ASSOCIATION MINING

ISE/DSA 5103

CHARLES NICHOLSON, PH.D.

ASSOCIATION RULES

- Study of “what goes with what”
  - Customers who bought X also bought Y
  - What symptoms go with what diagnosis
- Transaction-based or event-based
- Also called *market basket analysis* and *affinity analysis*, *frequent pattern mining*
- Originated with study of customer transactions databases to determine associations among items purchased

WHAT IS FREQUENT PATTERN ANALYSIS?

- Frequent pattern: a pattern (a set of items, subsequences, substructures, etc.) that occurs frequently in a data set
- Motivation: Finding inherent regularities in data
  - What products were often purchased together?— Beer and diapers?!
  - What are the subsequent purchases after buying a PC?
  - What kinds of DNA are sensitive to this new drug?
  - Can we automatically classify web documents?

APPLICATIONS

- **Market Basket Analysis:** given a database of customer transactions, where each transaction is a set of items the goal is to find groups of items which are frequently purchased together.
- **Credit Cards/ Banking Services:** each card/account is a transaction containing the set of customer's payments
- **Medical Treatments:** each patient is represented as a transaction containing the ordered set of diseases

Bound Away
Last Train Home

Share your own customer images

List Price: $16.98
Price: **$16.98** and eligible for **FREE Super Saver Shipping** on orders over $25. See details.

Availability: Usually ships within 24 hours

Want it delivered Tomorrow? Order it in the next 4 hours and 9 minutes, and choose **One-Day S** checkout. See details.

41 used & new from $6.99

See more product details

Based on customer purchases, this is the #82 Early Adopter Product in Alternative Rock.

801x612

Buy this title for only $.01 when you get a new Amazon Visa® Card

Apply now and if you're approved instantly, **save $30** off your first purchase, earn **3% rewards**, get a **0% APR,*** and pay **no**

Amazon Visa discount:	$30.00
Applied to this item:	- $16.97
Discount remaining:	$13.03

Find out how
(Don't show again)

Customers who bought this title also bought:

- Time and Water ~ Last Train Home (Why?)
- Cold Roses ~ Ryan Adams & the Cardinals (Why?)
- Tambourine ~ Tift Merritt (Why?)
- Last Train Home ~ Last Train Home (Why?)
- True North ~ Last Train Home (Why?)
- Universal United House of Prayer ~ Buddy Miller (Why?)
- Wicked Twisted Road [ENHANCED] ~ Reckless Kelly (Why?)
- Hacienda Brothers ~ Hacienda Brothers (Why?)

Association Rule Mining (ARM)

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

Market-Basket transactions

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**

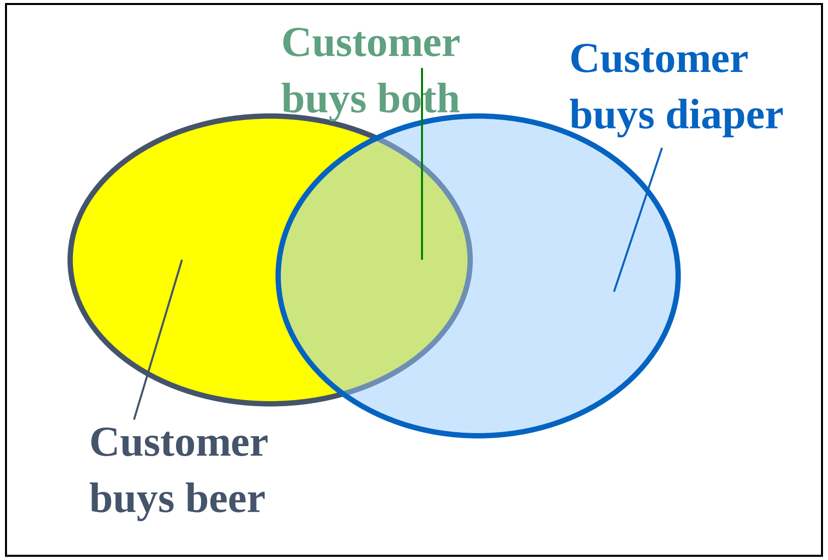
Example of Association Rules

{Diaper} $\rightarrow$ {Beer},
{Milk, Bread} $\rightarrow$ {Eggs,Coke},
{Beer, Bread} $\rightarrow$ {Milk},

Implication means co-occurrence, not causality!

