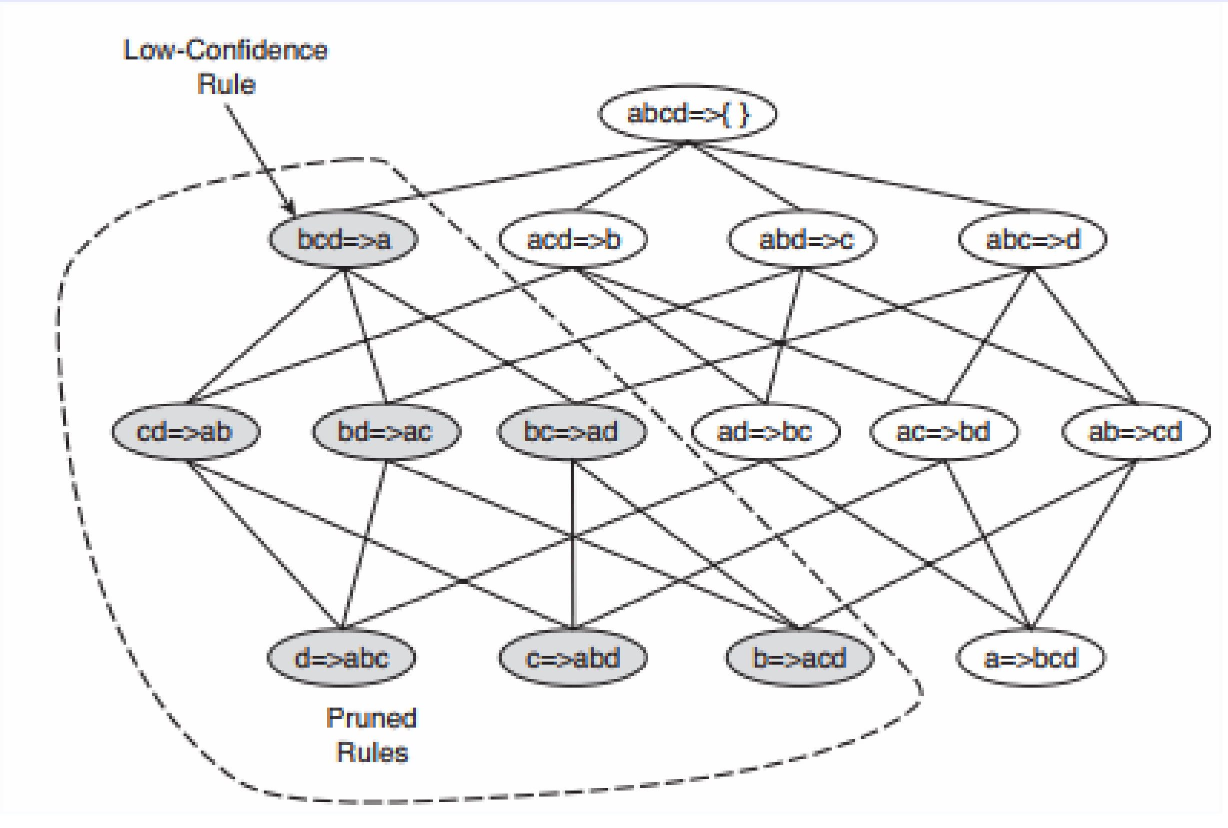
Generating association rules

- Rules are pruned using confidence.

Recall: $c = conf(X \rightarrow Y) = \dfrac{sup(XY)}{sup(X)}$

- Unlike support, confidence does not exhibit the monotone property … but ...
- If we consider rules generated from the same frequent itemset Y, then...
- **Theorem**: If a rule $X \rightarrow Y \setminus X$ does not satisfy the confidence threshold, then any rule $X' \rightarrow Y \setminus X'$, where $X' \subset X$, must not satisfy the confidence threshold as well.

Generating association rules



- **Theorem**: Assume rules are generated from the same frequent itemset. Then, if a rule X→ Y \ X does not satisfy the confidence threshold, then any rule X'→ Y \ X', where X'⊂X, must not satisfy the confidence threshold as well.

  Example: Y={a,b,c,d}, X={b,c,d};
  X → Y \ X =>
  {b,c,d} → {a,b,c,d} \ {b,c,d} =>
  {b,c,d} → {a}
  If {b,c,d} → {a} does not satisfy the confidence threshold, then any X'⊂X must not satisfy the confidence threshold as well.

