Data Mining: Association Analysis: Part III - extra

Laura Brown

Some slides adapted from G. Piatetsky-Shapiro;
Han, Kamber, & Pei; Tan, Steinbach, & Kumar; A. Wasilewska

Terminology

- Every subset of a frequent set is frequent!
  - If {A, B} is frequent. Each occurrence of A, B includes both A and B, then both A and B alone must also be frequent
- A long pattern (itemsets) contains a combinatorial number of sub-patterns (itemsets)
  - A frequent set with 100 items contains

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

- Solution: look at *closed patterns* and *max-patterns*

Maximal Frequent Itemsets

- Given a binary database $\boldsymbol{D} \subseteq \mathcal{T} \times I$**,** over the *tids* and *items*, let $\mathcal{F}$ be the set of all frequent itemsets,
$$\mathcal{F} = \{X \mid X \subseteq I \; and \; \sup(X) \geq minsup \}$$

- A frequent itemset $X \in \mathcal{F}$ is called maximal if it has no frequent supersets. Let $\mathcal{M}$ be the set of all maximal frequent itemsets,
$$\mathcal{M} = \{X \mid X \in \mathcal{F} \;\; and \;\; \nexists \, Y \supset X, s.t. \; Y \in \mathcal{F} \}$$

- The set $\mathcal{M}$ is a condensed representation of the set of all frequent itemset $\mathcal{F}$

Maximal Itemsets Example

Transaction Database, **D**

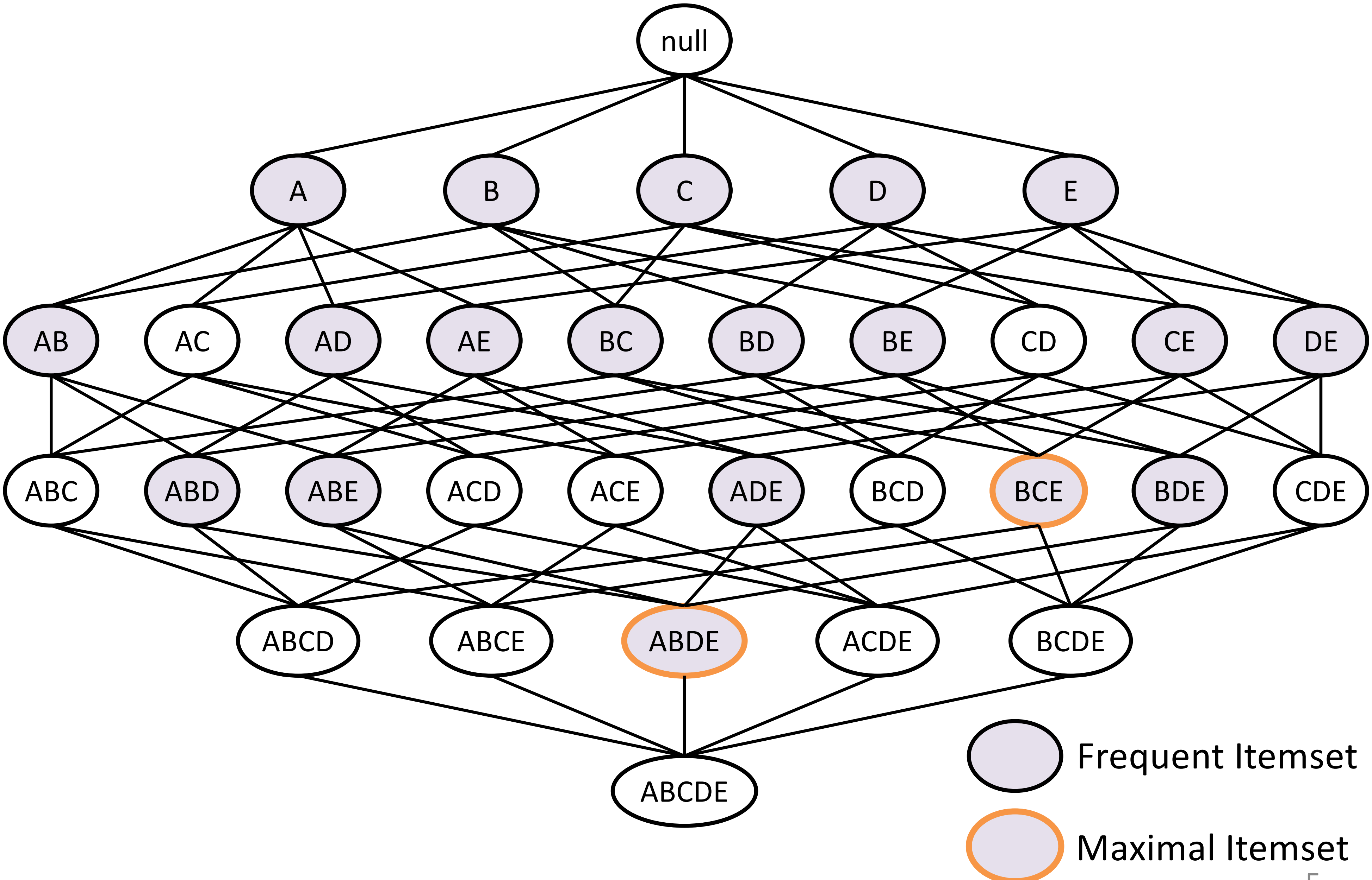
Tid	Itemset
1	$ABDE$
2	BCE
3	$ABDE$
4	$ABCE$
5	$ABCDE$
6	BCD

Frequent Itemsets (*minsup* = 3)

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

- Which are the maximal frequent itemsets?

Frequent and Maximal Itemsets



Maximal Itemsets Example

Transaction Database, **D**

Tid	Itemset
1	$ABDE$
2	BCE
3	$ABDE$
4	$ABCE$
5	$ABCDE$
6	BCD

Frequent Itemsets (*minsup* = 3)