Basic Concepts: Frequent Patterns

Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk



- ***itemset***: A set of one or more items
- ***k-itemset:*** $X = \{x_1, \ldots, x_k\}$
- ***(absolute) support***, or, ***support count*** of X: Frequency or occurrence of an itemset X
- ***(relative) support***, *s*, is the fraction of transactions that contains X (i.e., the probability that a transaction contains X)
- An itemset X is ***frequent*** if X's support is no less than a ***minsup*** threshold

Basic Concepts: Association Rules

Body → Consequent (Support , Confidence)

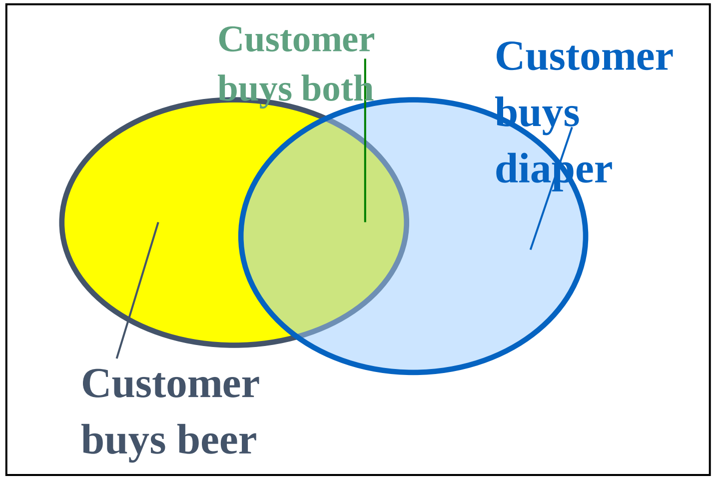
- *Body*: represents the examined data; i.e., the "IF" part = **antecedent**
- *Consequent*: represents a discovered property for the examined data; i.e., the "THEN" part

Antecedent and consequent are *disjoint* (i.e., have no items in common)

- *Support*: percentage of the records satisfying the *body* or the *consequent*
- *Confidence*: percentage of the records satisfying both the *body* and the *consequent* of those satisfying only the *body*

Basic Concepts: Association Rules

Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk



support, *s*, probability that a transaction contains X ∪ Y

confidence, *c*, conditional probability that a transaction having X also contains *Y*

Find **all** the rules *X* → *Y* with minimum support and confidence

Let min support = 50%, min confidence = 50%

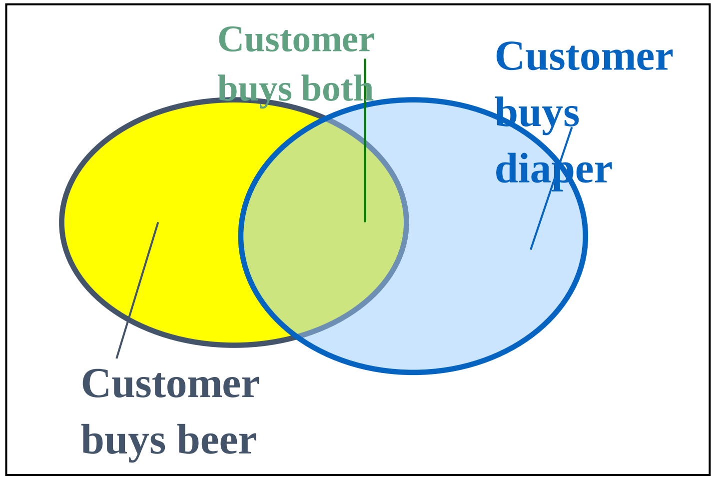
Frequncy Patterns:

Beer:3, Nuts:3, Diaper:4, Eggs:3, {Beer, Diaper}:3

- Association rules: (many more!)
  - *Beer* → *Diaper* (60%, 100%)
  - *Diaper* → *Beer* (60%, 75%)

Basic Concepts: Association Rules

Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk



- Find **all** the rules $X \rightarrow Y$ with minimum support and confidence
  - ***support***, s, probability that a transaction contains $X \cup Y$

$$s = P(X \cap Y)$$

  - ***confidence***, c, conditional probability that a transaction having X also contains Y

$$c = P(Y|X) = \frac{P(X \cap Y)}{P(X)}$$

Association-rule mining task

Given a set of transactions **D**, the goal of association rule mining is to find **all** rules having

- support ≥ ***minsup*** threshold
- confidence ≥ ***minconf*** threshold

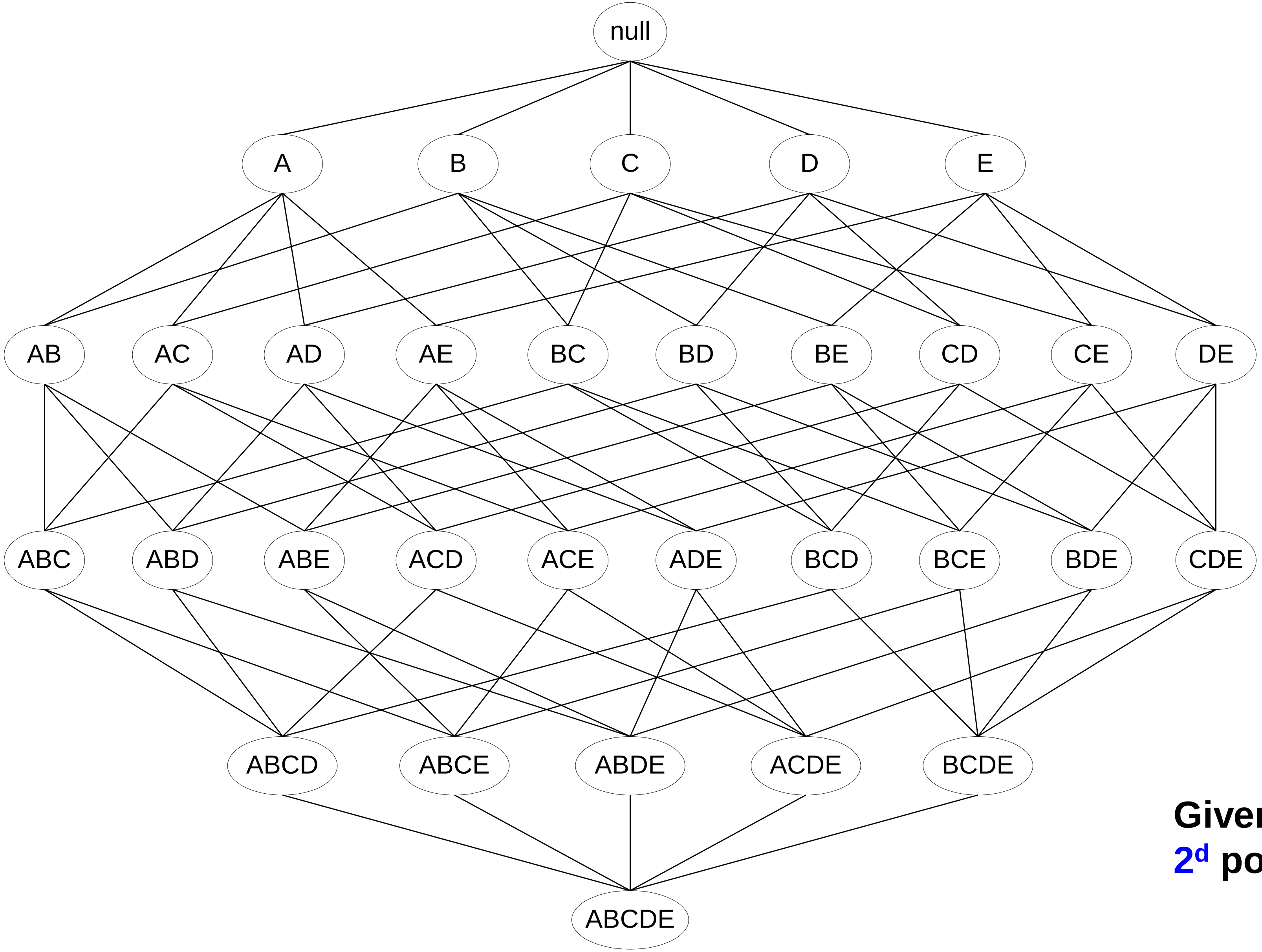
Finding frequent sets

- Notation: The input is a transaction database **D** where every transaction consists of a subset of items from some universe ***I***

- **Task:** Given a transaction database **D** and a **minsup** threshold find all frequent itemsets and the frequency of each set in this collection

- **Stated differently:** Count the number of times combinations of attributes occur in the data. If the fraction of the combination is above **minsup** report it.

How many itemsets are there?



Given d items, there are 2^d possible itemsets

When is the task sensible and feasible?

- If **minsup = 0**, then all subsets of ***I*** will be frequent and thus the size of the collection will be very large
- This summary is very large (maybe larger than the original input) and thus not interesting
- The task of finding all frequent sets is interesting typically only for relatively large values of **minsup**
  - *It is also probably only useful with minsup relatively large.*