Generating association rules

- Empirical example for the theorem:

Transaction list: {A,B,C}, {A,B,D}, {A,C}. Below, σ implies support count, and Y – X is the same as Y \ X.
$\sigma(\{A\}) = 3$
$\sigma(\{A, B\}) = 2$
$\sigma(\{A, B, C\}) = 1$
$\sigma(\{A, C\}) = 2$
$\sigma(\{B\}) = 2$

To begin with, Y = {A, B, C}, X = {}, so we have the rule X -> Y, or {} -> {A, B, C}.

Let the antecedent X = {A,B}, then Y - X = {C}.

The confidence of the rule X -> Y - X, or $C(X \rightarrow Y - X) = \sigma(X \cup (Y - X)) / \sigma(X) = \sigma(\{A, B, C\}) / \sigma(\{A, B\}) = 1/2 = 0.5$.

Let X' be a subset of X, ($X' \subset X$). Now, $C(X' \rightarrow Y - X') \leq C(X \rightarrow Y - X)$, or ≤ 0.5 (Theorem 6.2, Tan).

Let X' = {A}, then Y - X' = {B, C}.

The confidence of the rule X' -> Y - X' = $C(X' \rightarrow Y - X') = \sigma(X' \cup (Y - X')) / \sigma(X') = \sigma(\{A, B, C\}) / \sigma(\{A\}) = 1/3 = 0.3$.

If you let X' = {B} and Y - X' = {B, C}, you should also get that Theorem 6.2 holds.

Generating association rules

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

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	$AD, CE, DE, ABD, ADE, BCE, BDE, \boxed{ABDE}$

- Consider the frequent itemset ABDE(3).
- Assume *minconf* = 0.9.
- Recall: $c = conf(X \to Y) = \dfrac{sup(XY)}{sup(X)}$
- Step 1: Let Z = {ABDE}, and enumerate all proper subsets $X \subset Z$:

  A = {ABD(3), ABE(4), ADE(3), BDE(3), AB(4), AD(3), AE(4), BD(4), BE(5), DE(3), A(4), B(6), D(4), E(5)}

Generating association rules

- **A** = {ABD(3), ABE(4), ADE(3), BDE(3), AB(4), AD(3), AE(4), BD(4), BE(5), DE(3), A(4), B(6), D(4), E(5)}

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

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	$AD, CE, DE, ABD, ADE, BCE, BDE, \boxed{ABDE}$

- Step 2: Take the first subset, X = {ABD}, which becomes the antecedent. The consequent is the missing element(s), Y = {E}.
  conf(X → Y) = sup(ABDE)/sup(ABD) = 3/3 = 1.0

Recall: $c = conf(X \to Y) = \frac{sup(XY)}{sup(X)}$

- Calculated confidence ≥ *minconf*. (Recall, minconf=0.9)

Output rule: {ABD} → {E}, conf = 1.0

Generating association rules

- **A =** {ABE(4), ADE(3), BDE(3), AB(4), AD(3), AE(4), BD(4), BE(5), DE(3), A(4), B(6), D(4), E(5)}
- Step 2: Take the first subset, X = {ABE}, which becomes the antecedent. The consequent is the missing element(s), Y = {D}.

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

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	$AD, CE, DE, ABD, ADE, BCE, BDE, \boxed{ABDE}$

Recall: $c = conf(X \rightarrow Y) = \frac{sup(XY)}{sup(X)}$

conf(X → Y) = sup(ABED)/sup(ABE) = 3/4 = 0.75

- Calculated confidence < *minconf*.
  - REJECT rule {ABE} → {D}, conf = 0.75.
- Because we rejected ABE → D, we can remove from **A** all subsets of ABE (the antecedent).

Generating association rules

- **A =** {ABE(4), ADE(3), BDE(3), AB(4), AD(3), AE(4), BD(4), BE(5), DE(3), A(4), B(6), D(4), E(5)}
- Because we rejected ABE $\rightarrow$ D, wee can remove from **A** all subsets of ABE (the antecedent).

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

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	$AD, CE, DE, ABD, ADE, BCE, BDE, ABDE$

Recall: $c = conf(X \rightarrow Y) = \frac{sup(XY)}{sup(X)}$

A = {ADE(3), BDE(3), ~~AB(4)~~, AD(3), ~~AE(4)~~, BD(4), ~~BE(5)~~, DE(3), ~~A(4)~~, ~~B(6)~~, D(4), ~~E(5)~~}

A = {ADE(3), BDE(3), AD(3), BD(4), DE(3), D(4)}

Generating association rules

- **A =** {ADE(3), BDE(3), AD(3), BD(4), DE(3), D(4)}
- Step 2: Take the first subset, X = {ADE}, which becomes the antecedent. The consequent is the missing element(s), Y = {B}.