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

- Which are the maximal itemsets?
  - ABDE and BCE

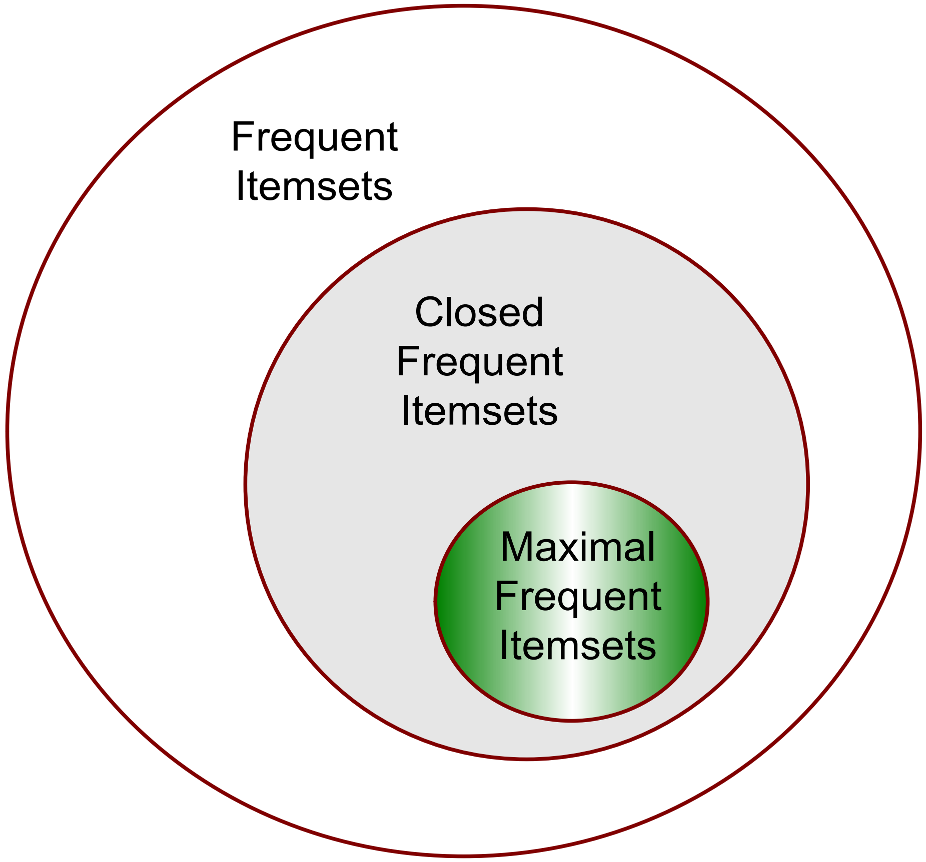
Closed Itemset

- An itemset is closed if there does not exist a proper super-itemset such that it has the same support count in the database
- An itemset is a closed frequent itemset if it is both closed and frequent
- The set of all closed frequent itemsets is defined as

$$\mathcal{C} = \{\, \mathrm{X} \mid \mathrm{X} \in \mathcal{F} \;\; and \;\; \nexists Y \supset X \; s.t. \sup(X) = \sup(Y) \,\}$$

X is closed if all supersets of *X* have strictly less support, that is, *sup*(*X*) > *sup*(*Y*), for all $Y \supset X$

Maximal vs. Closed Itemsets



Relationship between the set of all, closed, and maximal frequent itemsets

$$\mathcal{M} \subseteq \mathcal{C} \subseteq \mathcal{F}$$

Closed Frequent Itemsets Example

Transaction Database, **D**

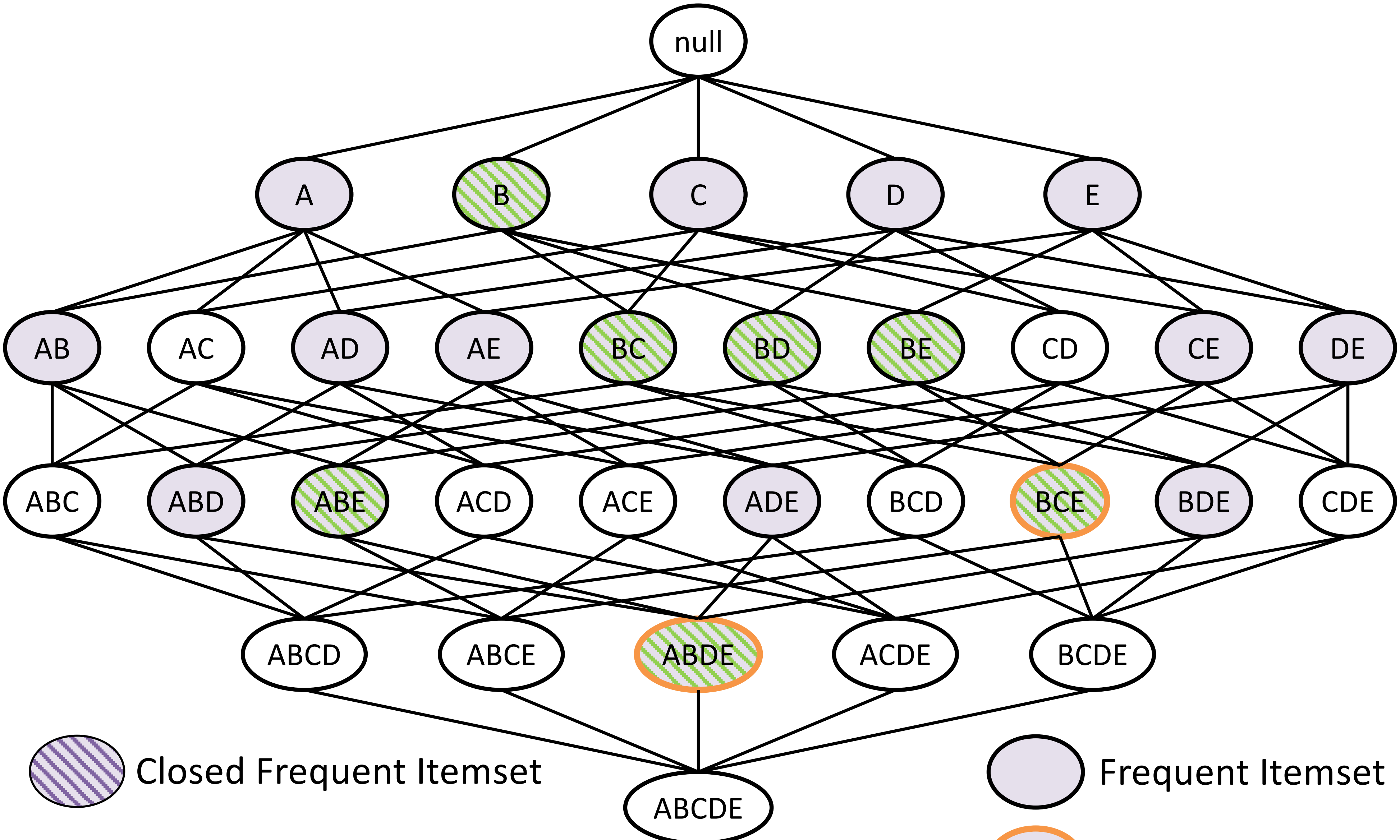
Tid	Itemset
1	$ABDE$
2	BCE
3	$ABDE$
4	$ABCE$
5	$ABCDE$
6	BCD

Frequent Itemsets (*minsup* = 3)

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

- Which are the closed frequent itemsets?

Frequent and Maximal Itemsets



Closed Frequent Itemsets Example

Transaction Database, **D**

Tid	Itemset
1	$ABDE$
2	BCE
3	$ABDE$
4	$ABCE$
5	$ABCDE$
6	BCD

Frequent Itemsets (*minsup* = 3)