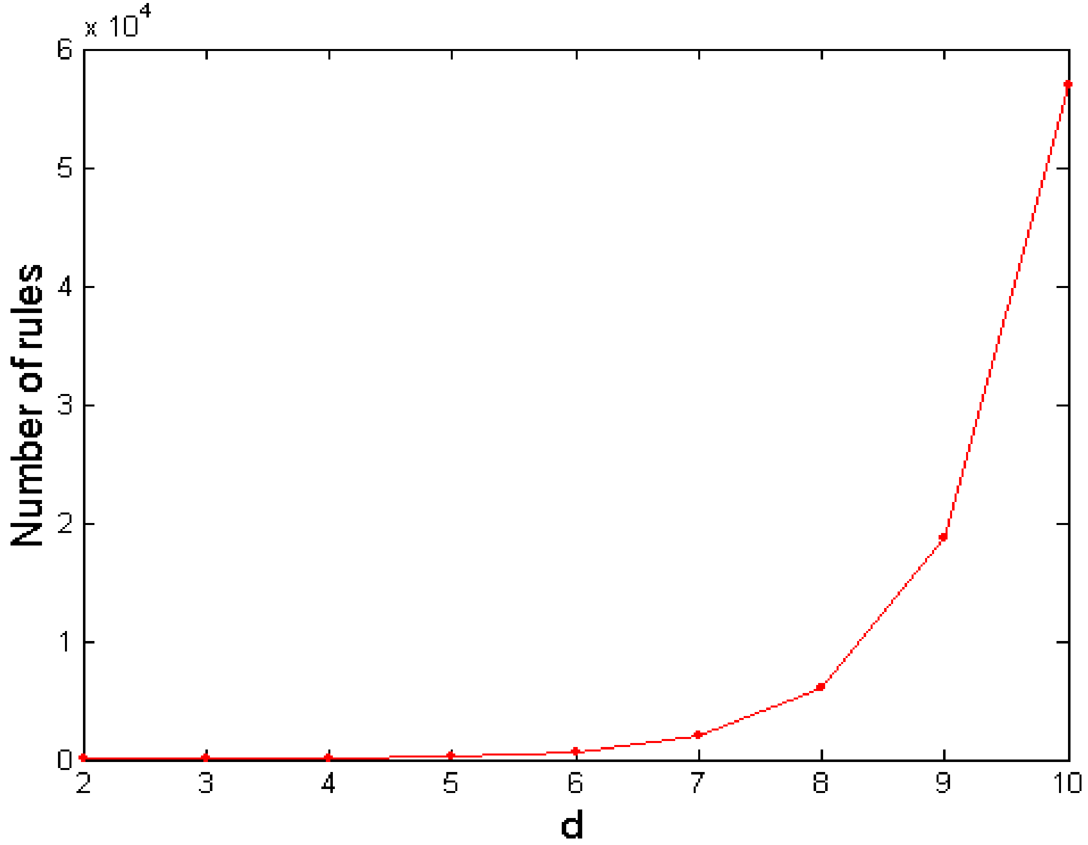
Brute-force algorithm for ARM

- List all possible association rules
- Compute the support and confidence for each rule
- Prune rules that fail the *minsup* and *minconf* thresholds

→ **Computationally prohibitive!**

How many association rules are there?

- Given **d** unique items:
  - Total number of itemsets = 2^d
  - Total number of possible association rules:



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

$$= 3^d - 2^{d+1} + 1$$

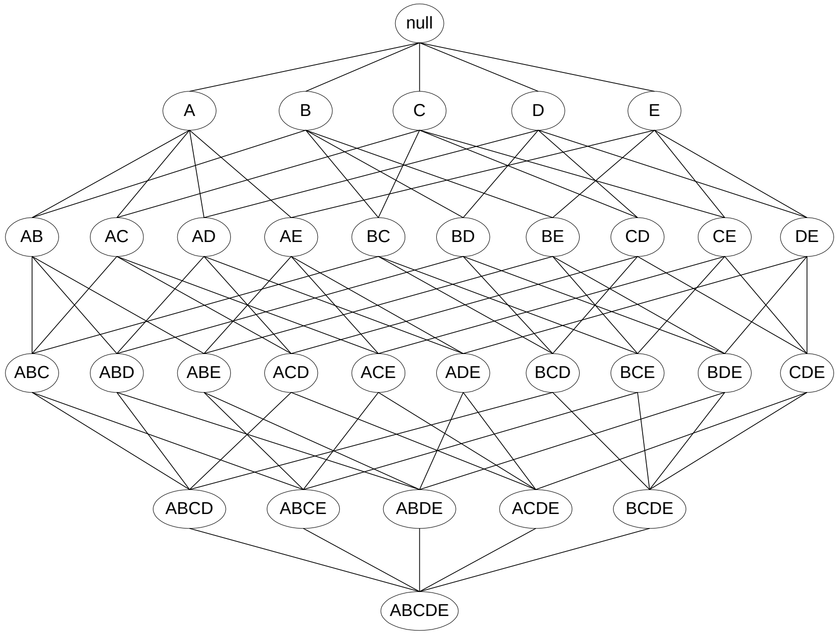
If d=6, R = 602 rules
If d=10, R = 57,002 rules
If d=20, R = 3,484,687,250 rules

Speeding-up the brute-force algorithm

- Reduce the number of candidates (M)
  - Complete search: $M=2^d$
  - Use pruning techniques to reduce M
- Reduce the number of transactions (N)
  - Reduce size of N as the size of itemset increases
  - Use vertical-partitioning of the data to apply the mining algorithms
- Reduce the number of comparisons (NM)
  - Use efficient data structures to store the candidates or transactions
  - No need to match every candidate against every transaction

Reduce the number of candidates

- **Apriori principle (main observation):**
  - If an itemset is frequent, then all of its subsets must also be frequent



- Apriori principle holds due to the following property of the support measure:

$$\forall X, Y : (X \subseteq Y) \Rightarrow s(X) \geq s(Y)$$

  - The support of an itemset ***never exceeds*** the support of its subsets
  - This is known as the ***anti-monotone*** property of support

Example

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

s(Bread) = 4/5
s(Bread, Beer) = 3/5

s(Bread) > s(Bread, Beer)