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

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	$AD, CE, DE, ABD, ADE, BCE, BDE, \boxed{ABDE}$

Recall: $c = conf(X \to Y) = \dfrac{sup(XY)}{sup(X)}$

conf(X $\rightarrow$ Y) = sup(ADEB)/sup(ADE) = 3/3 = 1.0

- Calculated confidence ≥ *minconf*.
  - Output rule {ADE} $\rightarrow$ {B}, conf = 1.0.

Generating association rules

- **A =** {BDE(3), AD(3), BD(4), DE(3), D(4)}
- Step 2: Take the first subset, X = {BDE}, which becomes the antecedent. The consequent is the missing element(s), Y = {A}.

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

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	$AD, CE, DE, ABD, ADE, BCE, BDE, \boxed{ABDE}$

Recall: $c = conf(X \rightarrow Y) = \frac{sup(XY)}{sup(X)}$

conf(X $\rightarrow$ Y) = sup(BDEA)/sup(BDE) = 3/3 = 1.0

- Calculated confidence ≥ *minconf*.
  - Output rule {BDE} $\rightarrow$ {A}, conf = 1.0.

Generating association rules

- **A =** {AD(3), BD(4), DE(3), D(4)}
- Step 2: Take the first subset, X = {AD}, which becomes the antecedent. The consequent is the missing element(s), Y = {BE}.

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

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	$AD, CE, DE, ABD, ADE, BCE, BDE, \boxed{ABDE}$

Recall: $c = conf(X \rightarrow Y) = \dfrac{sup(XY)}{sup(X)}$

conf(X $\rightarrow$ Y) = sup(ADBE)/sup(AD) = 3/3 = 1.0

- Calculated confidence ≥ *minconf*.
  - Output rule {AD} $\rightarrow$ {BE}, conf = 1.0.

Generating association rules

- **A =** {BD(4), DE(3), D(4)}
- Step 2: Take the first subset, X = {BD}, which becomes the antecedent. The consequent is the missing element(s), Y = {AE}.

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

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	$AD, CE, DE, ABD, ADE, BCE, BDE, \boxed{ABDE}$

Recall: $c = conf(X \to Y) = \frac{sup(XY)}{sup(X)}$

conf(X → Y) = sup(BDAE)/sup(BD) = 3/4 = 0.75

- Calculated confidence < *minconf*.
  - REJECT rule {BD} → {AE}, conf = 0.75.
- Because we rejected BD → AE, we can remove from **A** all subsets of BD (the antecedent).

Generating association rules

- **A =** {DE(3)}
- Step 2: Take the remaining subset, X = {DE}, which becomes the antecedent. The consequent is the missing element(s), Y = {AB}.

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

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	$AD, CE, DE, ABD, ADE, BCE, BDE, \boxed{ABDE}$

Recall: $c = conf(X \rightarrow Y) = \frac{sup(XY)}{sup(X)}$

conf(X → Y) = sup(DEAB)/sup(DE) = 3/3 = 1.0

- Calculated confidence ≥ *minconf*.
  - Output rule {DE} → {AB}, conf = 1.0.

Generating association rules

- We perform Steps 1-2 for **all the frequent itemsets** meeting a *minsup* to derive all of the rules!

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

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	$AD, CE, DE, ABD, ADE, BCE, BDE, ABDE$

Recall: $c = conf(X \to Y) = \dfrac{sup(XY)}{sup(X)}$

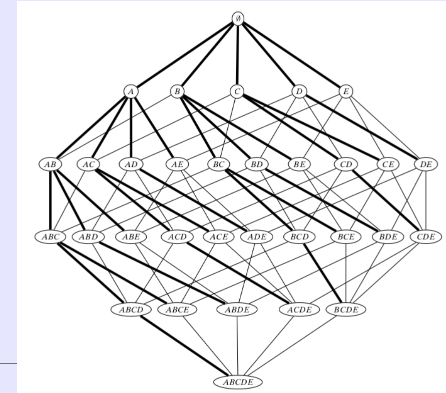
- Question: What about the null rule? That is: {null} → {ABDE}?

How many itemsets and rules?

How many item-sets?

k items generate up to 2^k frequent itemsets.