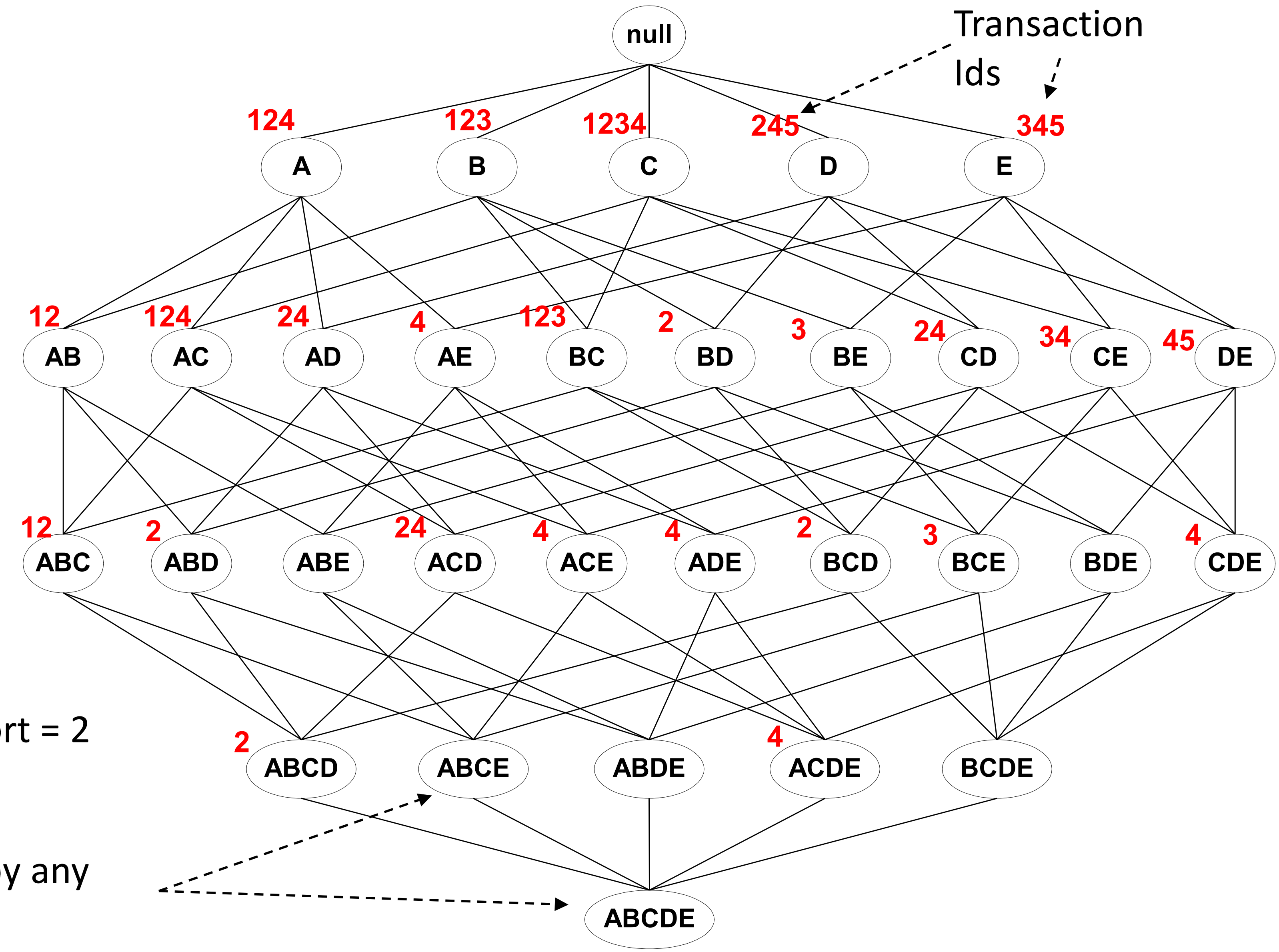
sup	Itemsets
6	B
5	E, BE
4	$A, C, D, AB, AE, BC, BD, ABE$
3	$AD, CE, DE, ABD, ADE, BCE, BDE, ABDE$

- Which are the closed frequent itemsets?
  - B, BE, BC, BD, ABE, BCE, ABDE

MAXIMAL AND CLOSED ITEMSETS: EXAMPLE

Maximal vs. Closed Frequent Itemsets

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



Minimum support = 2

Not supported by any transactions

Solution at End of Slides

MINING MAXIMAL FREQUENT ITEMSETS

Mining Maximal Frequent Itemsets

- Methods to mine frequent itemsets can be extended to mine maximal frequent itemsets by adding maximality checking steps
- A new frequent itemset *X* should be tested with the following maximality checks:
  - **Subset Check:** $\nexists Y \in \mathcal{M}$**,** such that $X \subset Y$**.** If such a *Y* exists, then *X* is not maximal. Otherwise, we add *X* to $\mathcal{M}$, as a potential maximal itemset
  - **Superset Check:** $\nexists Y \in \mathcal{M}$, such that $Y \subset X$. If such a *Y* exists, then *Y* can not be maximal and we must remove it from $\mathcal{M}$.

Algorithms

Mines Maximal Frequent itemsets

- GenMax – based on the dECLAT it uses diffset *tidset* intersection in its computation
    - Mines maximal frequent itemsets

Mines Closed Frequent itemsets

- Mining for closed frequent itemsets requires that we perform closure checks (can be very expensive)
- Charm – is a vertical *tidset* intersection method that efficient checks for closure using 3 properties.

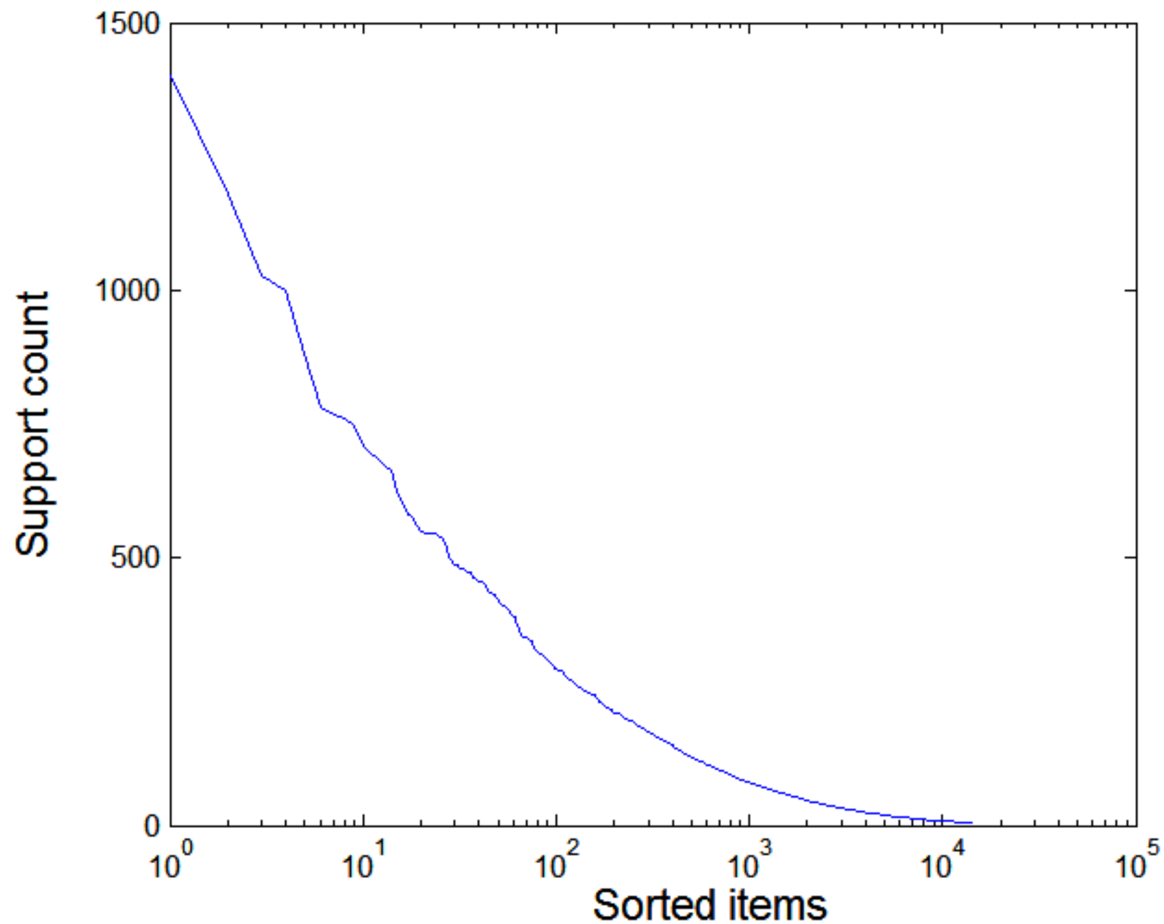
- More detail on these methods in Ch. 9 of book