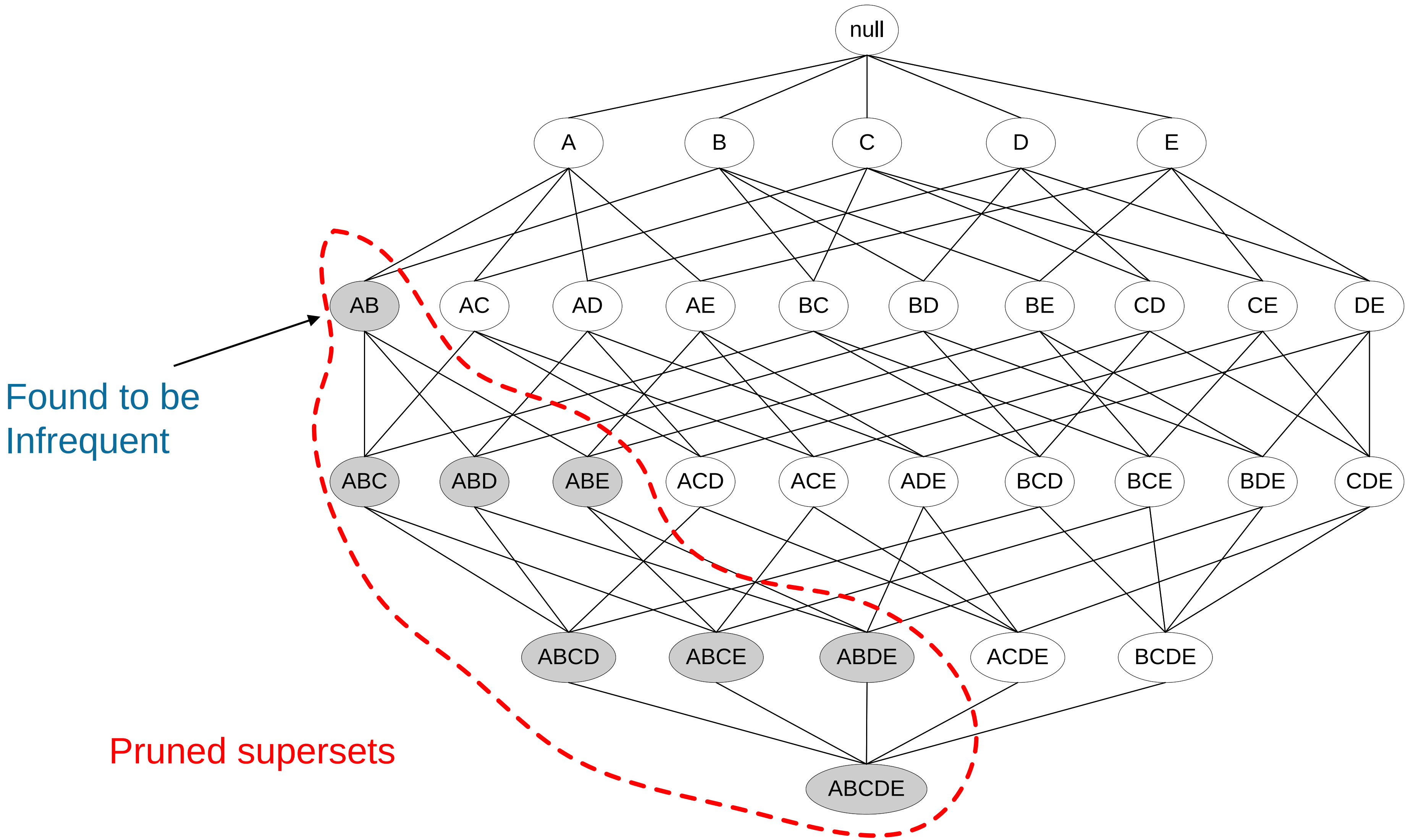
s(Milk) = 4/5
s(Bread, Milk) = 3/5

s(Milk) > s(Bread, Milk)

s(Diaper, Beer) = 3/5
s(Diaper, Beer, Coke) = 1/5

s(Diaper, Beer) > s(Diaper, Beer, Coke)

Illustrating the Apriori principle



Illustrating the Apriori principle

minsup = 3/5

Items (1-itemsets)

Item	Count
Bread	**4**
Coke	**2**
Milk	**4**
Beer	**3**
Diaper	**4**
Eggs	**1**

Pairs (2-itemsets)

No need to generate candidates involving Coke or Eggs!