Number of itemsets for k items =
$$\binom{k}{1} + \binom{k}{2} + \ldots + \binom{k}{k} = 2^k - 1,$$
excluding the null set; with the null set, 2^k

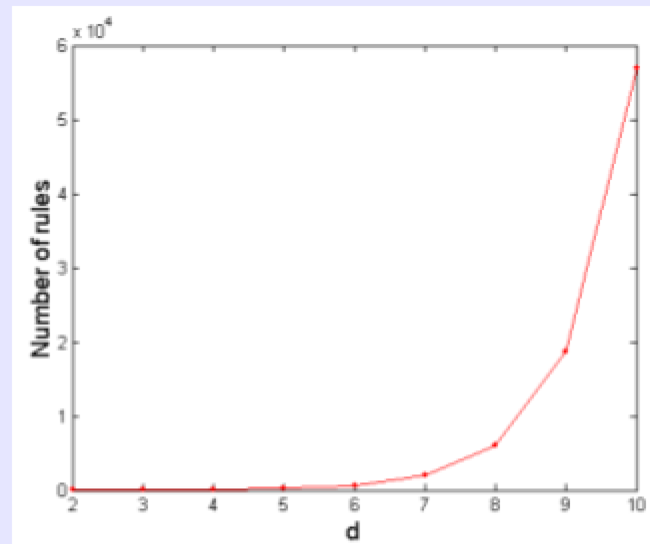


How many rules?

For a 3 item dataset {a,b,c}, the number of candidate rules will be:

d = No. of items
For d = 3, R = 12
For d = 6, R = 602

$$R = \sum_{k=1}^{d-1}\left[\binom{d}{k} \times \sum_{j=1}^{d-k}\binom{d-k}{j}\right] = 3^d - 2^{d+1} + 1$$



The actual rules are shown below:

```
{}    => {C}
{}    => {B}
{}    => {A}
{C}   => {B}
{B}   => {C}
{C}   => {A}
{A}   => {C}
{B}   => {A}
{A}   => {B}
{B,C} => {A}
{A,C} => {B}
{A,B} => {C}
```

Limitations of support and confidence

- What value should we pick for support (*minsup*)?
  - If *minsup* too high (0.20), we may miss interesting low-support items. Such items may correspond to expensive products (jewelry) that are seldom purchased by customers, but whose patterns are interesting to mine for the retailer.
  - If *minsup* is too low (0.001), we get information overload: too many frequent itemsets and too many spurious rules.
    - While {milk} → {diapers} is interesting, {milk} → {eggs} is not.

Limitations of support and confidence

- Drawback of confidence is more subtle.
- Some high confidence rules can be misleading.
- Example:

	$Coffee$	$\overline{Coffee}$	
Tea	150	50	200
$\overline{Tea}$	650	150	800
	800	200	1000

- Evaluate {Tea} → {Coffee}.
  - sup({Tea,Coffee}) = 150; rsup({Tea,Coffee}) = 150/1000 = 0.15 (15%).
  - conf({Tea} → {Coffee}) = sup({Tea,Coffee})/sup({Tea}) = ?????

Limitations of support and confidence

- Drawback of confidence is more subtle.
- Some high confidence rules can be misleading.
- Example:

	$Coffee$	$\overline{Coffee}$	
Tea	150	50	200
$\overline{Tea}$	650	150	800
	800	200	1000

- Evaluate {Tea} → {Coffee}.
  - sup({Tea,Coffee}) = 150; rsup({Tea,Coffee}) = 150/1000 = 0.15 (15%).
  - conf({Tea} → {Coffee}) = sup({Tea,Coffee})/sup({Tea}) = 150/200 = 0.75.
  - The rule {Tea} → {Coffee} appears to be robust. But is it really?

Limitations of support and confidence

- Drawback of confidence is more subtle. Example:

	$Coffee$	$\overline{Coffee}$	
Tea	150	50	200
$\overline{Tea}$	650	150	800
	800	200	1000

rsup = 0.15
conf = 0.75

- Evaluate {Tea} → {Coffee}.
  - Fraction of people who like coffee, regardless of if they like tea = 0.80.
  - Fraction of people who like tea and also like coffee = 0.75.
  - *Thus knowing that a person likes tea <u>decreases</u> her probability of liking coffee!*
    - The rule {Tea} → {Coffee} is, therefore, misleading.

Limitations of support and confidence

- Previous example shows that high-confidence rules can sometimes be misleading.
  - Confidence measure ignores the support of the itemset appearing in the rule consequent.

    $$\text{Recall: } c = conf(X \rightarrow Y) = \frac{sup(XY)}{sup(X)}$$
  - A new metric called **lift** accounts for the consequent and is defined as: $lift(X \rightarrow Y) = \dfrac{conf(X \rightarrow Y)}{rsup(Y)}$
  - By this measure:
    lift({Tea}$\rightarrow${Coffee}) = 0.75/0.80 = 0.94
  - Value of lift close to 1 implies that the support of the rule is expected; we look for values > 1 (above expectation) and << 1 (below expectation).
  - The rule {Tea} -> {Coffee} had a high confidence, but this was an aberration until we saw the lift associated with that rule. Lift normalizes the confidence of the rule using the support of the consequent (coffee). If the support of the consequent is high, i.e., the consequent appears many times in the *tid*set, then the lift will be low.