SUPPORT DISTRIBUTION

Effect of Support Distribution

- Many real data sets have skewed support distributions

Support distribution of a retail data set



Effect of Support Distribution

- How to select appropriate *minsup* threshold?
  - If *minsup* is too high, could miss itemset involving interesting rare items (e.g., expensive products)
  - If *minsup* is too low, computationally expensive and number of itemsets identified grows

- Use of a single minimum support threshold may not be effective

Multiple Minimum Support

- How to apply multiple minimum supports?
  - MS(i) : minimum support for item i
  - Ex. MS(Milk) = 5% MS(Coke) = 3%
    MS(Broccoli) = 0.1% MS(Salmon) = 0.5%
  - MS({Milk, Broccoli}) = min(MS(Milk), MS(Broccoli))
    = 0.1%
- Challenge: Support is no longer anti-monotone
  - Suppose: Support(Milk, Coke) = 1.5%
    Support(Milk, Coke, Broccoli) = 0.5%
  - {Milk, Coke} is infrequent, but {Milk, Coke, Broccoli} is frequent

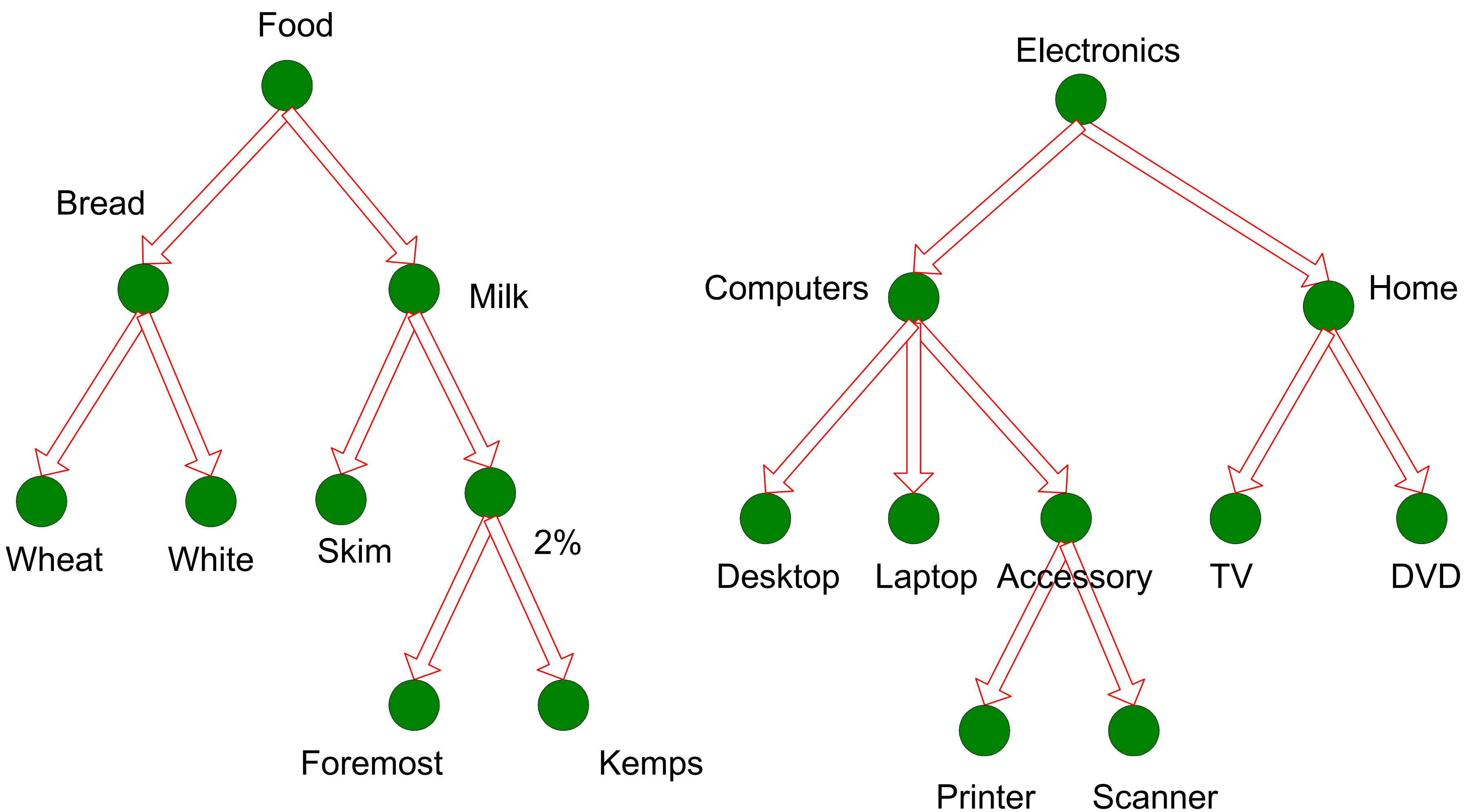
Multiple Minimum Support

- Order items according to their minimum support (ascending order)
  - Ex. Broccoli, Salmon, Coke, Milk
- Modify Apriori algorithm to support MMS
  - At 1-itemsets create, F_1, set of items that pass minimum support levels
  - C_2 is created from join of F_1 rather than L_1
  - Pruning must also be modified to account for itemized support
- Reference: Liu 1999

Advanced Pattern Matching

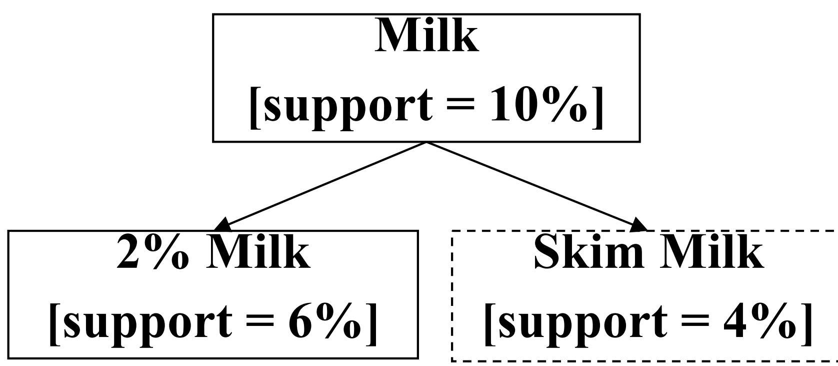
- Multi-level association rules
- Multi-dimensional association rules
- Quantitative association rules
- Mining Rare and Negative Patterns
- Constraint-based Pattern Mining
- Mining Colossal Patterns
- Application to Text Mining