Itemset	Count
{Bread,Milk}	**3**
{Bread,Beer}	**2**
{Bread,Diaper}	**3**
{Milk,Beer}	**2**
{Milk,Diaper}	**3**
{Beer,Diaper}	**3**

Triplets (3-itemsets)

No need to generate candidates involving {Bread, Beer} or {Milk,Diaper}!

Itemset	Count
{Bread,Milk,Diaper}	**3**

If every subset is considered,
$^6C_1 + {}^6C_2 + {}^6C_3 = 41$
With support-based pruning,
6 + 6 + 1 = 13

Exploiting the Apriori principle

1. Find frequent 1-items and put them to L_k (**k=1**)
2. Use L_k to generate a collection of *candidate* itemsets C_{k+1} with size (**k+1**)
3. Scan the database to find which itemsets in C_{k+1} are frequent and put them into L_{k+1}
4. If L_{k+1} is not empty
   - **k=k+1**
   - Goto step 2

If there is any itemset which is infrequent, its superset should not be generated/tested!

R. Agrawal, R. Srikant: "Fast Algorithms for Mining Association Rules", *Proc. of the 20th Int'l Conference on Very Large Databases*, 1994.

The Apriori Algorithm—An Example

minsup = 2/4

Database

Tid	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

1st scan

C_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

L_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{E}	3

C_2

Itemset
{A, B}
{A, C}
{A, E}
{B, C}
{B, E}