Evaluating association rules

- Many to choose from!
- For now, focus on support, confidence, and lift.

#	Measure	Formula
1	ϕ-coefficient	$\frac{P(A,B)-P(A)P(B)}{\sqrt{P(A)P(B)(1-P(A))(1-P(B))}}$
2	Goodman-Kruskal's (λ)	$\frac{\sum_j \max_k P(A_j,B_k)+\sum_k \max_j P(A_j,B_k)-\max_j P(A_j)-\max_k P(B_k)}{2-\max_j P(A_j)-\max_k P(B_k)}$
3	Odds ratio (α)	$\frac{P(A,B)P(\overline{A},\overline{B})}{P(A,\overline{B})P(\overline{A},B)}$
4	Yule's Q	$\frac{P(A,B)P(\overline{AB})-P(A,\overline{B})P(\overline{A},B)}{P(A,B)P(\overline{AB})+P(A,\overline{B})P(\overline{A},B)} = \frac{\alpha-1}{\alpha+1}$
5	Yule's Y	$\frac{\sqrt{P(A,B)P(\overline{AB})}-\sqrt{P(A,\overline{B})P(\overline{A},B)}}{\sqrt{P(A,B)P(\overline{AB})}+\sqrt{P(A,\overline{B})P(\overline{A},B)}} = \frac{\sqrt{\alpha}-1}{\sqrt{\alpha}+1}$
6	Kappa (κ)	$\frac{P(A,B)+P(\overline{A},\overline{B})-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A)P(B)-P(\overline{A})P(\overline{B})}$
7	Mutual Information (M)	$\frac{\sum_i \sum_j P(A_i,B_j)\log\frac{P(A_i,B_j)}{P(A_i)P(B_j)}}{\min(-\sum_i P(A_i)\log P(A_i), -\sum_j P(B_j)\log P(B_j))}$
8	J-Measure (J)	$\max\Big(P(A,B)\log(\frac{P(B\|A)}{P(B)}) + P(A\overline{B})\log(\frac{P(\overline{B}\|A)}{P(\overline{B})}),$ $P(A,B)\log(\frac{P(A\|B)}{P(A)}) + P(\overline{A}B)\log(\frac{P(\overline{A}\|B)}{P(\overline{A})})\Big)$
9	Gini index (G)	$\max\Big(P(A)[P(B\|A)^2+P(\overline{B}\|A)^2]+P(\overline{A})[P(B\|\overline{A})^2+P(\overline{B}\|\overline{A})^2]$ $-P(B)^2-P(\overline{B})^2,$ $P(B)[P(A\|B)^2+P(\overline{A}\|B)^2]+P(\overline{B})[P(A\|\overline{B})^2+P(\overline{A}\|\overline{B})^2]$ $-P(A)^2-P(\overline{A})^2\Big)$
10	Support (s)	$P(A,B)$
11	Confidence (c)	$\max(P(B\|A), P(A\|B))$
12	Laplace (L)	$\max\Big(\frac{NP(A,B)+1}{NP(A)+2}, \frac{NP(A,B)+1}{NP(B)+2}\Big)$
13	Conviction (V)	$\max\Big(\frac{P(A)P(\overline{B})}{P(A\overline{B})}, \frac{P(B)P(\overline{A})}{P(B\overline{A})}\Big)$
14	Interest (I)	$\frac{P(A,B)}{P(A)P(B)}$
15	cosine (IS)	$\frac{P(A,B)}{\sqrt{P(A)P(B)}}$
16	Piatetsky-Shapiro's (PS)	$P(A,B)-P(A)P(B)$
17	Certainty factor (F)	$\max\Big(\frac{P(B\|A)-P(B)}{1-P(B)}, \frac{P(A\|B)-P(A)}{1-P(A)}\Big)$
18	Added Value (AV)	$\max(P(B\|A)-P(B), P(A\|B)-P(A))$
19	Collective strength (S)	$\frac{P(A,B)+P(\overline{AB})}{P(A)P(B)+P(\overline{A})P(\overline{B})} \times \frac{1-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A,B)-P(\overline{AB})}$
20	Jaccard (ζ)	$\frac{P(A,B)}{P(A)+P(B)-P(A,B)}$
21	Klosgen (K)	$\sqrt{P(A,B)}\max(P(B\|A)-P(B), P(A\|B)-P(A))$