Multi-level Association Rules



Multi-level Association Rules

- Items often naturally form hierarchy
- Flexible support thresholds
  - rules at lower levels may not have enough support to appear in frequent itemsets
  - rules at lower levels of the hierarchy may be overly specific



Multi-level Mining (Agrawal & Srikant, VLDB ’95, Han & Fu, VLDB ‘95)

Multi-level Support and Confidence

- How do the support and confidence vary as we traverse the concept hierarchy?
  - If X is the parent item for both X1 and X2, then $\sigma(X) \leq \sigma(X1) + \sigma(X2)$

  - If $\sigma(X1 \cup Y1) \geq minsup$,
    and X is a parent of X1, Y is a parent of Y1
    then $\sigma(X \cup Y1) \geq minsup$, $\sigma(X1 \cup Y) \geq minsup$,
    $\sigma(X \cup Y) \geq minsup$

  - If $conf(X1 \rightarrow Y1) \geq minconf$,
    then $conf(X1 \rightarrow Y) \geq minconf$

Flexible Support and Redundancy filtering

- Flexible min-support thresholds: some items are more valuable by less frequent
  - use non-uniform, group-based min-support
  - e.g., {diamond, watch, camera}: 0.05%; {bread, milk, soda}: 5%, …
- Redundancy Filtering: some rules may be redundant due to “ancestor” relationships between items
  - milk -> wheat bread [support=8%, conf=70%]
  - 2% milk -> wheat bread [support=2%, conf=72%]

  The first rule is an ancestor of the second rule

  - A rule is redundant if its support is close to the “expected” value, based on the rule’s ancestor

Multi-level Association Rules

- Approach 1:
  - extend current association rule formulation by augmenting each transaction with higher level items

    Original Trans.: {skim milk, wheat bread}
    Augmented Trans.:
    {skim milk, wheat bread, milk, bread, food}
- Issues:
  - items that reside at higher levels have much higher support counts
  - increased dimensionality of the data

Multi-level Association Rules

- Approach 2
  - generate frequent patterns at highest level first
  - then, generate frequent patterns at the next highest level, and so on

- Issues
  - I/O requirements will increase dramatically because there are more passes over the data
  - may miss cross-level association patterns

Multi-Dimensional Association Rules

- Single-dimensional rules
  buys(X, “milk”) -> buys(X, “bread”)
- Multi-dimensional rules
  - inter-dimensional association (no repeated predicates)

    age(X, “19-25”) $\wedge$ occupation(X, “student”) -> buys(X, “coke”)
  - hybrid-dimensional association (repeated predicates)

    age(X, “19-25”) $\wedge$ buys(X, “popcorn”) -> buys(X, “coke”)
- Categorical Attributes: finite number of values
  transform attribute or data cube approach
- Quantitative Attributes: numeric values
  discretization, clustering, or other methods

Continuous and Categorical Attributes

- How to apply association analysis formulation to non-asymmetric binary variables?

Session Id	Country	Session Length (sec)	Number of Web Pages viewed	Gender	Browser Type	Buy
1	USA	982	8	Male	IE	No
2	China	811	10	Female	Netscape	No
3	USA	2125	45	Female	Mozilla	Yes
4	Germany	596	4	Male	IE	Yes
5	Australia	123	9	Male	Mozilla	No
…	…	…	…	…	…	…

- Example of Association Rule:

$$\text{No. of Pages} \in [5,10] \wedge \text{Browser=Mozilla} \rightarrow \text{Buy=No}$$

Handling Categorical Attributes

- Transform categorical attribute into asymmetric binary variables

- Introduce a new “item” for each distinct attribute-value pair
  - Example: replace Browser Type attribute with
    - Browser Type = IE
    - Browser Type = Mozilla
    - Browser Type = Netscape

Handling Categorical Attributes

Potential Issues

- What is attribute has many possible values
  - Example: attribute country has more than 200 possible values
  - Many of the attribute values may have very low support

- What if distribution of attribute values is highly skewed
  - Example: 95% of the visitors have Buy = No
  - Most of the items will be associated with (Buy = No) item

Handling Continuous Attributes

- Different kinds of rules:

$$Age \in [21,35] \wedge Salary \in [70K,120K] \rightarrow Buy$$

$$Salary \in [70K,120K] \wedge Buy \rightarrow Age: \mu = 28, \sigma = 4$$

- Different methods:
  - discretization-based
  - statistics-based
  - non-discretization-based
    - minApriori

Handling Continuous Attributes - Discretize

- Discretization methods
  - unsupervised
    - equal-width binning
    - equal-depth binning
    - clustering – Yang & Miller, SIGMOD97
  - supervised
  - statistical inference – Aumann & Lindell, KDD99
- Discretization types
  - static – data-cube methods
  - dynamic – Agrawal & Srikant, SIGMOD96
- Discretization issues
  - size of discretized intervals affect support & confidence
  - execution time

Statistics-based Methods

- Example Rule:

$$Browser = Mozilla \land Buy = Yes \rightarrow Age : \mu = 23$$

- Rule consequent consists of a continuous variable, characterized by their statistics
  - mean, median, standard deviation, etc.
- Approach
  - withhold target variable from the rest of the data
  - apply existing frequent itemset generation on the rest of the data
  - for each frequent itemset, compute the descriptive statistics for the corresponding target variable
    - frequent itemset becomes a rule by introducing the target variable as rule consequent
  - apply statistical test to determine interestingness of the rule

Statistics-based Methods

- How to determine whether an association rule interesting?
  - compare the statistics for segment of population covered by the rule vs segment of population not covered by the rule:

$$A \Rightarrow B : \mu \qquad \text{versus} \qquad \bar{A} \Rightarrow B : \mu$$

  - statistical hypothesis testing:

$$Z = \frac{\mu' - \mu - \Delta}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

    - null hypothesis $H0 : \mu' = \mu + \Delta$
    - alternative hypothesis $H1 : \mu' > \mu + \Delta$
    - Z has zero mean and variance 1 under null hypothesis

Statistics-based Methods

- Example Rule:

$$Browser = Mozilla \wedge Buy = Yes \rightarrow Age : \mu = 23$$

  - rule is interesting if difference between μ and μ’ is greater than 5 years (Δ=5)
  - for r, suppose n1=50, s1=3.5
  - for r’ (complement), n2=250, s2=6.5

$$Z = \frac{\mu' - \mu - \Delta}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} = \frac{30 - 23 - 5}{\sqrt{\frac{3.5^2}{50} + \frac{6.5^2}{250}}} = 3.11$$

  - for 1-sided test at 95% confidence level, critical Z-value for rejecting null hypothesis is 1.64
  - since Z is greater than 1.64, r is an interesting rule

Statistics-based Methods

- Issues:
  - multiple comparisons
  - if 10000 rules, then 0.05*10000=500 will seem interesting by chance alone!
  - may use corrections to try to avoid false discovery
    - Bonferoni correction
    - FDR