{C, E}

2nd scan

C_2

Itemset	sup
{A, B}	1
{A, C}	2
{A, E}	1
{B, C}	2
{B, E}	3
{C, E}	2

L_2

Itemset	sup
{A, C}	2
{B, C}	2
{B, E}	3
{C, E}	2

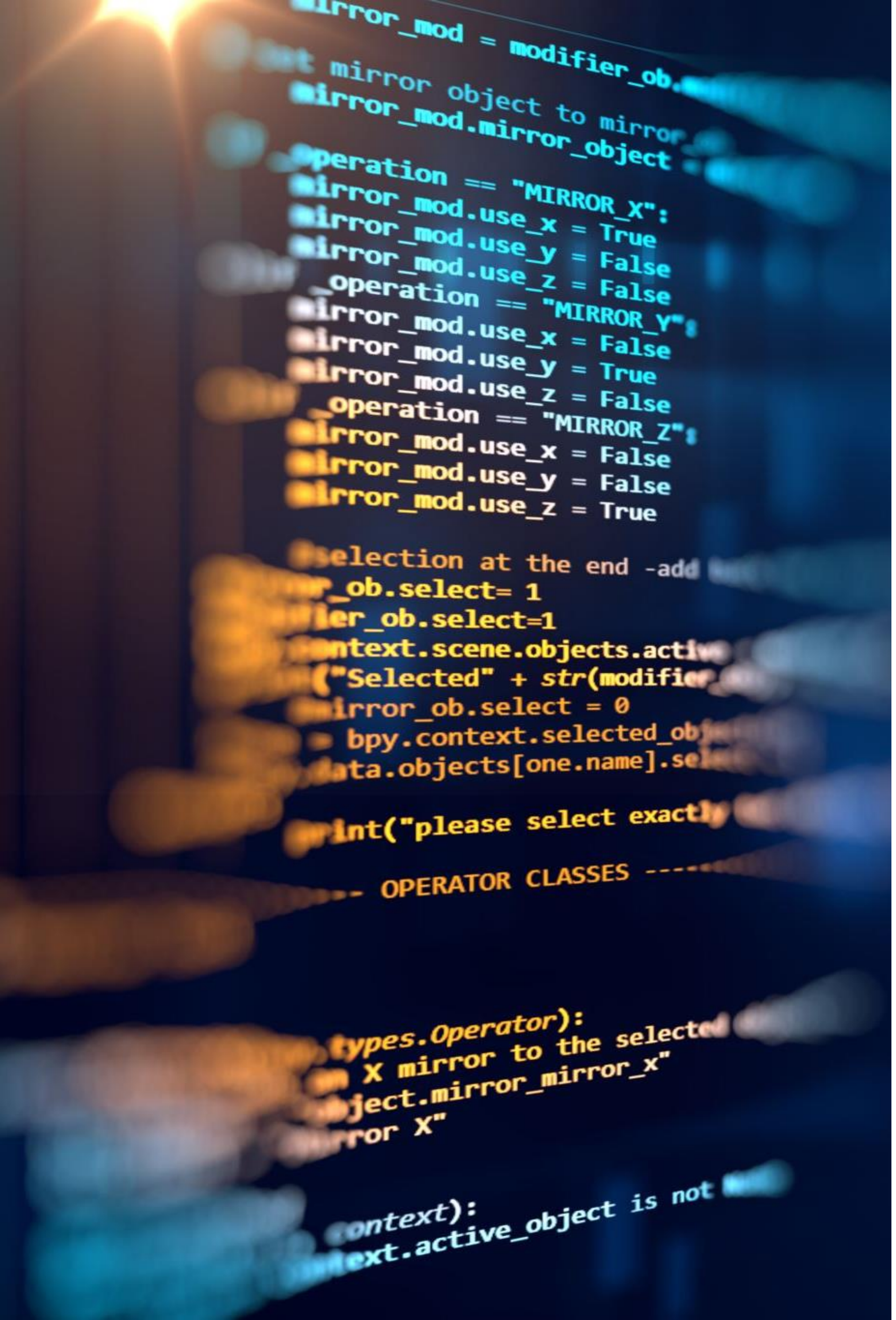
C_3

Itemset
{B, C, E}

3rd scan

L_3

Itemset	sup
{B, C, E}	2

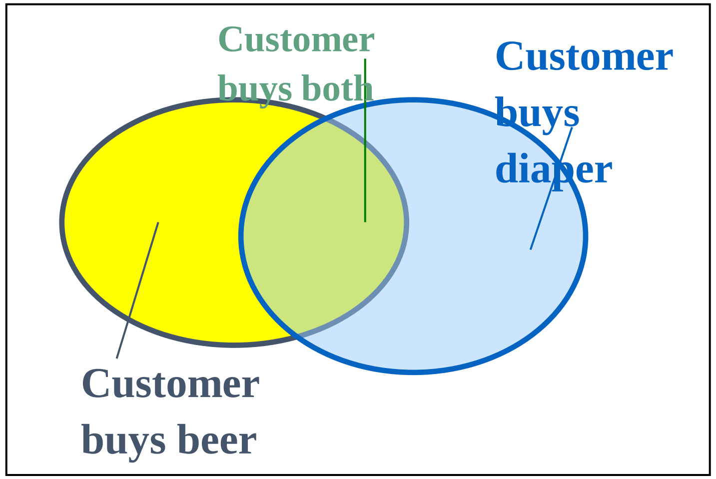


Apriori algorithm

- Much faster than the Brute-force algorithm
  - It avoids checking all elements in the lattice
- The running time is in the worst case **$O(2^d)$**
  - Pruning really prunes in practice
- It makes multiple passes over the dataset
  - One pass for every level **k**
- Multiple passes over the dataset is inefficient when we have thousands of candidates and millions of transactions

Association Rules

Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk



Rule: X→ Y

support, ***s(X→ Y)***, probability that transaction contains X ∪ Y

confidence, ***c(X→ Y)***,, conditional probability that a transaction having X also contains *Y*

coverage: *support of LHS, i.e.,* ***s(X)***

lift: *ratio of observed support to the expected support if the items were independent*

rule	support	confidence	coverage	lift	count
{grapes,mustard} => {onions}	0.000508	0.833333	0.00061	26.87158	5

See the file ARM.R for examples!