Rare and Negative Patterns

- Rare patterns: very low support but interesting
  - e.g. buying Rolex watches
  - Mining: setting individual-based or special group-based support threshold for valuable items
- Negative patterns
  - it is unlikely that someone buys both a Ford F150 and Toyota Prius together, there is a likely negatively correlated pattern
  - negatively correlated patterns that are infrequent tend to be more interesting than those that are frequent

Negative Correlated Patterns

- Definition 1 (support-based)
  - If itemsets X and Y are both frequent but rarely occur together, i.e.,
    $sup(X \cup Y) < sup(X) * sup(Y)$
  - then X and Y are negatively correlated
- Problem – support-based definition is not null invariant

Negative Correlated Patterns

- Definition 2 (negative itemset-based)
  - X is a negative itemset if (1) $X = \bar{A} \cup B$ where B is a set of positive items, and $\bar{A}$ is a set of negative items, $|\bar{A}| \geq 1$, and (2) s(X) ≥ μ
  - Itemset X is negatively correlated if

$$s(X) < \prod_{i=1}^{k} s(x_i), \text{ where } x_i \in X, \; s(x_i) \text{ support of } x_i$$

- Problem - similar null-invariant issue

Negative Correlated Patterns

- Definition 3 (Kulzynski measure-based)
  - If itemsets X and Y are frequent, but
    $(P(X \mid Y) + P(Y \mid X)) / 2 < \varepsilon$
    where ε is negative pattern threshold, then X and Y are negatively correlated

Constraint-based Mining

- Finding all the patterns in a database autonomously? — unrealistic!
  - The patterns could be too many but not focused!
- Data mining should be an interactive process
  - User directs what to be mined using a data mining query language (or a graphical user interface)
- Constraint-based mining
  - User flexibility: provides constraints on what to be mined
  - Optimization: explores such constraints for efficient mining — constraint-based mining: constraint-pushing, similar to push selection first in DB query processing
  - Note: still find all the answers satisfying constraints, not finding some answers in “heuristic search”

Different Constraints

- Knowledge type constraint:
  - classification, association, etc.
- Data constraint — using SQL-like queries
  - find product pairs sold together in stores in Chicago this year
- Dimension/level constraint
  - in relevance to region, price, brand, customer category
- Rule (or pattern) constraint
  - small sales (price < \$10) triggers big sales (sum > \$200)
- Interestingness constraint
  - strong rules: min_support > 3%, min_confidence > 60%

Constraint Properties

- Pattern space pruning constraints
  - Anti-monotonic: if constraint c is violated, mining may be terminated
  - Monotonic: If c is satisfied, no need to check c again
  - Succinct: c must be satisfied, so one can start with sets satisfying c
  - Convertible: c is not monotonic nor anti-monotonic, but it can be converted into them if items in the transaction can be properly ordered
- Data space pruning constraints
  - data succinct: data space can be pruned at the initial pattern mining process
  - data anti-monotonic: if a transaction t does not satisfy c, t can be pruned from further mining

Anti-Monotonic Constraints

- A constraint c is anti-monotone if the super pattern satisfies c, all of its sub-patterns do so to
  - that is, if an itemset S violates the contraint, so does any of its supersets
- Ex 1. sum(S.price) ≤ ε, anti-monotone
- Ex 2. range(S.profit) > 15, anti-monotone
- Ex 3. sum(S.price) ≥ ε, not anti-monotone
- Ex 4. support count, anti-monotone used in Apriori

TDB (min_sup=2)

TID	Transaction
10	a, b, c, d, f
20	b, c, d, f, g, h
30	a, c, d, e, f
40	c, e, f, g

Item	Profit
a	40
b	0
c	-20
d	10
e	-30
f	30
g	20
h	-10

Monotone Constraints

- A constraint c is monotone if the pattern satisfies c, we do not need to check c in subsequent mining
  - that is, if an itemset S satisifies the constraint, so does any of its supersets
- Ex 1. sum(S.Price) ≥ ε, monotone
- Ex 2. min(S.Price) ≤ ε, monotone
- Ex 3. range(S.Profit) ≤ ε, not monotone

TDB (min_sup=2)

TID	Transaction
10	a, b, c, d, f
20	b, c, d, f, g, h
30	a, c, d, e, f
40	c, e, f, g

Item	Profit
a	40
b	0
c	-20
d	10
e	-30
f	30
g	20
h	-10

Succinct Constraints

- Given A, the set of items satisfying a succinctness constraint c, then any set S satisfying c is based on A, i.e., S contains a subset belonging to A
- Idea: without looking at database, determine whether an itemset S satisfies constraint c based on a selection of items
  - min(S.Price) $\leq \varepsilon$, succinct
  - sum(S.Price) $\leq \varepsilon$, not succinct

Apriori + Constraint

Database D

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D →

C_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

L_1

itemset	sup.
{1}	2
{2}	3
{3}	3
~~{5}~~	~~3~~

C_2

itemset
{1 2}
{1 3}
~~{1 5}~~
{2 3}
~~{2 5}~~
~~{3 5}~~

Scan D

C_2

itemset	sup
{1 2}	1
{1 3}	2
~~{1 5}~~	~~1~~
~~{2 3}~~	~~2~~
~~{2 5}~~	~~3~~
~~{3 5}~~	~~2~~

L_2

itemset	sup
{1 3}	2
~~{2 3}~~	~~2~~
~~{2 5}~~	~~3~~
~~{3 5}~~	~~2~~

C_3

itemset
~~{2 3 5}~~

Scan D →

L_3

itemset	sup
~~{2 3 5}~~	~~2~~

Constraint:

Sum{S.price} < 5

Apriori + Constraint

Database D

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D

C_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

L_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3

C_2

itemset
{1 2}
{1 3}
{1 5}
~~{2 3}~~
~~{2 5}~~
~~{3 5}~~

not immediately to be used

Scan D

C_2

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
~~{2 3}~~	~~2~~
~~{2 5}~~	~~3~~
~~{3 5}~~	~~2~~

L_2

itemset	sup
{1 3}	2
~~{2 3}~~	~~2~~
~~{2 5}~~	~~3~~
~~{3 5}~~	~~2~~

C_3

itemset
~~{2 3 5}~~

Scan D

L_3

itemset	sup
~~{2 3 5}~~	~~2~~

Constraint:

min{S.price } <= 1

Mining Colossal Patterns

- Many algorithms exist, but can colossal patterns be mined? – 50 to 100 items, NO!
- Why not? – curse of downward closure of frequent patterns
  - any sub-pattern of a frequent pattern is frequent
  - using either breadth-first (Apriori) or depth-first (Fpgrowth), too many patterns
- Many applications need solution to this problem
  - no hope for a complete solution

Pattern-Fusion Strategy

- Pattern-Fusion traverses the tree in a bounded-breadth way
  - always pushes down a frontier of a bounded-size candidate pool
  - only a fixed number of patterns in the current candidate pool will be used as the starting nodes to go down in the pattern tree – thus avoids exponential search
- Pattern-Fusion identifies shortcuts whenever possible
  - pattern growth is not performed by single-item addition but by leaps: agglomeration of multiple patterns in the pool
  - the search gets directed down the tree more rapidly towards colossal patterns

Robustness of Colossal Patterns

- Core patterns
  - intuitively, for a frequent pattern A, a sub-pattern B is a τ-core pattern of A if B shares a similar support set with A, where τ is called the core ratio
- Robustness of colossal patterns
  - a colossal pattern is robust in the sense that it tends to have much more core patterns than small patterns

Example: Core Patterns

- A colossal pattern has far more core patterns than a small-sized pattern
- A colossal pattern has far more core descendants of a smaller size c
- A random draw from a complete set of pattern of size c would more likely to pick a core descendant of a colossal pattern
- A colossal pattern can be generated by merging a set of core patterns

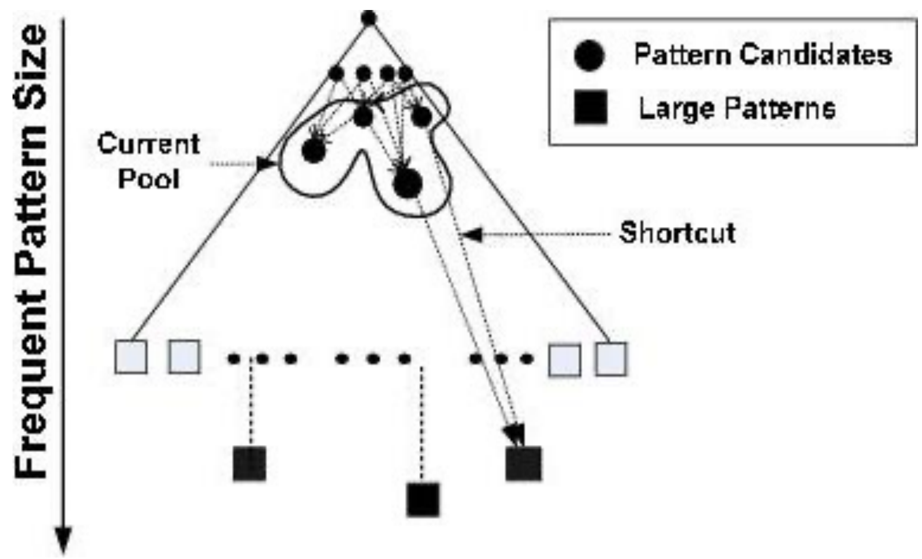
Transaction (# of Ts)	Core Patterns ($\tau = 0.5$)
(abe) (100)	(abe), (ab), (be), (ae), (e)
(bcf) (100)	(bcf), (bc), (bf)
(acf) (100)	(acf), (ac), (af)
(abcef) (100)	(ab), (ac), (af), (ae), (bc), (bf), (be) (ce), (fe), (e), (abc), (abf), (abe), (ace), (acf), (afe), (bcf), (bce), (bfe), (cfe), (abcf), (abce), (bcfe), (acfe), (abfe), (abcef)

Pattern-Fusion Algorithm

- Initialize: use an existing algorithm to mine all frequent patterns up to a small size, e.g., 3
- Iteration (Iterative Pattern Fusion)
  - At each iteration, k seed patterns are randomly selected from the current pool
  - For each seed pattern, find all patterns within a bounding ball centered at the seed pattern
  - All of these patterns are fused together to generate a set of super-patterns. All the super-patterns form the pool for the next iteration
- Termination: when the current pool contains no more than K patterns at the beginning of an iteration

Why is Pattern-Fusion Efficient?

- A bounded-breadth pattern tree traversal
  - avoids explosion in mining mid-sized patterns
  - randomness helps to stay on right path
- Ability to identify shortcuts and take leaps
  - fuse small patterns together



Text Mining

- Consider a document-term matrix

TID	W1	W2	W3	W4	W5
D1	2	2	0	0	1
D2	0	0	1	2	2
D3	2	3	0	0	0
D4	0	0	1	0	1
D5	1	1	1	0	2

- Example:
  - W1 and W2 appear in the same documents

Han et al., Min-Apriori

Text Mining

- Data contains only continuous attributes of the same type
  - frequency of words in a document
- Potential solution:
  - convert into 0/1 matrix, then applying existing algorithms
    - lose word frequency information
  - discretization does not apply as users want association among words not ranges of words

Text Mining

- How to determine the support of a word?
  - sum up frequency, support count will be greater than total number of documents
  - normalize the word vectors – use L1 norm
  - each word has support equals to 1.0

TID	W1	W2	W3	W4	W5
D1	2	2	0	0	1
D2	0	0	1	2	2
D3	2	3	0	0	0
D4	0	0	1	0	1
D5	1	1	1	0	2

Normalize →

TID	W1	W2	W3	W4	W5
D1	0.40	0.33	0.00	0.00	0.17
D2	0.00	0.00	0.33	1.00	0.33
D3	0.40	0.50	0.00	0.00	0.00
D4	0.00	0.00	0.33	0.00	0.17
D5	0.20	0.17	0.33	0.00	0.33

Min-Apriori

- New definition of support

$$\sup(C) = \sum_{i \in T} \min_{j \in C} D(i, j)$$

TID	W1	W2	W3	W4	W5
D1	0.40	0.33	0.00	0.00	0.17
D2	0.00	0.00	0.33	1.00	0.33
D3	0.40	0.50	0.00	0.00	0.00
D4	0.00	0.00	0.33	0.00	0.17
D5	0.20	0.17	0.33	0.00	0.33

Example:

Sup(W1,W2,W3)

= 0 + 0 + 0 + 0 + 0.17

= 0.17

Anti-monotone property of support

TID	W1	W2	W3	W4	W5
D1	0.40	0.33	0.00	0.00	0.17
D2	0.00	0.00	0.33	1.00	0.33
D3	0.40	0.50	0.00	0.00	0.00
D4	0.00	0.00	0.33	0.00	0.17
D5	0.20	0.17	0.33	0.00	0.33

Example:

Sup(W1) = 0.4 + 0 + 0.4 + 0 + 0.2 = 1

Sup(W1, W2) = 0.33 + 0 + 0.4 + 0 + 0.17 = 0.9

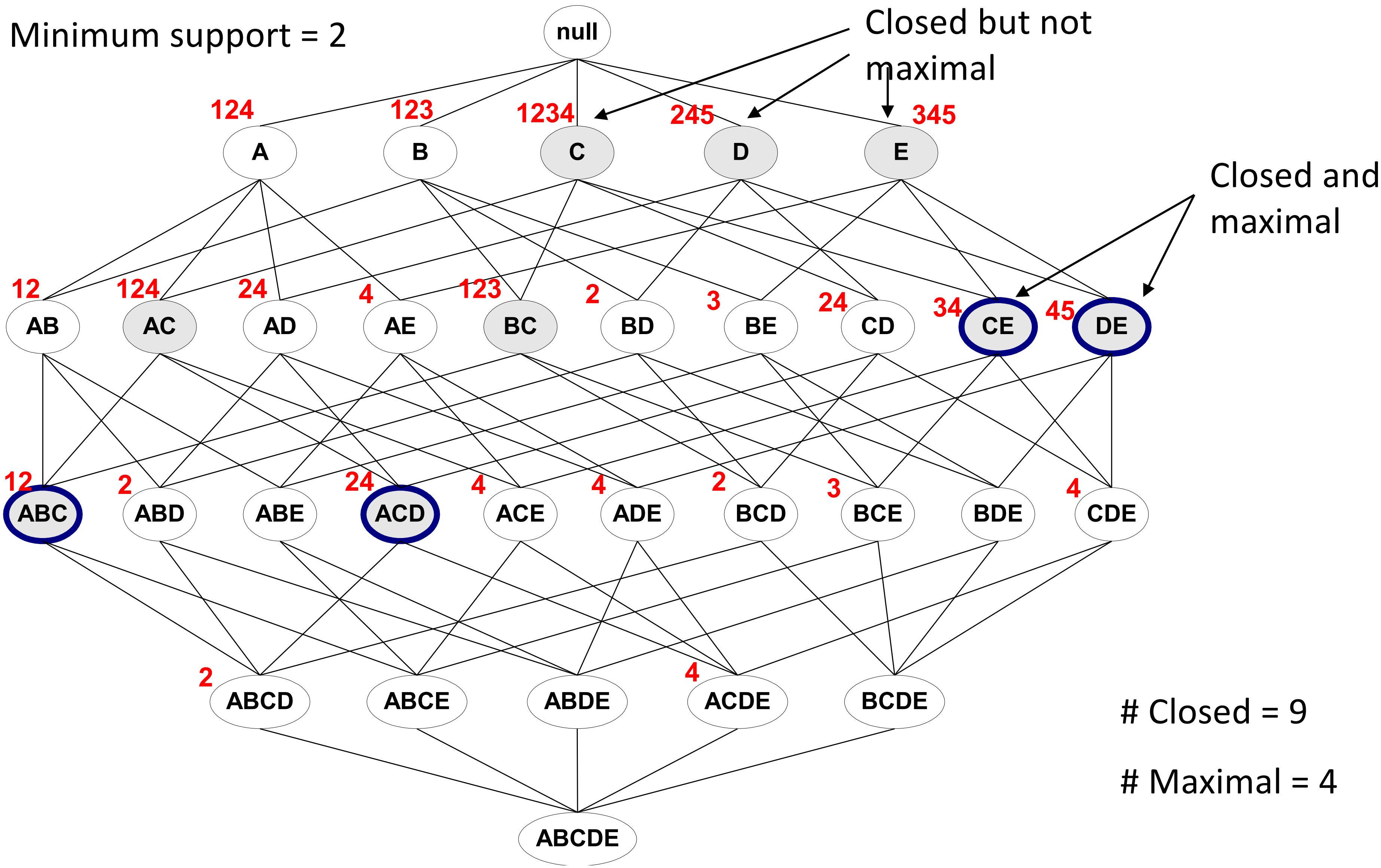
Sup(W1, W2, W3) = 0 + 0 + 0 + 0 + 0.17 = 0.17

Summary

- Multi-level association rules
- Multi-dimensional association rules
- Quantitative association rules
- Mining Rare and Negative Patterns
- Constraint-based Pattern Mining
- Mining Colossal Patterns
- Application to Text Mining

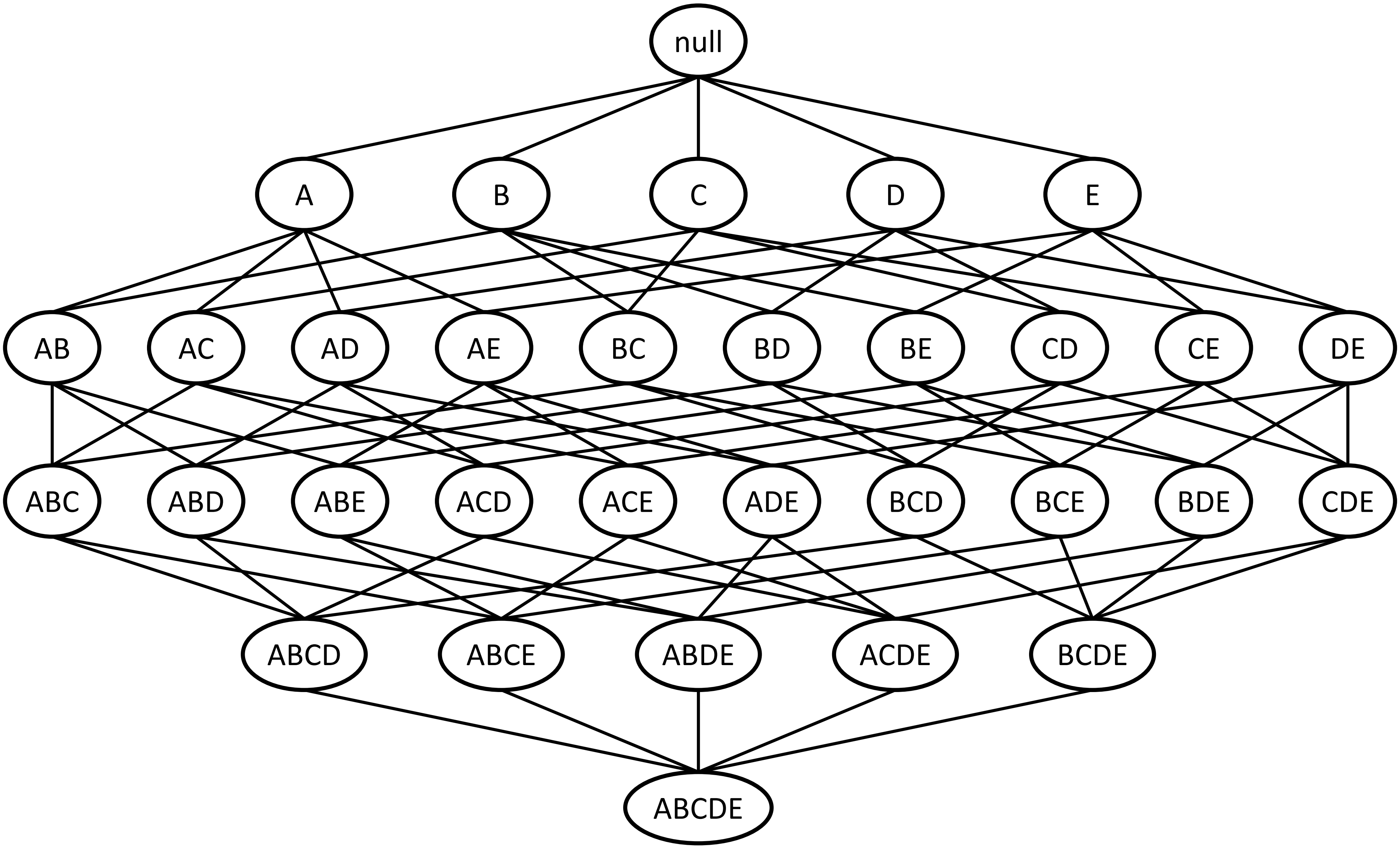
SOLUTION TO EXAMPLES

Maximal vs. Closed Frequent Itemsets: Soln



Closed = 9

Maximal = 4


null
A
B
C
D
E
AB
AC
AD
AE
BC
BD
BE
CD
CE
DE
ABC
ABD
ABE
ACD
ACE
ADE
BCD
BCE
BDE
CDE
ABCD
ABCE
ABDE
ACDE
BCDE
ABCDE